symmetry

MDPI

Article

Secure Registration Protocol for the Internet of Drones Using Blockchain and Physical Unclonable Function Technology

Norbert Oláh *,†, Botond Molnár † and Andrea Huszti

Department of Data Science and Visualization, University of Debrecen, 4028 Debrecen, Hungary; molnar.botond@inf.unideb.hu (B.M.); huszti.andrea@inf.unideb.hu (A.H.)

* Correspondence: olah.norbert@inf.unideb.hu

† These authors contributed equally to this work.

Abstract: Unmanned aerial vehicles (UAVs) have become increasingly popular in recent years and are applied in various fields, from commercial and scientific to military and humanitarian operations. However, their usage presents many challenges, including limited resources, scalability issues, insecure communication, and inefficient solutions. We developed a secure and scalable registration protocol to address these issues using LoRa technology. Our solution involves the usage of the physical unclonable function (PUF) and blockchain technology for key exchange. PUF also ensures security against physical tampering, and blockchain is applied to share the symmetric key among the base stations. After the registration, the later communication messages are encrypted with AES-GCM to provide authentication and confidentiality between the parties. We conducted a security analysis of the registration protocol using the ProVerif tool, and our solution meets the security requirements , including the mutual authentication of entities, key freshness, key secrecy and also key confirmation properties. Besides the Proverif-based analysis, an informal security analysis is also provided that shows that the registration is protected against a variety of well-known active and passive security attacks. As drone resources are limited, we also prepared a proof of concept to test our solution under real-life conditions, focusing on efficiency and lightweight operations.

Keywords: unmanned aerial vehicles; physical unclonable function; Internet of Drones; blockchain; Proverif; lightweight cryptography; LoRa

Citation: Oláh, N.; Molnár, B.; Huszti, A. Secure Registration Protocol for the Internet of Drones Using Blockchain and Physical Unclonable Function Technology. *Symmetry* **2023**, *15*, 1886. https://doi.org/10.3390/sym15101886

Academic Editor: Kuo-Hui Yeh

Received: 6 August 2023
Revised: 19 September 2023
Accepted: 25 September 2023
Published: 7 October 2023

1. Introduction

Nowadays, the Internet of Things (IoT) is increasingly essential in our daily lives and appears in many areas, such as smart homes, Industrial IoT (IIoT) or Internet of Drones (IoD). Sensors, devices and applications make our lives easier; however, they also collect sensitive data, which can lead to security problems. Drones are unmanned vehicles that operate remotely or can be autonomous, and they can operate on water, land and air. One type of drones used as a flying vehicle is the unmanned aerial vehicle (UAV), which has gained popularity in recent years due to its affordability, accessibility, and versatility. Originally developed for military purposes, drones have found their way into various civilian applications, such as video recording, agricultural work, sports activities, and territorial supervision. With the ever-increasing demand for drone technology, their usage is predicted to become a part of daily life soon, which is shown by the package delivery services by Amazon, the ambulance drones adopted by ambulance services of the Netherlands, providing essential medicine until the arrival of ambulance personnel, and other commercial drones. The Internet of Drones (IoD) is an architecture designed to provide communication and coordinated access to controlled airspace over the Internet between drones, base stations and users. Figure 1 represents a typical IoD architecture.

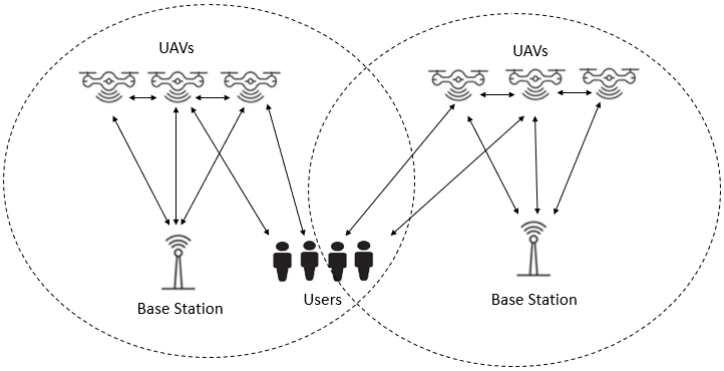

Figure 1. Internet of Drones architecture.

The security of an Internet of Drones (IoD) network is critical, as inappropriate authentication and unauthorized access to the network can result in several security issues, like sensitive data theft, impersonation of the IoD participants, or drone hijacking and interception. Although the potential benefits of such a network are vast, these benefits come with potential risks, which must be addressed through robust and secure schemes and protocols.

1.1. Related Work

In the scientific literature, various authentication methods for IoD systems have been suggested. These approaches fall into different categories, such as identity-based cryptography ([1,2]), certificateless ([3–5]), signcryption ([6,7]), blockchain-based ([8–10]), or physical unclonable function (PUF)-based solutions ([11,12]). In [13], the aims are to create a centralized IoD scheme, where MEC (mobile edge computing) operators are only in contact with one central UAV service provider (USP). In their proposition, when a drone (D_1) intends to connect to MEC_2 from MEC_1, the system's flexibility and the drone's freedom of movement are restricted. Since MEC_1 and MEC_2 belong to different service providers, D_1 must be authorized to connect to MEC_2. In such a scenario, the drones' access rights stored by MEC_1 and MEC_2's may fall out of sync, leading to a deadlock where MEC_1 permits the drone to enter the territory or join the network while MEC_2 blocks the action.

A unique aspect of a PUF is that the drone does not store any keys physically. The storage of keys in non-volatile memory typically exposes ICs to hardware attacks, enabling adversaries to read the memory content. In contrast, applying PUF, the drone does not store the key at all. Even if someone tries to probe PUFs, their response to a challenge can be significantly affected. Therefore, managing a key using PUFs is highly secure and resistant to physical attacks, and physical unclonable function (PUF)-based authentication for drones can offer several advantages. Moreover, it provides protection against forgery or impersonation by using unique bitstrings based on physical characteristics during the authentication process. Another advantage of PUFs over other hardware-based security systems is that even if a hacker has physical access to the device, he or she cannot clone its internal features. Gope et al. introduced an anonymous authentication system ([14]) for UAV applications utilizing radio frequency identification (RFID) and PUFs. The scheme consists of two entities: a UAV with a tag and a server. The proposed scheme provides mutual authentication and scalability, and the security of their scheme is based on Ouafi and Phan's security model. In contrast to our proposition, their registration phase is based on a secure communication channel between the drone and the server, which increases the time complexity. Moreover, in later communication, the drones cannot communicate with each other, only the server. Another PUF-based authentication scheme for drones, proposed by Alladi et al. ([15]), is where the usage of PUFs provides a lightweight and secure method to authenticate the IoD parties and achieve UAV-to-ground station and UAV-to-UAV communications. The scheme includes a system model with a ground station, legitimate UAVs, and an adversary drone. The proposed scheme has security features, in-

cluding key agreement, mutual authentication, and forward secrecy. Using Mao Boyd logic, formal security analysis has shown that this scheme is secure. The work of Alladi et al. is similar in many ways to our proposal, but the essential differences are that in their scheme, the base station generates the session key between the two drones, and the base station stores the PUF values in a central database. Since IoD systems are distributed, applying private blockchains to store, share and manage these values may be desirable.

In [16], proposed by Bera et al. a new blockchain-based framework (BSD2C-IoD) for secure data management among communication entities in IoD environments is presented. Their approach can effectively withstand various potential attacks, and the formal security analysis of BSD2C-IoD is conducted through the ROR oracle model, while formal security verification is carried out by utilizing the AVISPA tool. This solution uses the blockchain to store and validate the data. The communication and registration of drones are executed through a secure channel with the control room and verified by a registration authority as a trusted third party. Drones are resource-constrained devices; therefore, certificate-based cryptography for the secure channel can be inefficient. Tan et al. suggest a distributed key management system ([17]) for a flying ad hoc network (FANET) using blockchain technology in their research. The system includes multiple clusters of UAVs, each consisting of a head UAV with strong transmission power, ample storage space, high computing capabilities, and multiple-member UAVs, which can only handle lightweight tasks. The head UAVs are responsible for distributing cluster keys, updating public/private key pairs, migrating between clusters, and securely revoking any malicious UAVs. Security analysis and performance evaluations demonstrate that this system can withstand various attacks. Although Tan's key management system takes advantage of blockchain, drones use symmetric and asymmetric encryption and Schnorr signatures, but compared to our scheme, we find that ours is more lightweight. Some studies are reviewed and compared that are relevant to our solution. In this survey, you can find more comprehensive reviews of these approaches [18].

1.2. Our Contribution

In this work, a registration protocol is proposed, where a new drone joins the IoD system and exchanges secret keys. During the design, we considered that most of the drones are resource-constrained; hence lightweight cryptographic algorithms are applied. An important innovation in the registration scheme is the use of both PUF and blockchain technology. PUF technology provides a secure authentication, even for physical attacks (e.g., drone capture attack), and to verify the correctness of the PUF values securely, they are stored on the blockchain with only base station access. Besides the formal and informal security analysis, an efficiency evaluation is provided.

When registering a new drone, it is essential that the new drone is authenticated and exchanges a key between the participants to ensure confidential and authenticated communication later. If PUF-based authentication is used, it is necessary to ensure that other parties in the IoD network know this value; therefore, the secrecy of this value must be preserved. As drones can be intercepted or hijacked, we restrict registration so that only base stations have access to this PUF value. However, with traditional central data storage solutions, several problems can arise. Such concerns include single points of failure, compromise of the central database, synchronization of the participants, and the immutability of providing data. Therefore, we propose a private blockchain, where only the base stations have access to the distributed ledger containing the PUF values of the drones. In this way, we can take advantage of the distributed nature of the blockchain, which fits the IoD architecture, share these data between base stations, manage or restrict access to this sensitive information, and synchronize and make these PUF values available with higher availability. Furthermore, the blockchain provides an immutable property, i.e., the data origin is ensured, and it can be seen that the manufacturer sets the value, and it has not been changed. Another major problem with the IoD network is the lack of scalability. When designing our protocol, we allowed any number of drones to be

registered and added to the network. In addition to scalability, efficiency is another critical consideration, and we prefer using operations that are not resource-intensive (message authentication code, scalar multiplication) regarding the computational resource limitations of most of the drones. Lightweight cryptographic algorithms are designed to consume minimal computational resources. The NIST report [19] provides an overview of their lightweight cryptography project, where NIST approves the AES-GCM block cypher to provide the confidentiality and integrity of messages and standalone MACs, like CMAC, GMAC, and HMAC, for authentication. However, the use of lightweight cryptography is not only valuable for IoD, but it is also a frequently proposed solution for other areas of IoT [20,21]. Our scheme applies these cryptographic primitives to correspond to the NIST. After the successful registration, the drones can send authenticated and confidential messages to every participant (registered drones or the base station) in the IoD network. It is vital to prove the security of the proposed protocol to ensure that it meets the security requirements and that attackers cannot compromise the registration. The security analysis of our scheme is carried out by ProVerif, and we check that an attacker cannot impersonate the participants, and the confidentiality of the key exchange is also provided. Besides examining the security requirements specific to traditional key exchange protocols, we also provide an informal security analysis in which we discuss other aspects as well. Such aspects include denial of service, jamming attacks, blockchain manipulability, drone capture attacks, man-in-the-middle, and replay attacks. Finally, we were careful in the practical implementation of the designed scheme. During the implementation, we investigate and compare the possible communication modes, and we choose LoRa technology due to its advantages. The proof of concept of the scheme consists of a fixed-wing drone and a base station and an efficiency analysis of the proposed scheme prepared. Our proposal is evaluated against other existing solutions in the Related Work and the Practical issues sections (Tables 1 and 4). We consider whether a key is exchanged between the parties (whether the system is capable of drone-to-drone or drone-to-base station communication), efficiency (whether the solution uses lightweight algorithms), physical security (whether it uses PUF), the security of system-critical sensitive data (whether it uses blockchain or centralized data storage), and whether security analysis and implementation are available. We also evaluate the number of interaction steps and the communication and storage costs.

Table 1. Summary table of IoD solutions. D2D: drone-to-drone, D2BS: drone-to-base station.

Comparison	D2D & D2BS	Lightweight	PUF	Blockchain	Sec. Analys.	Impl.
Gope et al. ([14])		✓	✓		✓	✓
Alladi et al. ([15])	✓	✓	✓		✓	✓
Tan et al. ([17])	✓			✓	✓	✓
Bera et al. [16]	✓			✓	✓	✓
Our scheme	✓	✓	✓	✓	✓	✓

1.3. Outline of Article

In Section 2, we give the necessary preliminaries; then, in Section 3, we present our IoD scheme. The security analysis of the registration by ProVerif is given in Section 4, which includes the security requirements. We implement our suggestion and analyze the computation and communication costs in Section 5. Finally, Section 6 contains the conclusion of this paper.

2. Preliminaries

In this section, we give the definitions of cryptographic primitives, which are used in our propositions. The registration of the drones applies the physical unclonable function (PUF), which determines a unique PUF value for every drone, and these values are stored on a private blockchain. The registration between the drones and the base station is based

on symmetric cryptographic primitives (e.g., message authentication code, symmetric encryption scheme). This section discusses the concepts of elliptic curves over finite fields. For more comprehensive information, we refer the reader to [22].

Let E denote an elliptic curve defined over a finite field $\mathbb{F}_q$, where q is a large prime and $P \in E(\mathbb{F}_q)$ is a point of order $\mathbf{n}$. Elliptic curve parameters are chosen in such a way that the system resists all known attacks on the elliptic curve discrete logarithm problem in $< P >$. Let σ denote the length of an elliptic curve point's binary representation. Elliptic curve cryptography (ECC) relies on the infeasibility of the elliptic curve discrete logarithm problem, which can be defined as follows:

Definition 1. *Let $P \in E(\mathbb{F}_q)$ be a point of order $\boldsymbol{n}$, and let $< P >$ be the subgroup of $E(\mathbb{F}_q)$ generated by P. The elliptic curve discrete logarithm problem is to determine the value of $a \in \mathbb{Z}_{\boldsymbol{n}}$ in the equation $A = aP$, for a given point $A \in < P >$.*

This is considered computationally infeasible; hence, it is the fundamental building block for elliptic curve cryptography. We review the definition of the computational Diffie–Hellman problem as well.

Definition 2. *Given $P, aP, bP \in P$ for some $a, b \in \mathbb{Z}_{\boldsymbol{n}}$, compute abP.*

The Diffie–Hellman protocol ([23]) is the most often used key agreement protocol for securely exchanging a session key between two parties. This key can be used to encrypt and authenticate messages. The elliptic curve Diffie–Hellman protocol is based on the infeasibility of computing elliptic curve discrete logarithms, where if CDHP is hard, then ECDLP is also hard.

Definition 3. *A message authentication code (MAC) is a set of polynomial-time algorithms that include functions Key_M, Mac, and Ver.*

1. *Key_M is a probabilistic algorithm and requires the security parameter κ, which determines the length of the key as an input to generate a key. The output key K is $|K| \geq \kappa$.*
2. *To generate a tag for a message, the algorithm Mac inputs a key K and the message $m \in \{0,1\}^*$. The output tag t is denoted by $t := Mac_K(m)$. Mac must be a deterministic algorithm.*
3. *To verify the authenticity of a message, the verification algorithm Ver inputs a key K, a message m, and a tag t. The algorithm's output is a bit b, where $b = 1$ indicates validity and $b = 0$ indicates invalidity. It is assumed that Ver is deterministic, and we express this as $b := Ver(m, t)$.*

Definition 4. *A symmetric encryption scheme is a tuple of polynomial-time algorithms that include functions (Key_E, Enc, Dec):*

1. *Key_E is a probabilistic algorithm and inputs the security parameter 1^κ to generate a key. The output of the algorithm is a random key $K \in \{0,1\}^\kappa$.*
2. *Enc is a probabilistic encryption algorithm that requires key K and plaintext $m \in \{0,1\}^*$, and outputs a ciphertext c.*
3. *Dec is a deterministic decryption algorithm that inputs key K and ciphertext c. The output of Dec is the plaintext m.*

A physical unclonable function (PUF) is a one-way function implanted in hardware components. The output of the PUF depends on small random deviations that occur during the manufacturing process of the chip. It means that, when queried with the same challenge multiple times, the outcome of the PUF may differ slightly due to environmental and operational factors, such as ambient temperature and terminal voltages. However, fuzzy extractors can eliminate divergences and convert them into deterministic functions [24,25]. PUFs can be used for hardware authentication and secure key generation for IoT security

due to their robustness against physical and invasive attacks and their ability to retain keys without storing them.

The blockchain is a decentralized and distributed database system whose primary goal is to eliminate and minimize the involvement of a trusted third-party. By utilizing cryptographic primitives, the blockchain ensures that the inserted data are immutable and the history of the data has not been changed. The distributed nature of the blockchains provides greater availability.

Definition 5. *Blockchain is a peer-to-peer distributed ledger that is cryptographically secure, append-only, immutable, and updateable only via consensus or agreement among peers [26].*

Private blockchains assure the secrecy of the data stored. Although various blockchains differ in handling and building blocks, their structures are similar. Each block has a unique header, and its block header hash value can identify it. Additionally, each block's header stores the previous hash, meaning an attacker must modify the entire chain to change a block's content. The hash function's avalanche effect and the one-way property are vital aspects of the proof of immutability. Ethereum (ETH) is a blockchain platform that was launched in 2015 ([27]). It is open-source and allows for the design and execution of distributed applications through the use of smart contracts, which are Turing-complete programs.

3. The Proposed Scheme

Our solution aims to provide a secure and scalable scheme for the Internet of Drones which is operable by a wide variety of devices, from small quadcopters to larger fixed-wing UAVs, even with relatively weak hardware. Our proposed scheme consists of the registration phase, where the new drone exchanges symmetric keys with all participants within the base station domain. These keys as input serve the authenticated encryption in later communication. In the registration, we apply the PUF to authenticate the new drone. It is important that the manufacturer uploads the PUF value to the private blockchain after drone production.

In the case of a private blockchain (or sometimes called permissioned blockchain), a single organization issues permission for the nodes to join and maintain the network; hence, it is less decentralized compared to the public ones. Since the organization chooses all the nodes, permissioned blockchains are more secure and also more efficient than the public chains. There are three performance metrics defined for blockchains [28]. The number of transactions carried out successfully in unit time is called throughput. The time between the submission of the transaction and its completion is defined as latency. Moreover, resource consumption is the amount of CPU and memory usage of the host system during the execution. Most of the private blockchains use voting-based consensus algorithms, e.g., Byzantine fault tolerance (BFT), or proof of elapsed time (PoET). Private blockchains are more efficient in utilizing resources. According to [28], for example, Hyperledger Fabric private blockchain completes about 50 transactions per second with about 5 s latency when 600 transactions are loaded in the case of simple applications (storage only). For our case, private blockchains are perfect solutions.

When a user buys a new drone, using the PUF value to register the drone, only the base station accesses the drone's unique PUF value in the blockchain Listing A1. That is why the new drone can communicate with other drones utilizing the base station during the registration, where the base station verifies and confirms the authenticity of the new drone for other drones. The PUF value is used with MAC to maintain the message integrity between the new drone and the base station.

After registration, the new drone will have a new symmetric key for everyone, which must be updated after each communication. This way, the communication will still meet the lightweight requirement later on. The scheme is scalable, as the user can add or delete a device to the network anytime. Since multiple base stations are part of the scheme with a given range in IoD, and the drones are mobile, we need to register the device in various

domains. In the case of base stations, to reach the PUF value, we need to share this value, where it is essential to have high availability and reduce the possibility of a single point of failure occurrence. A private blockchain can be an excellent solution to this, which provides the confidentiality of the PUF value and is synchronously available for the base stations, and, for example, reduces the number of erroneous queries.

3.1. Registration

The first step is generating the system parameters, and then the registration is executed between the drone and IoD network participants (the other drones and the base station). We differentiate three participants: a *new drone* (ND) asks for registration, the *base station* (BS) manages the registration and communicates with the earlier *registered drones* ($D_1, \ldots, D_n$). The set of all binary strings of finite length is represented by $\{0,1\}^*$. The concatenation of x and y strings is represented by $x||y$. Before registering, the system generates all necessary parameters and keys. System parameters par are given by $par = (E, q, \mathbf{n}, P, Mac)$, where $E, q, \mathbf{n}, P$ are the parameters of elliptic curve E; $Mac : \{0,1\}^* \rightarrow \{0,1\}^\nu$; $\nu \in \mathbb{N}$ is a MAC function; and ν is the size of the MAC value. System parameters are publicly known. To provide message authenticity between the base station and other drones, each drone possesses n symmetric keys ($K_{B_1}, \ldots, K_{B_n}$). In addition, for symmetric keys, each participant generates a random value ($s, s_0, \ldots, s_n \in \mathbb{Z}^*_{\mathbf{n}}$) and then computes the public $sP, s_0P, \ldots, s_nP$ which are elliptic curve points represented by a bitstring that is necessary for the key agreement. At the end of the registration, the new keys are generated by the elliptic curve Diffie–Hellman key exchange.

The device has a PUF key (Pf_K) generated by the static random access memory (SRAM) or the cryptography module of the drone's main central processing unit (CPU). This physical fingerprint is unique to every CPU and memory protection unit (MPU). After the drone device has been manufactured, the PUF value is registered in the private blockchain, and it is accessed by base stations. Pf_K guarantees authenticity between the new drone and the base station.

As the first registration step (Figure 2), the BS sends the list of the IoD network participants (ID_i), where $i \in \{1, \ldots, n\}$ and applies the new drone's PUF key Pf_K to provide the list's integrity ($MAC_{Pf_K}(ID_1, \ldots, ID_n)$). The BS has access to the private blockchain; therefore, it knows the associated PUF key based on the ID_{ND} of the drone to be registered. The first step when starting the registration is to query it. ND creates a MAC message $MAC_{Pf_K}(s_0P)$ using its Pf_K and the s_0P value, which is the element of the ND of the EC Diffie–Hellman key exchange, where $s_0 \in Z_{\mathbf{n}}$ is chosen randomly and kept secret and sends it to the base station. The base station checks whether the ND message for the key exchange is valid.

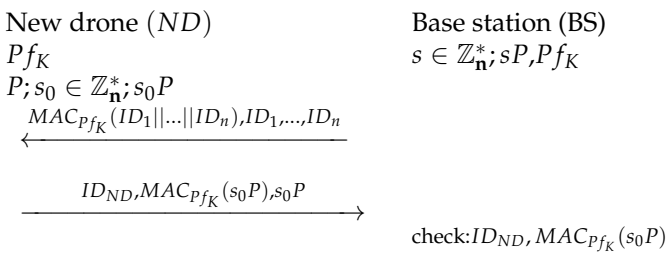

Figure 2. Request registration.

When BS checks the validity of the ND message, then as Figure 3 shows, BS computes new MAC values ($MAC_{K_{B_i}}(s_0P)$) using the earlier exchanged symmetric keys K_{B_i} ($i \in \{1, \ldots, n\}$) with the registered drones and broadcasts the key exchange value (s_0P) of the new drone. When the BS MAC values are received by drones D_i, where $i \in \{1, \ldots, n\}$, all D_i validate the BS message in parallel and calculate the new symmetric key for the new drone (K_{iD}). Each drone D_i creates the response ($MAC_{K_{B_i}}(s_iP)$, $MAC_{K_{iD}}(s_iP)$) and sends

them to the BS. BS checks the correctness of the D_i messages and then also computes the symmetric key (K_{BND}) with the new drone.

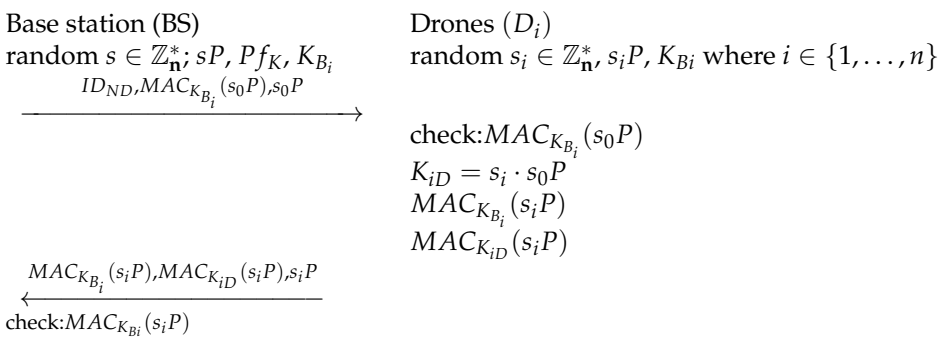

Figure 3. Communication between the base station and drones.

In the third phase of the registration (Figure 4), BS uses the Pf_K and MAC function to calculate the response to the new drone. When the BS response is received, ND verifies the message, and if it is correct, ND creates the symmetric keys (K_{BND}, K_{iD}), which are related to the IoD participants. To confirm the new symmetric keys, ND generates $rd \in \mathbb{Z}^*_{\mathbf{n}}$ random value and uses the new keys computing $MAC_{BND}(rd)$ and $MAC_{K_{iD}}(rd)$, then ND sends directly to the other drones and the base station. The drones and the base station check whether the new drone is calculated and possesses the correct symmetric keys.

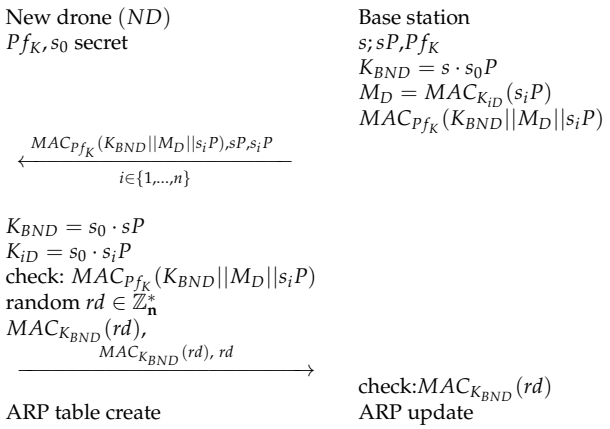

Figure 4. Drones registration key generation.

In the last step of the registration (Figure 5), the participants update their ARP (address resolution protocol) table, which consists of the IDs, IP addresses and symmetric keys.

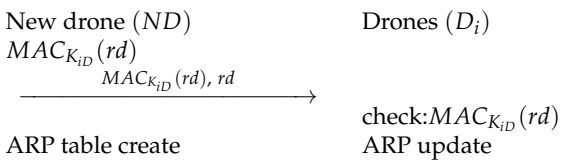

Figure 5. End of the drone registration and ARP table update.

ARP Table

The ARP table is a collection of data necessary for device communication with others. Table 7 demonstrates the structure of the table, which includes device addresses, statuses and types, buffer transfers and symmetric keys for each device. This allows peer-to-peer communication with any drone in the range of the base station, where the drone simply selects the address and associated symmetric key in the ARP table. When the message is created, which contains the value for the new key, the drone uses the AES-GCM symmetric-key cryptographic block cypher and sends the message to the selected party. In the table,

each drone's data take up only 18 bytes of storage, meaning even hundreds of drones would not take up 1 kilobyte of space. In Section 5, we also analyze the proposed ARP table's solution and find that it requires much less data storage compared to other solutions in related works. The communication needs to include a key exchange process, where we recommend the usage of the elliptic curve Diffie–Hellman key exchange.

While ARP tables are essential for simplifying device communication, they also pose security risks. Network participants should never share their ARP table with anyone, as it can expose symmetric keys, compromising the device's communication channels. A malicious attacker with access to a complete ARP table can impersonate a compromised device, participate in the network, and access sensitive data. The best solution is to avoid sharing any information present in the ARP tables over the network, except for device addresses. A possible solution to preserve the confidentiality of the values in the ARP table is to encrypt the table data with the PUF value. Even if a device falls into the hands of hostile entities or an unsecured area, the encrypted ARP table minimizes the risk of device hacking.

3.2. Communication

After a device goes through the registration process, it must establish a secure communication channel with other devices. When two drones want to communicate directly, all the initiating party has to do is look up the recipient's details in the ARP registry structure table. The receiving party looks up the decryption key for the received encrypted message. Then, using the appropriate symmetric key, the encrypted message is created using the AES-GCM block cypher algorithm. It is essential that AES-GCM is an authenticated encryption that ensures the confidentiality, integrity and authenticity of the message between the parties. During communication, the symmetric key needs to be updated, which can be done, for example, by using a challenge and response protocol besides the encrypted message to exchange a new key.

This method prevents UAV connection with unauthorized platforms, and each device pair will have its symmetric key, preventing unauthorized access. Even if one communication channel is compromised, the others remain secure. The algorithm also offers a lightweight and authenticated way of encrypting messages for added authenticity. The AES-GCM encryption method is ideal for its compatibility and scalability, even for weaker MCUs.

4. Security Analysis

In this section, we present the security analysis of our registration phase. After defining the security requirements, we provide analysis with applied pi calculus with the help of ProVerif. ProVerif was developed by Bruno Blanchet and is a tool that automatically verifies cryptographic protocols in the formal model [29]. It uses a representation of the protocol based on Horn clauses.

4.1. Security Requirements

We analyze the registration phase as a key exchange protocol. The typical security requirements for mutual entity authentication schemes and key-related requirements are considered. We prove the following four properties:

1. Authentication of parties.
   (a) Authentication of a new drone: it should not be possible for an adversary to act like a legitimate drone.
   (b) Authentication of the base station: adversaries should not be able to impersonate a legal base station.
   (c) Authentication of the other drones: adversaries should not be able to impersonate the other legal drones.
2. Secrecy of the key: during the key exchange, the newly generated key remains confidential, and an adversary should not have any information about the new key.

3. Key freshness: a new, randomly chosen key should be exchanged during a protocol runs.
4. Parties must confirm that the other party knows the new symmetric key and their ability to use it.

4.2. Adversarial Model

In the case of our protocol, the adversary's goals are to exchange a key with the base station and the drones successfully. We use the Dolev–Yao model [30] for our analysis. In the Dolev–Yao model, an adversary has the following properties:

1. The entire network is controlled by the adversary, who can act like a legitimate drone, intercepting, composing and creating any message, and is only constrained by the limitations of cryptographic techniques.
2. The adversary can initiate the protocol with any party and can be a receiver to any party.

4.3. Formal Model

We formalize the protocol as follows. We differentiate four processes. In the main process, Figure 6, the identification numbers for the participants, PUF keys, and symmetric keys are generated. Three sub-processes represent the protocols for the new drone (Figure A1), the base station and the already registered drones. We consider an unbounded number of sub-processes that run simultaneously. Our goal is to create a model representing the interactions between the new drone, the base station and the already registered drones. We also establish formal communication between the base station and the registered drones.

```
 let IoD =
(!BS(BSid,Did)) | (!D(BSid,Did)).
process
((!ND(NDid)) | !IoD)
```

Figure 6. Main process.

We formalize a registration process in which participants undergo mutual authentication. During this phase, the IoD participants and the new drone securely generate a new key.

4.4. Security Properties

ProVerif can analyze the properties of reachability, correspondence assertions, and observational equivalences. It is important to note that while ProVerif is sound, it is not complete. It means that if ProVerif determines that a property is satisfied, the model guarantees that it is valid. However, ProVerif may not be able to prove that a property holds.

ProVerif uses queries that might be a fact or correspondence for security evaluations. Primarily, we query whether a term *sec* is secret for the attacker: query attacker (sec). The query is a fact in the case of *reachability*, and we test whether the fact holds.

A *correspondence* is a form of $X \rightarrow Y$, which means if X holds, then Y also holds. We define events in the model as important stages and test whether event a has been executed, and then whether event b was previously executed. The $queryevent : a(s,t) \rightarrow event : b(t,z)$. means that for all s, t and each occurrence of $a(s,t)$, there is a previous occurrence of $b(t,z)$ for some z. To prove the one-to-one relationship, we apply injective correspondences. The $queryinj-event : a(s,t) \rightarrow inj-event : b(t,z)$. means that for each occurrence of event $a(s,t)$, there is a *distinct* earlier occurrence of the event $b(t,z)$ for some z.

We use the security queries and define six events demonstrated by Figure 7.

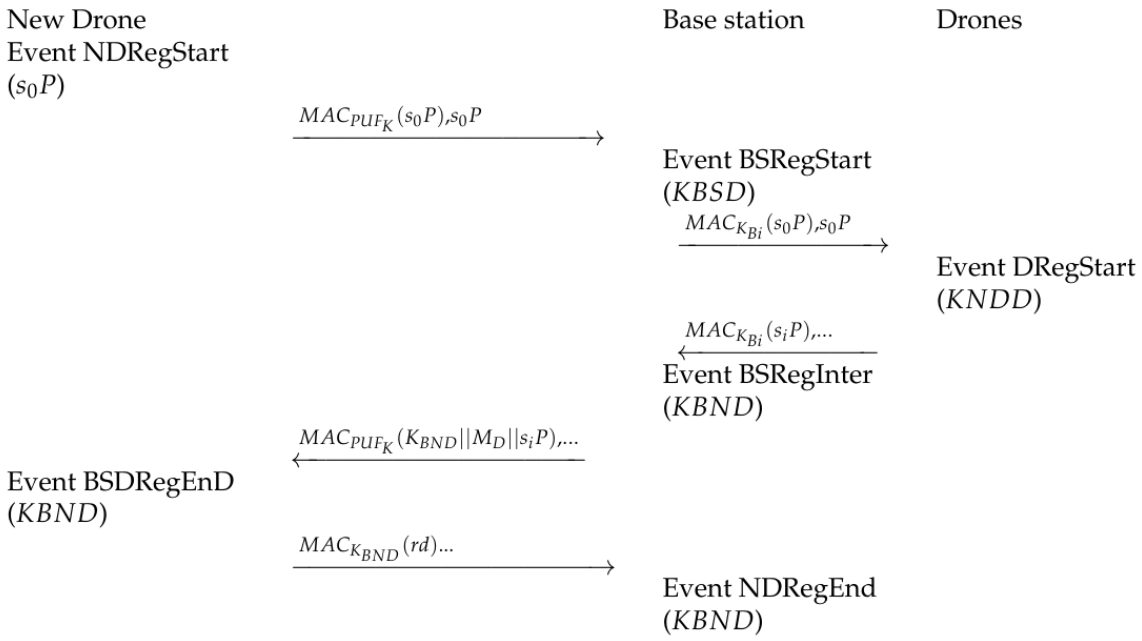

Figure 7. ProVerif-Events.

In the registration phase, we apply injective correspondences for mutual authentication of the new drone, base station and registered drones. We prove new drone authentication with the nested correspondence:

query a1:bitstring,a2:bitstring;a3:bitstring,a4:bitstring; inj-event(NDRegEnd(a4)) ==> (inj-event(BSRegEnd(a3))==>(inj-event(BSRegStart(a2))==>inj-event(NDRegStart(a1)))).

With another nested correspondence, we show the authentication of the base station and the already registered drones:

query a1: bitstring,a2:bitstring,a3:bitstring,a4:bitstring; inj-event(BSDRegEnD(a4)) ==> (inj-event(BSRegInter(a3))==> (inj-event(DRegStart(a2))==> inj-event(BSRegStart(a1)))).

All the queries above return the value true. Therefore, from the security requirements, participant authentication, i.e., mutual authentication of the new drone, the already registered drones, and the base station, are held, and the secrecy of the symmetric keys is also maintained in our model. Moreover, all verifications are fulfilled in the registration. Therefore, parties confirm that the other party knows and can use the new symmetric key when the user verifies the base station response or the base station, and the drones check the MAC value with the random input value in the last step of the protocol ($MAC_{K_{BND}}(rd), MAC_{K_{DND}}(rd)$). The key freshness property is also achieved because every registration requires a Diffie–Hellman key exchange, which results in a new, random key. Since the listed security requirements hold, the protocol is secure against man-in-the-middle and replay attacks as well. ProVerif is based on the perfect cryptography assumption, which means cryptographic primitives, such as symmetric encryption or MAC, are ideal black boxes. The cryptographic primitives need to satisfy the required security properties. We extend the formal security analysis with an informal analysis in order to take into account the typical attacks against IoD systems. The following informal security analysis explicates that registration and communication phases are immune to several attacks, such as drone capture attacks, denial-of-service attacks or jamming attacks.

- Drone capture attack: When physically capturing a drone, it is of the utmost importance to ensure that the attacker cannot extract any sensitive data. This precautionary measure is necessary to defend against potential breaches of secrecy of the information. In our case, the drone does not store the secret keys in the memory because of the PUF solution. PUF values cannot be extracted from the drones after they are closed. In our case, after the production, the PUF value is retrieved and stored on the blockchain

by the manufacturer, who closes the system. Hence, the adversaries cannot gain information about the PUF value, not even by physical tampering.

- Denial of service: Since most of the drones are resource-constrained, it is essential for them to be able to withstand various denial-of-service attacks. We consider the DOS attack only on the protocol level. Regarding the registration protocol, it is an important aspect that all messages are authenticated. If the validation fails, the incoming request is dropped. On the other hand, all calculations, including verifications, are chosen to be lightweight, and a time limit is also set for incoming responses.
- Forward and backward key secrecy: In the case of registration, the secret keys exchanged are randomized, and hence they are independent. If a secret key is leaked, neither the previous keys nor the latter ones can be calculated. Hence, forward and backward secrecy of the keys is assured.
- Jamming attacks: Many anti-jamming techniques exist today, including frequency and channel hopping, frame masking or special antenna designs for spatial filtering, but most of them are not suitable in practice. However, LoRa can utilize redundancy to improve packet resilience against jamming attacks by configuring a code rate (CR) parameter, controlling the ratio of the actual data to the forward-error-correcting capability added to the payload. The disadvantage of this solution is that the data rate may be reduced.

5. Practical Issues

Regarding new drone registration, we created an efficient way to register new drones using mainly MAC functions. Additionally, the registration process is distributed, which means other network participants can register new devices in parallel with the assistance of the base station.

When considering the communication cost, using the P-256 Weierstrass curve and HMAC-SHA-256, the space complexity is s $384 + i*208$ bytes cost, where i is the number of drones. The lengths of the variables are the following: IDs are 16 bytes, an HMAC is 32 bytes, and P-256 consist of 32-Byte x values and 32-Byte y values each. Assuming a swarm of 10 drones, our proposed protocol requires only 2.4 kilobytes of data for new drone registration. We conducted the analysis, which includes the number of operations and the execution time of each. We utilized a PC, a Raspberry PI 4 and an ESP32 for the analysis. The PC we used had an AMD Ryzen 5 5600X processor with 16 GB of RAM. The Raspberry PI is a widely popular, small on-board computer (SBC) that comes equipped with all the necessary software for basic computing. Although the manufacturer recommends Raspbian as the official OS, numerous other operating systems are available as well. In our case, we used a RasPi that features a Broadcom BCM2711, Quad core Cortex-A72 1.5GHz processor and 2 GB RAM. In the tables, i depends on the number of registered drones. Tables 2 and 3 show the results. Furthermore, as the number of drones was increased, we measured the registration times, which are shown in Figure 8 for the different devices.

Table 2. Execution time of operations in the registration.

Operation	Raspberry	ESP32	PC
AES-GCM	0.0000016	0.00155	0.0000033
HMAC	0.0000059	0.0082	0.0000011
EC scalar mult.	0.0205	0.2945	0.002880

Table 3. Number of operations in the registration.

Device\Number of Operation	Key Generation	HMAC	EC Multiplication
New Drone	1	3 + i	3 + i
Base station	-	4 + i	2
Registered drones	-	3	2

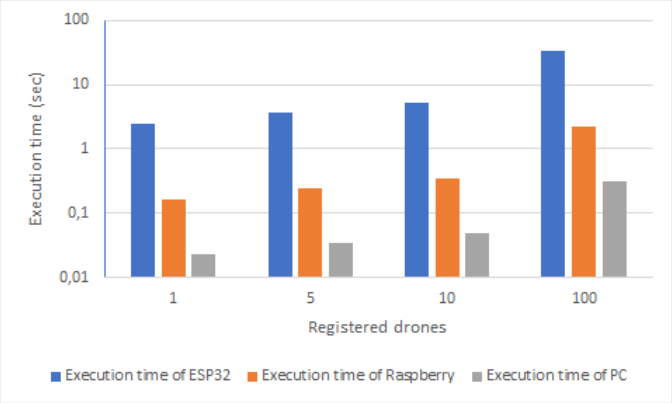

Figure 8. Execution time of registration on different devices.

We compared the efficiency analysis with the solutions in our contribution, considering the operations in the protocol steps (xor, concatenation, and random generation are ignored), summarizing the results in Table 4. Since the protocols to be compared are not all scalable, we performed the comparison for one drone registration and key exchange. In addition to the number of operations, we also considered the number of interactions, the amount of data sent and the size of the data stored on the drone. The motivation behind the latter aspect was that drones have limited computing and storage capacity, and the number of interactions can be important for protection against DoS and other attacks. Analyzing the other papers, the proposals use the SHA-1 hash algorithm, which has 20 bytes output; however, it does not provide collision resistance [31]. Therefore, we used a standard HMAC with 32 bytes output in our implementation, which increased our communication costs but provided collision resistance. Table 4 indicates that our protocol is cost-effective in terms of communication and storage. Additionally, it requires fewer interactions or operations in most cases.

Table 4. Number and execution time of operations for new drone.

Features	Gope [14]	Alladi [15]	Berra [16]	Our Scheme
Hash	13	6	12	8
PUF	2	2	-	1
EC multiplication	-	-	10	4
EC addition	-	-	3	-
Interactions	5	3	8	4
Storage cost at the UAV	96 bytes	44 bytes	152 bytes	18 bytes
Communication cost	224 bytes	200 bytes	280 bytes	384 bytes (180 bytes)

5.1. Prototype

We developed and tested the protocol in a real environment, and the implementation code is accessible through the link provided in [32]. First, we review the structure and size of each packet in the network interface, followed by the fields of the ARP table. Next, we discuss the elements of our proof of concept, including the base station and the drone program. Lastly, we compare the use of LoRa technology with other possible alternatives.

The LoRa network protocol generates packets. Each packet has a max size of 255 bytes and consists of two parts: header (Table 5) and payload (Table 6). Messages over 246 bytes are split into multiple encrypted packets using AES-GCM before transmission. The receiving end then rebuilds the message and checks the CRC (cyclic redundancy check) before processing it. Although CRC is the default in LoRa, the AES-GCM we use also ensures data integrity protection. Replay attacks are prevented by using a different IV with every message. An expiration date must be given for each IV, which is not too long and not too short, taking the hardware capabilities and security requirements into consideration. Each message must have its identifier, for example, *GPS coordinates for the next waypoint*.

Table 5. Packet header structure.

Name	Description	Size
Source Device Address	The device which transmits the packet.	1 byte
Destination Device Address	The device which receives the packet	1 byte
Number of Packets	The number of packets the total message is made of.	1 byte
Packet Number	The number of the packet, among the packets.	1 byte
Payload Size	The number of bytes the packet's payload is made of.	1 byte
Header CRC	16-bit cyclic redundancy check for error-checking purposes.	2 bytes

Table 6. Packet payload structure.

Name	Description	Size
Payload	The data of the packet	Max. 246 bytes
Payload CRC	16 bit cyclic redundancy check for payload	2 bytes

The ARP table has multiple fields for each registry (see Table 7), all necessary for tracking message activity on the network. Each registry has to contain the device's address on the network, the exchanged secret key, and the used initialization vectors with an expiration period. The device status serves as info about the state of the communication (e.g., if the key exchange is in progress or lost connection). Since, for the drone, it is vital that attackers do not have access to the memory and, thus, to the ARP table inside, we require that the fields of the ARP table are encrypted with the PUF key.

Table 7. ARP registry structure.

Name	Description	Size
Address	The registered device's address	1 byte
Device Status	Describes the device status	1 byte
AES key	Symmetric AES key	16 bytes
Received Initialization Vectors	Storing IVs to prevent replay attacks. Expiration time is required when working with low-resource devices, the exact value is platform and mission specific.	Not Specified

5.2. Proof of Concept

The prototype has two parts: a fixed-wing drone and a ground base station. We pushed the prototype to its limits using manual fly mode. At the same time, it was disturbed by two

other devices using LoRa on the same frequency, which can pose a challenge because the device obtains new instructions from the base station every 30 milliseconds. The ground base station is assembled from an ESP32 development board, LoRa RA-02 module, 2004A LCD alphanumeric LCD screen displaying status, joystick, and 10 kΩ potentiometer for controlling the throttle, all packed in a 3D-printed box. The drone platform is a 3D-printed fixed-wing aircraft model bought from Eclipson Airplanes. Its electronics are built of a 30A ESC (Electronic Speed Controller), 1200 kv A2212 brushless DC motor, 3000 mAh 3S LiPo battery, ESP32 development board and six pieces of servo motors. Even with obstacles, the drone still follow the commands, demonstrating that the communication system is reliable. To demonstrate the PUF in ESP-IDF, one small block of memory is closed for access to extract a key, which is based on the hardware attribute, and this key is uploaded to the blockchain, and any base station that participates in the blockchain can register and authenticate the drone. This blockchain can be accessed only by multiple service providers while remaining closed off to undesirable entities. It means that when a drone needs to switch to a new base station, the base stations do not have to synchronize with each other about the change. Instead, the drone can notify the old base station that it wants to disconnect and authenticate itself with the new base station, reducing the network overhead.

5.3. LoRa

Semtech, the developer and creator of LoRa technology, suggested LoRa, which is a wireless platform designed for the Internet of Things (IoT). LoRa, which stands for long range, is a spread spectrum modulation technique based on chirp spread spectrum (CSS) technology. It is known for its long-range capabilities and low power consumption, making it the preferred choice for IoT applications ([33]). Our comparison is made considering three main viewpoints: range as the most important factor, power consumption, data rate and availability.

LoRa is a versatile technology that can be used with any computing unit that has a free SPI (serial peripheral interface) interface. This makes it simple to integrate the network interface with different MCUs. Additionally, the affordability of LoRa makes it a practical physical layer for IoD applications.

Bluetooth: Although Bluetooth has low power consumption and is readily available, its range of 100 m makes it unsuitable for integrating into an IoD network because UAVs typically fly farther apart from each other than 100 m. As a result, the excellent properties of Bluetooth, such as bandwidth and connection reliability, are rarely used ([34]).

Bluetooth low energy: BLE has a longer-range mode that can cover up to 1000 m, making it suitable for a wider range of scenarios than regular Bluetooth. While it may not be ideal for large groups, BLE can transfer large amounts of data, such as images and videos, for a group of UAVs within a 1 km radius and part of the same network ([34]).

WiFi: It is a wireless technology based on radio data transmission. However, it is unsuitable for IoD since it has an even shorter maximum effective range than Bluetooth on 2.4 GHz (approximately 45 m indoors and 70 m outdoors using the 802.11 n protocol), making it an even less capable technology for IoD purposes. [35].

Cellular: The first viable technology for implementing an IoD network is 4G LTE. It has a significantly longer range that varies depending on the band it uses and terrain conditions, with low band having a range of approximately 40 km, mid-band ranging from 1.6 km to 12 km, and high band ranging from 15 to 600 m. Fourth-generation LTE also offers high data rates, particularly at lower frequencies, which, combined with its excellent range, makes it an excellent physical layer for the Internet of Drones applications. Fourth-generation LTE outperforms the overall capabilities of LoRa in various use cases and environments. Its most significant advantage is its range and flexibility, as well as its direct access to the internet ([36]). However, 4G LTE also has several drawbacks, including high power consumption, which can be significantly higher than that of LoRa when transmitting. At maximum transmit power, tested with an sx1278 LoRa module, the power consumption

of LoRa is around 370–400 mW at 20 dBm transmit power, while that of SIMCOM SIM7600E-H is around 3 W at 22.5 dBm transmit power. Additionally, 4G LTE comes with high costs due to service provider fees and the cost of modules. It is also heavily reliant on tower coverage and service providers, meaning that if UAVs venture into areas which are not covered by cellular towers, the devices lose connection to the network, leading to device and data loss. Furthermore, it may become unable to complete the mission without crucial data provided by other devices.

As a result, the use of cellular networks has been deemed unsuitable for IoD applications that require precision agriculture and military service, eliminating two of the most significant use cases. While cellular networks may be a viable solution for IoD networks placed in or near populated areas in developed countries, where 4G cellular coverage extends to 99% of these areas, this cannot be guaranteed in developing countries. Even if placed near populated areas, coverage may be limited, with 3G cellular coverage in developing regions such as Sub-Saharan Africa barely reaching 75%, despite rapid infrastructure expansion ([37]).

Satellite: This communication is another possible alternative to LoRa, with the added advantage of having a limitless range, as network traffic is handled by satellites. It means that if two communicating UAVs have satellite connections, they can communicate with each other, providing reasonable data rates and making it a suitable solution for IoD. However, there are also some concerns with this technology, such as its high cost and dependency on service providers, vulnerability to signal interference, high power consumption, and high latency.

5.4. Advantages of LoRa Technology for the Internet of Drones

We ultimately chose LoRa, as it was the most optimal option among those mentioned. LoRa stood out as a general-purpose physical layer due to its affordable price, low power consumption, long range, and disruption-resistant properties. Its main drawback is its relatively low data rate, which, even in the best-case scenario, maxes out at 22 kbps. Still, assuming only smaller messages are going through the network (GPS coordinates, events, status, and commands), this property's weakness is negligible (Table 8).

Table 8. LoRa comparison to other technologies.

Standpoint	LoRa	Bluetooth	BLE 5 Long Range	Wi-Fi (2.4 GHz)	Cellular (4G)
Range	5–10 km	50–100 m	400–1000 m	45–75 m	600–40,000 m
Affordability	EUR 7–EUR 8 per module	built-in or EUR 5–EUR 20 per module	built-in or EUR 5–EUR 20 per module	built-in or EUR 3–EUR 50 per module	EUR 20–EUR 50 per module + service provider fees
Maximum Possible Data Rate	22 kbps	2 Mbps	2 Mbps	72–600 Mbps (802.11n)	4–12 Mbps are common

6. Conclusions

Addressing and managing the risks associated with security incidents and physical damage has become critical, as our world applies more intelligent environments. We integrated a private blockchain and the PUF function to ensure suitable drone registration in the IoD systems. During the design, we paid attention to the scalability issue and effectively stored the keys on the ARP table. We also considered that the exchanged keys should be stored securely on the devices, and we assumed that the table's data are encrypted by the PUF key of the device. We used the exchanged symmetric key to provide authentication and confidentiality between the drone-to-drone and drone-to-base station

communications. For our application, we found that LoRa is an appropriate physical layer due to its long range, low power consumption, interference-resistant properties, and acceptable data rate. Our prototype tests proved that LoRa is a reliable physical layer that can support communication among multiple network participants at the same time. Additionally, our security analysis using ProVerif showed that attackers cannot access the exchanged key, and mutual authentication, key freshness, and key confirmation were also provided. In addition to the formal security analysis, we also aimed to informally examine the impact of some other drone-related attacks. We compared the communication and storage costs to solutions mentioned in the related literature for the practical aspects. The field of the Internet of Drones will face many challenges in the future, both from regulatory and technical points of view. These problems include privacy issues related to the data collected by drones, swarm management issues, the impact of artificial intelligence on drones, and many other topics.

Author Contributions: Conceptualization, N.O. and A.H.; Software, B.M.; Formal analysis, N.O. and A.H. All authors have read and agreed to the published version of the manuscript.

Funding: The presented research has been supported by the project no. 101083965—DigitalTech EDIH—DIGITAL-2021-EDIH—01 that has been implemented with the support provided from the European Union, co-funded by the European Commission´s Digital Europe Programme, financed under the DIGITAL-2021-EDIH funding scheme. The research was supported by the Ministry of Culture and Innovation NRDI Office within the framework of the Infocommunication and Information Technology National Laboratory Program.

Data Availability Statement: Prototype GitHub Link. Available online: https://github.com/Gepsonka/TDK (accessed on 1 January 2023).

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

Appendix A

```
 let ND(id:bitstring,Pkey:bitstring)=
in(c, (idBS:bitstring,idD:bitstring,M1:bitstring));
let IDs = concatenate(idBS,idD) in
let CMac1=mac(Pkey,IDs) in
if M1=CMac1 then
new s0:nonce;
let s0P=exp(p,s0) in
event first1(s0P);
let M2 = mac(Pkey, s0P) in
out(c,(M2,s0P));
in(c,(M5:bitstring,s1P:bitstring,sP:bitstring));
let KNDB = exp(sP,s0) in
let KNDD = exp(s1P,s0) in
let CMac51 = mac(KNDD, s1P) in
let CMac52 = mac2(Pkey,KNDB,CMac51,s1P,sP) in
if M5 = CMac52 then
new rd: nonce;
let brd = btyperd(rd) in
let M61 = mac(KNDB, brd) in
let M62 = mac(KNDD, brd) in
event fourth(KNDB);
event sixth(KNDD);
out(c,(M61,M62, brd)).
```

Figure A1. ProVerif—new drone process.

Listing A1. Smart contract for extracting and adding PUF keys.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract PUFStorage {
        bytes16[] private byteArrayList;

        function addPUFKey(bytes16 data) public {
                byteArrayList.push(data);
        }

        function getPUFKey(uint256 index) public view returns (bytes16) {
                require(index < byteArrayList.length, "Index out of range");
                return byteArrayList[index];
        }

        function getNumberOfPUFKeys() public view returns (uint256) {
                return byteArrayList.length;
        }

        function findPUFKey(bytes16 searchData) public view returns (bool, uint256) {
                for (uint256 i = 0; i < byteArrayList.length; i++) {
                        if (byteArrayList[i] == searchData) {
                                return (true, i);
                        }
                }
                return (false, 0);
        }
}
```

References

1. Jan, S.U.; Khan, H.U. Identity and aggregate signature-based authentication protocol for IoD deployment military drone. *IEEE Access* **2021**, *9*, 130247–130263. [CrossRef]
2. Haque, M.S.; Chowdhury, M.U. A new cyber security framework towards secure data communication for unmanned aerial vehicle (UAV). In Proceedings of the Security and Privacy in Communication Networks: SecureComm 2017 International Workshops, ATCS and SePrIoT, Niagara Falls, ON, Canada, 22–25 October 2017; Proceedings 13; Springer: Berlin/Heidelberg, Germany, 2018; pp. 113–122.
3. Nyangaresi, V.; Petrovic, N. Efficient PUF Based Authentication Protocol for Internet of Drones. In Proceedings of the 2021 International Telecommunications Conference (ITC-Egypt), Alexandria, Egypt, 13–15 July 2021; pp. 1–4. [CrossRef]
4. Khan, M.A.; Ullah, I.; Nisar, S.; Noor, F.; Qureshi, I.M.; Khanzada, F.U.; Amin, N.U. An Efficient and Provably Secure Certificateless Key-Encapsulated Signcryption Scheme for Flying Ad-hoc Network. *IEEE Access* **2020**, *8*, 36807–36828. [CrossRef]
5. Li, J.; Wang, Y.; Ding, Y.; Wu, W.; Li, C.; Wang, H. A certificateless pairing-free authentication scheme for unmanned aerial vehicle networks. *Secur. Commun. Netw.* **2021**, *2021*, 1–10. [CrossRef]
6. Khan, M.A.; Ullah, I.; Abdullah, A.M.; Mohsan, S.A.H.; Noor, F. An Efficient and Conditional Privacy-Preserving Heterogeneous Signcryption Scheme for the Internet of Drones. *Sensors* **2023**, *23*, 1063. [CrossRef] [PubMed]
7. Won, J.; Seo, S.H.; Bertino, E. A secure communication protocol for drones and smart objects. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 14 April–17 March 2015; pp. 249–260.
8. Singh, J.; Venkatesan, S. Blockchain mechanism with Byzantine fault tolerance consensus for Internet of Drones services. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4235. [CrossRef]
9. Aggarwal, S.; Shojafar, M.; Kumar, N.; Conti, M. A new secure data dissemination model in internet of drones. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
10. Alqarni, K.S.; Almalki, F.A.; Soufiene, B.O.; Ali, O.; Albalwy, F. Authenticated Wireless Links between a Drone and Sensors Using a Blockchain: Case of Smart Farming. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4389729. [CrossRef]
11. Semal, B.; Markantonakis, K.; Akram, R.N. A Certificateless Group Authenticated Key Agreement Protocol for Secure Communication in Untrusted UAV Networks. In Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK, 23–27 September 2018; pp. 1–8. [CrossRef]
12. Pu, C.; Li, Y. Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system. In Proceedings of the 2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), London, UK, 10–11 July 2023; IEEE: New York, NY, USA, 2020; pp. 1–6.
13. 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]
14. Gope, P.; Millwood, O.; Saxena, N. A provably secure authentication scheme for RFID-enabled UAV applications. *Comput. Commun.* **2021**, *166*, 19–25. [CrossRef]
15. Alladi, T.; Naren; Bansal, G.; Chamola, V.; Guizani, M. SecAuthUAV: A Novel Authentication Scheme for UAV-Ground Station and UAV-UAV Communication. *IEEE Trans. Veh. Technol.* **2020**, *69*, 15068–15077. [CrossRef]

16. Bera, B.; Saha, S.; Das, A.K.; Kumar, N.; Lorenz, P.; Alazab, M. Blockchain-Envisioned Secure Data Delivery and Collection Scheme for 5G-Based IoT-Enabled Internet of Drones Environment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9097–9111. [CrossRef]
17. Tan, Y.; Liu, J.; Kato, N. Blockchain-Based Key Management for Heterogeneous Flying Ad Hoc Network. *IEEE Trans. Ind. Informatics* **2021**, *17*, 7629–7638. [CrossRef]
18. Samanth, S.; Kv, P.; Balachandra, M. Security in Internet of Drones: A Comprehensive Review. *Cogent Eng.* **2022**, *9*, 2029080. [CrossRef]
19. Dworkin, M. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2007.
20. Alasmary, H.; Tanveer, M. ESCI-AKA: Enabling Secure Communication in an IoT-Enabled Smart Home Environment Using Authenticated Key Agreement Framework. *Mathematics* **2023**, *11*, 3450. [CrossRef]
21. Tanveer, M.; Badshah, A.; Khan, A.U.; Alasmary, H.; Chaudhry, S.A. CMAF-IIoT: Chaotic map-based authentication framework for Industrial Internet of Things. *Internet Things* **2023**, *23*, 100902. [CrossRef]
22. Menezes, A.; Vanstone, S.; Okamoto, T. Reducing elliptic curve logarithms to logarithms in a finite field. In Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, New Orleans, LA, USA, 5–8 May 1991; pp. 80–89.
23. Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654.
24. Van Herrewege, A.; Katzenbeisser, S.; Maes, R.; Peeters, R.; Sadeghi, A.R.; Verbauwhede, I.; Wachsmann, C. Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs. In Proceedings of the Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, 27 Februray–2 March 2012; Revised Selected Papers 16; Springer: Berlin/Heidelberg, Germany, 2012; pp. 374–389.
25. Delvaux, J.; Gu, D.; Verbauwhede, I.; Hiller, M.; Yu, M.D. Efficient fuzzy extraction of PUF-induced secrets: Theory and applications. In Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems, Santa Barbara, CA, USA, 17–19 August 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 412–431.
26. Bashir, I. *Mastering Blockchain*; Packt Publishing Ltd.: Birmingham, UK, 2017.
27. Buterin, V. Ethereum White Paper: A Next Generation smart Contract & Decentralized Application Platform (2013). Available online: https://github.com/ethereum/wiki/wiki/White-Paper (accessed on 1 January 2023).
28. Alom, I.; Ferdous, M.S.; Chowdhury, M.J.M. BlockMeter: An Application Agnostic Performance Measurement Framework for Private Blockchain Platforms. *IEEE Trans. Serv. Comput.* **2023**, 1–14. [CrossRef]
29. Blanchet, B. *Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif*; INRIA: Paris, France, 2016; Volume 1, pp. 1–135.
30. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
31. Wang, X.; Yin, Y.L.; Yu, H. Finding collisions in the full SHA-1. In Proceedings of the Advances in Cryptology–CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2005; Proceedings 25; Springer: Berlin/Heidelberg, Germany, 2005; pp. 17–36.
32. Prototype GitHub Link. Available online: https://github.com/Gepsonka/TDK (accessed on 1 January 2023).
33. Semtech. What Is LoRa? Available online: Https://www.semtech.com/lora/what-is-lora (accessed on 1 January 2023).
34. Sponas, J.G. Things You Should Know About Bluetooth Range. Available online: https://blog.nordicsemi.com/getconnected/things-you-should-know-about-bluetooth-range (accessed on 1 January 2023).
35. Mitchell, B. What Is the Range of a Typical Wi-Fi Network? 2020. Available online: https://www.lifewire.com/range-of-typical-wifi-network-816564 (accessed on 1 January 2023).
36. Simmons, A. Cell Tower Range: How Far Do They Reach? 2022. Available online: https://dgtlinfra.com/cell-tower-range-how-far-reach/(accessed on 1 January 2023).
37. Wyrzykowski, R. *Mobile Connectivity in Sub-Saharan Africa: 4G and 3G Connections Overtake 2G for the First Time*; GSMA: London, UK, 2020; Volume 16.