drones

MDPI

Article

Secure Blockchain-Enabled Authentication Key Management Framework with Big Data Analytics for Drones in Networks Beyond 5G Applications

Amit Kumar Mishra [1,2], Mohammad Wazid [1,*], Devesh Pratap Singh [1], Ashok Kumar Das [3], Jaskaran Singh [1] and Athanasios V. Vasilakos [4,*]

[1] Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India; akmishra@gehu.ac.in (A.K.M.); devesh.csit@geu.ac.in (D.P.S.); jaskaransingh_19021315.cse@geu.ac.in (J.S.)
[2] Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun 248 002, India
[3] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India; ashok.das@iiit.ac.in
[4] Center for AI Research (CAIR), University of Agder (UiA), 4879 Grimstad, Norway
* Correspondence: mohammadwazid.cse@geu.ac.in (M.W.); thanos.vasilakos@uia.no (A.V.V.); Tel.: +30-697-744-9705

Abstract: One of the most significant recent advances in technology is the advent of unmanned aerial vehicles (UAVs), i.e., drones. They have widened the scope of possible applications and provided a platform for a wide range of creative responses to a variety of challenges. The Internet of Drones (IoD) is a relatively new concept that has arisen as a consequence of the combination of drones and the Internet. The fifth-generation (5G) and beyond cellular networks (i.e., drones in networks beyond 5G) are promising solutions for achieving safe drone operations and applications. They may have many applications, like surveillance or urban areas, security, surveillance, retaliation, delivering items, smart farming, film production, capturing nature videos, and many more. Due to the fact that it is susceptible to a wide variety of cyber-attacks, there are certain concerns regarding the privacy and security of IoD communications. In this paper, a secure blockchain-enabled authentication key management framework with the big data analytics feature for drones in networks beyond 5G applications is proposed (in short, SBBDA-IoD). The security of SBBDA-IoD against multiple attacks is demonstrated through a detailed security analysis. The Scyther tool is used to perform a formal security verification test on the SBBDA-IoD's security, confirming the system's resistance to various potential attacks. A detailed comparative analysis has identified that SBBDA-IoD outperforms the other schemes by a significant margin. Finally, a real-world implementation of SBBDA-IoD is shown to evaluate its effect on several measures of performance.

Keywords: Internet of Drones (IoD); unmanned aerial vehicles (UAVs); cyber-attacks; blockchain; data analytics; security

Citation: Mishra, A.K.; Wazid, M.; Singh, D.P.; Das, A.K.; Singh, J.; Vasilakos, A.V. Secure Blockchain-Enabled Authentication Key Management Framework with Big Data Analytics for Drones in Networks Beyond 5G Applications. *Drones* **2023**, *7*, 508. https://doi.org/10.3390/drones7080508

Academic Editor: Emmanouel T. Michailidis

Received: 2 July 2023
Revised: 28 July 2023
Accepted: 31 July 2023
Published: 2 August 2023

1. Introduction

The term Internet of Drones (IoD) refers to the infrastructure that has been set up to enable users, servers, and drones to communicate with one another and share resources via the Internet. In point of fact, drones are rapidly becoming more mainstream goods, allowing users to pilot many drones simultaneously for various purposes within confined spaces [1]. Drones, also known as unmanned aerial vehicles (UAVs), are valuable instruments that can be used to address the issues that occur in people's day-to-day lives. An Internet of Drones (IoD) that connects drones is a trend that is widely desired to improve the safety and quality of flight in light of the growing number of drones that are operating in low-altitude airspace. This is because connecting drones will create an IoD [2,3]. Some of the important components of a drone are the RC transmitter, multirotor frame, motors/speed

controller, propellers, flight controller, battery, and landing gear. Integration of unmanned aerial vehicles (UAVs, also known as drones) into "fifth-generation (5G) and beyond cellular networks" is a promising solution for achieving safe UAV operation in addition to permitting expanded uses with the mission-specific payload delivery of information. This is because of the developments in cellular technologies and the widespread deployment of cellular facilities [4,5]. They may have many applications, i.e., surveillance of a city, security, and surveillance of border areas, delivery of items (i.e., medicines, vaccines), smart farming, film productions, capturing nature's activities (i.e., a volcano city, waterfall), etc., [1,6]. The navigation and control of the airspace around them are essential for all these applications.

1.1. Research Motivation

Entities in an IoD-based network, such as drones, servers, and users, communicate through an open channel, i.e., the Internet. As a result, there may be certain concerns regarding the confidentiality and safety of the information that is communicated over IoD. It is possible that it could be susceptible to attacks such as message replaying, impersonation, man-in-the-middle (MiTM), the physically compromising a drone, malware propagation, credential leakage, disclosure of sensitive data, and other similar attacks [7,8]. Thus, we require some security frameworks to protect the IoD networks from the various attacks that could be launched against them [9,10]. Data are stored within the blocks of a blockchain, which is a sort of distributed ledger technology. Blockchains are used in cryptocurrency transactions. These can safely process and store the data, and these guard the data against any attack that could involve data disclosure or data change [11,12]. The information kept in the blockchain can be accessed and used for various purposes, including the prediction of certain occurrences (for example, the traffic situation in a city or the weather forecast). Various approaches to data analysis based on machine learning can be utilized for undertakings of this nature [8]. Hence, a similar approach has been followed in the proposed scheme.

The following section will detail the research contributions made by this paper.

1.2. Research Contributions

The following is a list of the research contributions that this work makes:

- A secure blockchain-enabled authentication key management framework with big data analytics for drones in networks beyond 5G applications is proposed. In short, we call it SBBDA-IoD.
- The resilience of SBBDA-IoD against multiple attacks is demonstrated through a security analysis.
- The Scyther tool is used to perform a formal test of SBBDA-IoD's security, confirming the system's resistance to a variety of cyber-attacks.
- The comparative analysis found that SBBDA-IoD outperformed the other schemes by a significant margin.
- A real-world implementation of SBBDA-IoD is shown to evaluate its effect on several measures of performance.

2. Related Work

In this section, we discuss some of the existing authentication and key agreement methods and provide specifics regarding these methods.

Ali et al. [13] introduced an improved method known as a temporal credential-based anonymous lightweight authentication scheme (iTCALAS). This made use of the lightweight symmetric key primitives in conjunction with temporal credentials. The suggested method, while retaining its lightweight nature, provides security against a wide variety of previously recognized risks, such as traceability and stolen verifiers, while at the same time maintaining its inherent simplicity. The extended scalability of the iTCALAS that has been presented also allows it to function in an IoD environment with many flying zones or clusters. The authentication mechanisms that are going to be used for the secure commu-

nication of unmanned aerial vehicles (UAVs) were discussed by Rodrigues et al. [14]. They investigated and compared two authentication algorithms designed for wireless sensor networks (WSNs) and adapted for use with UAVs. The tests were carried out by examining the amount of time spent executing security-related activities such as hash tables and elliptic curve operations. Ever [15] has shown a safe authentication framework for mobile sinks, which could be utilized in applications for the Internet of Drones. The UAVs that had the potential to operate as mobile sinks were taken into consideration. The work that had already been performed on the authentication of the WSN-UAV environment was expanded. It was stated that there was a secure authentication framework that makes use of elliptic-curve cryptosystems. The aforementioned structure was put through a series of tests to establish whether or not it was resistant to substantial and well-known conceivable attacks. These attacks included those that targeted data secrecy, mutual authentication, password guessing, and key impersonation, among other things. Bera et al. [16] came up with an idea for a technique of access control that may be used in the Internet of Drones (IoD) setting for the purpose of identifying and mitigating the effects of unauthorized UAVs. They have used blockchain technology in their scheme. The transactional data were recorded on a private blockchain that was legitimate and authentic in every way. These data included the standard secure data that were transferred from a drone to the ground station server and the anomalous (suspected) data utilized to detect unauthorized UAVs that were stored over the private blockchain. These data were collected and transmitted by the ground station server.

Yazdinejad et al. [11] proposed a secure authentication model intended to use blockchain technology. The approach was intended to be used by drones in smart cities. The strategy ensured the fewest possible delays in the process. In a network of drones, they created a zone-based architecture and used a tailored decentralized consensus mechanism for drones in a smart city called "drone-based delegated proof of stake (DDPOS)". Both of these technologies are referred to as drone-based delegated proof of stake. When utilizing this strategy, drones did not need to go through the re-authentication process when moving between zones. Singh et al. [12] also addressed the development of the Internet of Drones as well as the industrial applications of this emerging technology. The development of this technology has brought about major worries, one of which has always been related to the unmanned robots' level of security. As a result, they brought attention to the most important security concerns and then suggested using cutting-edge blockchain technology as the most important answer to these concerns. Feng et al. [17] suggested a solution for blockchain-based cross-domain authentication that was designed for the intelligent Internet of Drones with 5G. This approach was developed with the intention of overcoming the limitations that were discussed earlier. Their technique relied on a large number of signatures, each of which was established by means of threshold sharing; this allowed them to successfully build an identity federation for collaborative domains. Because of this, they were able to facilitate joining and leaving domains. Utilizing smart contracts as a means of authentication allowed for reliable communication between devices that operated in many domains. A blockchain-based drone delivery system (GaRuDa system) was proposed by Gupta et al. [7]. This system might be utilized for applications that are linked to Healthcare 5.0. Their plan combined the Internet of Things and blockchain technology by way of an Internet that was enabled with 5G capabilities to facilitate the low-latency and responsive distribution of medical supplies that could be chronologically monitored and tracked among many stakeholders. In addition, this distribution of medical supplies could be monitored and tracked in real-time. Bera et al. [8] offered a security architecture for safe communication in IoD that was supported by smart contracts based on blockchain technology and was envisioned using artificial intelligence (AI). The security analysis indicated that the framework being presented was safe against the several different sorts of attacks that may be carried out against it.

Lwin et al. [18] explored the creation of a city geographic dashboard, which had the ability to collect, exchange, and visualize geographic data that were gathered from satellites,

Internet of Things devices (i.e., drones), and other types of big data. Abualigah et al. [19] provided a complete examination of the Internet of Things and its applications, deployments, and integrations. The Internet of Things applications, cloud and fog computing frameworks, unmanned aerial vehicles (UAVs), wireless sensor networks (WSNs), mobile computing, and business models were the key areas of focus for them. Gharibi et al. [1] offered a theoretical framework as an example for the construction of the IoD. They also determined the criteria that an IoD system must meet based on the architecture of the system in order for it to be regarded as successful.

Pu et al. [20] suggested a technique for user authentication and key agreement that was simple to implement and kept users' personal information private. They developed a physical unclonable function (PUF) and chaotic system in order to allow mutual authentication and establish a secure session key between the various communication participants. This was accomplished through the use of cryptography. The goal of this study, which was conducted by Yahuza et al. [6], was to conduct an analysis of recent advancements in the privacy and security issues that were related to IoD. They looked into the different types of drones to determine how much risk they posed to privacy and safety. After that, they discussed the importance of a secure architecture for IoD and proposed one. In addition to this, they provided a full taxonomy of attacks, which was possible in IoD systems.

Krichen et al. [21] examined the current state-of-the-art formal methods that have been applied to the definition and verification of smart contracts. This was performed with the intention of reducing the likelihood that caused errors and bugs. It avoided any costs that might have been caused as a result. In addition, they have identified a number of difficulties as well as potential guidelines for future research in relation to this new research tonic. Abdellatif and Brousmiche et al. [22] developed a formal modeling approach to verify the behavior of a smart contract within the environment in which it could be executed. They tested their formalism by applying it to a real-world example of a smart contract and analyzing the breaches it contained using a statical model checking approach.

Most of the schemes discussed here lack important functionality features and are vulnerable to various attacks. The blockchain-enabled mechanism can also help improve the stored data's security, which has been utilized for secure big data analytics. Hence, in this paper, we focus on the design of a secure blockchain-enabled authentication key management framework with big data analytics applicable for networks beyond 5G applications. The assessment of the closely related existing schemes is given in Table 1.

Table 1. Assessment of the closely related existing schemes.

Scheme	Technique Used	Features	Disadvantages and Security Flaws
Ali et al. [13]	A temporal credential-based anonymous lightweight authentication scheme (iTCALAS)	Introduced a temporal credential-based anonymous lightweight authentication scheme (iTCALAS) via lightweight symmetric key primitives. The extended scalability of the iTCALAS that has been presented also allows it to function in an IoD environment with many flying zones or clusters.	It did not have important security and functionality features, i.e., the presence of formal security verification using Scyther tool, support for the blockchain-based solution, support for anonymity and untraceability, and support for big data analytics. Moreover, it was vulnerable to ephemeral secret leakage (ESL) attack under the CK-adversary model.
Rodrigues et al. [14]	Authentication methods for UAV	They investigated and compared two authentication algorithms designed for WSNs and adapted for use with UAVs. The tests were carried out by examining the amount of time spent executing security-related activities such as hash tables and elliptic curve operations	It did not have important security and functionality features, i.e., the presence of formal security verification using the Scyther tool, the presence of the dynamic drone/device addition phase, support for the blockchain-based solution, and support for big data analytics. Moreover, it was vulnerable to ESL attack under the CK-adversary model.

Table 1. *Cont.*

Scheme	Technique Used	Features	Disadvantages and Security Flaws
Ever [15]	Secure authentication scheme	A safe authentication framework for mobile sinks has been shown which could be utilized in applications for the Internet of Drones.	It did not have important security and functionality features, i.e., the presence of formal security verification using Scyther tool, support dynamic drone/device addition phase, support for the blockchain-based solution, support for anonymity and untraceability, and support for big data analytics. Moreover, it was vulnerable to ESL attack under the CK-adversary model.
Bera et al. [16]	Private blockchain-based access control mechanism	They presented a technique for access control that may be used in an Internet of Drones (IoD) setting for the purpose of identifying and mitigating the effects of unauthorized UAVs. The transactional data were recorded on a private blockchain that was legitimate and authentic in every way.	It did not have support for the important big data analytics phase.

3. System Model

The system models, i.e., the network model (overall architecture) of the proposed SBBDA-IoD and its associated threat model, are discussed below.

3.1. Network Model

In Figure 1, we see the network representation of SBBDA-IoD. This network includes drones, ground-based servers, cloud-based servers, and control rooms. Drones collect data as they fly over designated areas (for city monitoring, smart farming, drug delivery, etc.) and transmit it to the ground station servers. The drone data are received by the ground station servers, where they are processed. The data are transmitted from the ground station servers to the connected cloud servers. The communications among the drones, ground station servers, and cloud servers happen through the open channel (i.e., the Internet), on which various attacks are possible. In order to estimate the traffic on a city street, determine the likelihood of a security attack, etc., the data are saved and analyzed on cloud servers. As discussed earlier, numerous cyber-attacks can target the numerous forms of communication that occur among the aforementioned entities. To protect against these attacks and the data being communicated, we need some sort of security mechanism (such as authentication, access control, or key management). In this paper, we present a method for both drone-to-drone and drone-to-ground station servers to successfully establish mutual authentication and session keys [16]. First, there will be secure mutual authentication among the drones, ground station servers, and cloud servers; then, they establish session keys for their secure data transmission. The control room also houses the network registration authority, which is responsible for registering the various nodes that make up the network (drones, ground station servers, and cloud nodes). Cloud servers are the resource-rich entities (having high communication, computation, and storage capabilities) of the network and are the building blocks of a peer-to-peer cloud server (P2PCS) network. Cloud servers handle the most crucial operations, such as implementing blockchain and analyzing massive amounts of data [11,12]. This means that cloud servers are the semi-trusted entities of the network. It is assumed that the network's registration authority has not been compromised because it is the trusted entity of the network. In the SBBDA-IoD, we can have multiple control rooms along with multiple registration authorities, which can be organized as per the network's size and requirements.

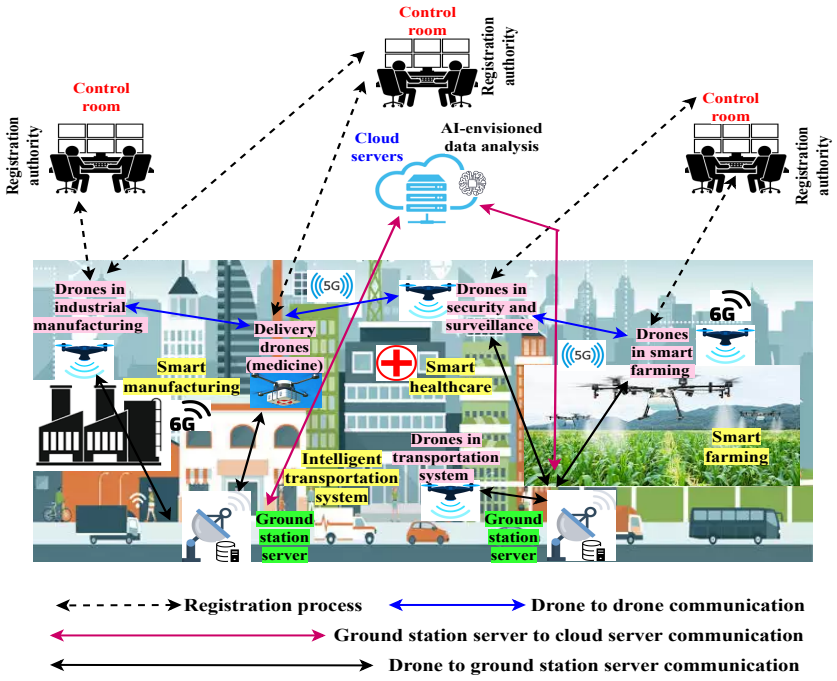

Figure 1. Network model of the proposed SBBDA-IoD.

3.2. Threat Model

The Dolev–Yao (DY) threat model, which at the present time is thought of as the standard de facto, was used in the design of the SBBDA-IoD [23]. According to the DY model, two separate entities can begin communicating with one another across an unsecured network (for instance, over the Internet). Unreliability exists at the endpoint entities, which include things like drones, ground station servers, and cloud servers. An adversary $\mathcal{A}$, whether active or passive, may read, change, or delete the communications that are sent across a network that is not protected from outside interference. Some of the potential attacks that the proposed SBBDA-IoD takes care of include replay attacks, man-in-the-middle (MiTM) attacks, impersonation attacks, privileged insider attacks, stolen verifier attacks, physical drone capture attacks, ephemeral secret leakage (ESL) attacks, secret data leakage attacks, etc. Another significant adversary model established by Canetti and Krawczyk (CK) is one we used in the SBBDA-IoD as well [24]. At this point, $\mathcal{A}$ can use all of the DY model's capabilities. In addition, $\mathcal{A}$ has the ability to acquire the session states, which are the credentials and session keys associated with a particular session. $\mathcal{A}$ can manually grab some drones by utilizing a technique that involves complex power analysis and then access the information that is stored in the memory of those devices [25]. The information gathered can also be used to start further operations and launch malicious attacks, such as establishing secret session keys and credentials, conducting the privileged-insider attack, data replaying, conducting the drone physical capture attack, data replaying, conducting the privileged-insider attack, conducting the malware injecting and scripting attacks, conducting the DoS attack, impersonation attack, credentials leakage, and man-in-the-middle (MiTM) attacks. Attacks with malware injection can take many forms, including spyware attacks, rootkit attacks, ransomware attacks, the insertion of Trojan horses, and the launch of virus and worm attacks. The ground station servers are the entities within the network that are only partially trusted, and they are also deployed with some degree of physical security (such as a locking system or security guards) [26]. As a result, there is no way that its physical integrity may be compromised via physical stealing. The cloud servers are also assumed to be the trusted network entities, and they are responsible for the data analysis. The DY model does not account for all possible forms of attack, such as the unauthorized disclosure of session states and the unauthorized computation of session

keys. The CK-adversary model takes into account attacks of this nature. As a result, in the design of the SBBDA-IoD, we considered both the CK-adversary model and the DY model.

4. The Proposed Scheme: SBBDA-IoD

In this section, we provide the details of the proposed SBBDA-IoD. Various notations used in this paper are listed in Table 2. The SBBDA-IoD is divided into several phases, i.e., "registration phase, authentication, and key establishment phase, dynamic device addition phase, key management phase, blockchain implementation phase, and big data analytics phase". Here, it is important to mention that the secure mutual authentications and key establishment phases are executed between a drone and the other legitimate drone and between the drone and the ground station server. These phases are elaborated as follows.

Table 2. Notations used in the SBBDA-IoD.

Notation	Meaning
DR_i and DR_j	ith and jth drones
GSS_k	kth ground station server
CS_l	lth cloud server
CRA	Control room's registration authority
ID_{CRA}, RID_{CRA}, and k_{CRA}	Identity, pseudo-identity, and secret key of CRA
ID_{DR_i}, RID_{DR_i}, and k_{DR_i}	Identity, pseudo-identity, and secret key of DR_i
TOT_{DR_i}	Temporary one-time identity of DR_i
RIN_{DR_i}	A pseudo-identification number of DR_i
ERP_{DR_i}	Essential registration parameter of DR_i
$E_q(u, v)$	A non-singular elliptic curve
G	A base point in $E_q(u, v)$
n_{DR_i}	A secret key number of DR_i
$N_{DR_i} = n_{DR_i}.G$	Corresponding public parameter of n_{DR_i}
ID_{GSS_k} and RID_{GSS_k}	Identity and pseudo-identity of GSS_k
k_{GSS_k}	Secret key of GSS_k
$Q_{GSS_k} = k_{GSS_k}.G$	Public key of GSS_k
ERP_{GSS_k}	Essential registration parameter of GSS_k
ID_{CS_l} and RID_{CS_l}	Identity and pseudo-identity of CS_l
k_{CS_l}	Secret key of CS_l
$Q_{CS_l} = k_{CS_l}.G$	Public key of CS_l
$\mathcal{A}$	An adversary

4.1. Registration Phase

In this phase, the control room's registration authority CRA performs the registration of other entities, i.e., drones, ground station servers, and cloud servers. The steps are given below.

4.1.1. Registration of Drones

The control room's registration authority, CRA, registers a drone as per the following steps.

- **RGDR1:** The CRA generates its identity as ID_{CRA} and secret key as k_{CRA}. It then computes its pseudo-identity as $RID_{CRA} = h(ID_{CRA}||\, k_{CRA})$. Then, CRA generates identity for drone DR_i as ID_{DR_i} and secret key as k_{DR_i}. CRA then computes the pseudo-identity of DR_i as $RID_{DR_i} = h(ID_{DR_i}||\, k_{DR_i}||\, RID_{CRA})$, temporary one-time identity as TOT_{DR_i} and an essential registration parameter as $ERP_{DR_i} = h(ID_{DR_i}||\, RTS_{DR_i}||\, k_{DR_i}||\, RID_{CRA})$, where RTS_{DR_i} is the registration timestamp value of DR_i.

- **RGDR2:** CRA again generates a pseudo-identification number for DR_i as RIN_{DR_i}. After that, "CRA selects a non-singular elliptic curve over a finite field" as given below. Suppose there are "2 constants $u \in Z_q$ and $v \in Z_q$, where $Z_q = \{0, 1, \ldots, q-1\}$ and $q > 3$ should be a prime". Again, RA chooses "a non-singular elliptic curve $E_q(u, v)$: $y^2 = x^3 + ux + v$ over the finite field $GF(q)$". For instance, "$4u^3 + 27v^2 \neq 0 \pmod q$ with a point at infinity or zero point $\mathcal{O}$". Let G be a base point in $E_q(u, v)$ with a similar big order like q. CRA then generates a secret key number of DR_i as n_{DR_i} and the associated public parameter as $N_{DR_i} = n_{DR_i}.G$. Finally, CRA stores $\{RIN_{DR_i}, RID_{DR_i}, TOT_{DR_i}, (n_{DR_i}, N_{DR_i}), ERP_{DR_i}, h(\cdot), E_q(u, v), G\}$ in the memory of DR_i. The drone DR_i can then be dispatched to the designated area for use.
- **RGDR3:** In a similar way, the registration of DR_j is performed. Then, DR_j contains values like, $\{RIN_{DR_j}, RID_{DR_j}, TOT_{DR_j}, (n_{DR_j}, N_{DR_j}), ERP_{DR_j}, h(\cdot), E_q(u, v), G\}$ in its memory.

4.1.2. Registration of Ground Station Servers

Using the following steps, the control room's registration authority CRA registers a ground station server GSS_k.

- **RGGS1:** For the registration of GSS_k, CRA first generates its identity as ID_{GSS_k} and secret key as k_{GSS_k}, it then computes the pseudo-identity of GSS_k as $RID_{GSS_k} = h(ID_{GSS_k} || k_{GSS_k} || RID_{CRA})$. CRA also computes the public key of GSS_k as $Q_{GSS_k} = k_{GSS_k}.G$. Furthermore, CRA computes its essential registration parameter as $ERP_{GSS_k} = h(ID_{GSS_k} || RTS_{GSS_k} || k_{GSS_k} || RID_{CRA})$, where RTS_{GSS_k} is the registration timestamp value of GSS_k. CRA also securely shares the registration information of drones, i.e., TOT_{DR_i} and RID_{DR_i} with GSS_k by making use of shared master secret key $MK_{CRA-GSS}$.
- **RGGS2:** Finally, CRA stores values like, $\{(TOT_{DR_i}, RID_{DR_i}) | \, i = 1, 2, \cdots, num_{DR}, RID_{GSS_k}, ERP_{GSS_k}, (k_{GSS_k}, Q_{GSS_k}), E_q(u, v), G, h(\cdot)\}$ in the database of GSS_k.

4.1.3. Registration of Cloud Servers

The control room's registration authority CRA registers a cloud server CS_l using the following steps.

- **RGCS1:** For the registration of CS_l, CRA first generates its identity as ID_{CS_l} and secret key as k_{CS_l}. After that, CRA produces the pseudo-identity of CS_l as $RID_{CS_l} = h(ID_{CS_l} || k_{CS_l} || RID_{CRA})$. CRA also computes the public key of CS_l as $Q_{CS_l} = k_{CS_l}.G$.
- **RGCS2:** Finally, CRA stores values such as $\{RID_{CS_l}, (k_{CS_l}, Q_{CS_l}), G, h(\cdot)\}$ in the database of CS_l.

4.2. Authentication and Key Establishment Phase

Secure mutual authentications and key establishment between drones and between drones and the ground station servers necessitate this step. Detailed explanations of each stage are given as follows.

4.2.1. Authentication and Key Establishment between Drone DR_i and Drone DR_j

The following describes the authentication and key establishment between drones DR_i and DR_j.

- **AKADD1:** Before starting the process of authentication and key establishment, the drones DR_i and DR_j share their pseudo-identification number with each other in a secure way. For example, for this task, DR_i can send message $MM_1 = E_{N_{DR_j}}(RIN_{DR_i})$ to DR_j. DR_j can decrypt and read the value RIN_{DR_i} as $D_{n_{DR_j}}(MM_1) = RIN_{DR_i}$. In a similar way, DR_i can obtain the value of RIN_{DR_j} from DR_j in a secure way. Furthermore, DR_i produces a random secret value rv_{DR_i} and a fresh timestamp value T_1. Then, DR_i computes $M_1 = h(rv_{DR_i} || RID_{DR_i}) \oplus h(RIN_{DR_i} || T_1)$ and $M_2 = h(h(rv_{DR_i} ||$

$RID_{DR_i})|| N_{DR_i}|| N_{DR_j}|| T_1)$. After calculating these values DR_i sends the message $msg_1 = \{M_1, M_2, T_1\}$ to DR_j through an insecure channel.

- **AKADD2:** When DR_j receives msg_1 from DR_i, it first verifies the correctness of T_1 by solving the equation $|T_1 - T_1^*| \leq \Delta T$, where ΔT is the maximum transmission delay and T_1^* is the time at which msg_1 was received.
  Then, DR_j computes $h(rv_{DR_i}|| RID_{DR_i}) = M_1 \oplus h(RIN_{DR_i}|| T_1)$ and $M_2' = h(h(rv_{DR_i}|| RID_{DR_i})|| N_{DR_i}|| N_{DR_j}|| T_1)$. DR_j then checks $M_2' = M_2$? If it matches, then DR_i is authenticated with Dr_j. Furthermore, DR_j generates a random secret value rv_{DR_j} and a fresh timestamp value T_2. Again, DR_j computes $M_3 = h(rv_{DR_j}|| RID_{DR_j}) \oplus h(RID_{DR_j}|| T_2)$. After these calculations, DR_j computes its session key as $SK_{DR_j,DR_i} = h(h(rv_{DR_i}|| RID_{DR_i})|| h(rv_{DR_j}|| RID_{DR_j})|| RIN_{DR_i}|| RIN_{DR_j}|| N_{DR_i}|| N_{DR_j}|| T_1|| T_2)$ and another important parameter as $M_4 = h(SK_{DR_j,DR_i}|| RIN_{DR_i}|| RIN_{DR_j}|| T_1|| T_2)$. DR_j then sends message $msg_2 = \{M_3, M_4, T_2\}$ to DR_i through an open insecure channel.
- **AKADD3:** When DR_i receives msg_2 from DR_j, it first verifies the correctness of T_2 by solving the equation $|T_2 - T_2^*| \leq \Delta T$, where ΔT is the maximum transmission delay and T_2^* is the time at which msg_2 was received. DR_i then computes $h(rv_{DR_j}|| RID_{DR_j}) = M_3 \oplus h(RID_{DR_j}|| T_2)$ and session key as $SK_{DR_i,DR_j} = h(h(rv_{DR_i}|| RID_{DR_i})|| h(rv_{DR_j}|| RID_{DR_j})|| RIN_{DR_i}|| RIN_{DR_j}|| N_{DR_i}|| N_{DR_j}|| T_1|| T_2)$ and $M_4' = h(SK_{DR_j,DR_i}|| RIN_{DR_i}|| RIN_{DR_j}|| T_1|| T_2)$. DR_i checks $M_4' = M_4$? If it matches, then DR_j is authenticated with DR_i, and the computed session key by DR_i is correct. Again DR_i generates another fresh timestamp value T_3 and computes $M_5 = h(SK_{DR_i,DR_j}|| T_3)$ and sends message $msg_3 = \{M_5, T_3\}$ to DR_i via open channel.
- **AKADD4:** When DR_j receives msg_3 from DR_i, it first verifies the correctness of T_3 by solving the equation $|T_3 - T_3^*| \leq \Delta T$, where ΔT is the maximum transmission delay and T_3^* is the time at which msg_3 was received. Furthermore, DR_j computes $M_5' = h(SK_{DR_j,DR_i}|| T_3)$ and checks $M_5' = M_5$? If it matches, then DR_j assumes that the session key computed by DR_i is correct. Eventually, DR_i and DR_j establish session key $SK_{DR_i,DR_j} = (SK_{DR_j,DR_i})$ for their secure communication.

4.2.2. Authentication and Key Establishment between Drone DR_i and Ground Station Server GSS_k

The authentication and key establishment details between the drone DR_i and the ground station server GSS_k are given below.

- **AKADG1:** DR_i initiates the process with the generation of a random secret parameter (i.e., a variable) rs_1 and fresh timestamp value t_1. It then computes $m_1 = h(rs_1|| ERP_{DR_i}|| t_1) \oplus h(RID_{DR_i}|| t_1)$ and $m_2 = h(h(rs_1|| ERP_{DR_i}|| t_1)|| RID_{DR_i}|| t_1)$. After that, DR_i sends the message $MSG_1 = \{TOT_{DR_i}, m_1, m_2, t_1\}$ to GSS_k via open channel.
- **AKADG2:** When GSS_k receives MSG_1 from DR_i, it first verifies the correctness of t_1 by solving the equation $|t_1 - t_1^*| \leq \Delta T$, where ΔT is the maximum transmission delay and t_1^* is the time at which MSG_1 was received. Then, GSS_k fetches RID_{DR_i}, corresponding to the received TOT_{DR_i}. It then computes $h(rs_1|| ERP_{DR_i}|| t_1) = m_1 \oplus h(RID_{DR_i}|| t_1)$ and $m_2' = h(h(rs_1|| ERP_{DR_i}|| t_1)|| RID_{DR_i}|| t_1)$. GSS_k then checks $m_2' = m_2$? If it matches, then DR_i is authenticated with GSS_k. Furthermore, GSS_k generates a random secret parameter (i.e., a variable) rs_2 and the fresh timestamp value t_2. It again computes $m_3 = h(rs_2|| ERP_{GSS_k}|| t_2) \oplus h(RID_{DR_i}|| t_2)$ and session key as $SK_{GSS_k,DR_i} = h(h(rs_1|| ERP_{DR_i}|| t_1)||h(rs_2|| ERP_{GSS_k}|| t_2)|| RID_{DR_i}|| t_1|| t_2)$. After that, GSS_k computes $m_4 = h(SK_{DR_i,GSS_k}|| h(rs_2|| ERP_{GSS_k}|| t_2)|| RID_{DR_i}|| t_2)$ and a new temporary one time identity as $TOT_{DR_i}^{new}$. It again computes $m_5 = TOT_{DR_i}^{new} \oplus h(h(rs_1|| ERP_{DR_i}|| t_1)|| RID_{DR_i}|| t_2)$. After that GSS_k sends the message $MSG_2 = \{m_3, m_4, m_5, t_2\}$ to DR_i through an open (insecure) medium.

- **AKADG3:** When DR_i receives MSG_2 from GSS_k, it first verifies the correctness of t_2 by solving the equation $|t_2 - t_2^*| \leq \Delta T$, where ΔT is the maximum transmission delay and t_2^* is the time at which MSG_2 was received. It again computes $h(rs_2 || ERP_{GSS_k} || t_2) = m_3 \oplus h(RID_{DR_i} || t_2)$ and session key as $SK_{DR_i,GSS_k} = h(h(rs_1 || ERP_{DR_i} || t_1) || h(rs_2 || ERP_{GSS_k} || t_2) || RID_{DR_i} || t_1 || t_2)$. After that, DR_i computes $m_4' = h(SK_{DR_i,GSS_k} || h(rs_2 || ERP_{GSS_k} || t_2) || RID_{DR_i} || t_2)$ and checks $m_4' = m_4$? If it matches, then GSS_k is authenticated with DR_i, and the session key computed by DR_i is correct. Again, DR_i computes its new temporary one-time identity as $TOT_{DR_i}^{new} = m_5 \oplus h(h(rs_1 || ERP_{DR_i} || t_1) || RID_{DR_i} || t_2)$. DR_i generates another fresh timestamp value as t_3 and computes $m_{SKV} = h(SK_{DR_i,GSS_k} || t_3)$. After that, DR_i sends message $MSG_3 = \{m_{SKV}, t_3\}$ to GSS_k through the open (insecure) medium.
- **AKADG4:** When GSS_k receives MSG_3 from DR_i, it first verifies the correctness of t_3 by solving the equation $|t_3 - t_3^*| \leq \Delta T$, where ΔT is the maximum transmission delay and t_3^* is the time at which MSG_3 was received. It again computes $m_{SKV}' = h(SK_{GSS_k,DR_i} || t_3)$ and checks $m_{SKV}' = m_{SKV}$? If they match, GSS_k assumes that the session key that DR_i came up with is right. In this case, the session key verification works. Eventually, DR_i and GSS_k agree on session key $SK_{DR_i,GSS_k} = (SK_{GSS_k,DR_i})$ so that they can send data securely.

4.3. Dynamic Device Addition Phase

This step must be performed before new drones can be added to the network. As a drone can sometimes malfunction or have its work stopped. We need to take the following steps for a new drone to be added to the network.

- **DDDR1:** CRA generates an identity for drone DR_i^ν as $ID_{DR_i}^\nu$ and secret key as $k_{DR_i}^\nu$. CRA then computes the pseudo-identity of DR_i^ν as $RID_{DR_i}^\nu = h(ID_{DR_i}^\nu || k_{DR_i}^\nu || RID_{CRA})$, a temporary one-time identity as $TOT_{DR_i}^\nu$ and essential registration parameter as $ERP_{DR_i}^\nu = h(ID_{DR_i}^\nu || RTS_{DR_i}^\nu || k_{DR_i}^\nu || RID_{CRA})$, where $RTS_{DR_i}^\nu$ is the registration timestamp value of DR_i^ν.
- **DDDR2:** CRA again generates a pseudo-identification number for DR_i^ν as $RIN_{DR_i}^\nu$. CRA then generates a secret key number of DR_i^ν as $n_{DR_i}^\nu$ and its corresponding public parameter as $N_{DR_i}^\nu = n_{DR_i}^\nu.G$. Finally, CRA stores $\{RIN_{DR_i}^\nu, RID_{DR_i}^\nu, TOT_{DR_i}^\nu, (n_{DR_i}^\nu, N_{DR_i}^\nu), ERP_{DR_i}^\nu, h(\cdot), E_q(u,v), G\}$ in the memory of DR_i^ν. Then, DR_i^ν is deployed in the assigned zone as per the requirement. CRA also shares the registration information of drones, i.e., $TOT_{DR_i}^\nu$ and $RID_{DR_i}^\nu$ with the existing GSS_k in a secure way through shared master secret key $MK_{CRA-GSS}$.

4.4. Key Management Phase between GSS_k and CS_l

For the secure transmission of their data, GSS_k and CS_l can use their secret and public key pairs, i.e., (k_{GSS_k}, Q_{GSS_k}) and (k_{CS_l}, Q_{CS_l}). For example, when GSS_k has to send its data DT_{GSS_k} to CS_l, then GSS_k can perform encryption over it through Q_{CS_l}, i.e., $MSG_{GC_1} = E_{CS_l}(DT_{GSS_k})$. Upon the arrival of MSG_{GC_1}, CS_l can perform decryption as $D_{k_{CS_l}}(MSG_{GC_1})$ and read out the value of DT_{GSS_k} in a secure way. Here, it is important to mention that, to perform the above-discussed encryption/decryption, the standard ECC algorithm can be used on both GSS_k and CS_l sides as these are resource-rich devices. Both GSS_k and CS_l have good computation and communication capabilities.

4.5. Blockchain Implementation Phase

This phase is used to implement the blockchain of the drone-related data. When a drone, say DR_i, sends some information to the connected GSS_k, that GSS_k creates a partial block PB_{GSS_k}. The partial block includes information like an owner of the block (i.e., OW_{GSS_k}), the public key of the owner (i.e., Q_{GSS_k}), and encrypted transactions (i.e., $TR_{X_i} | i = 1, 2, \cdots, num_{tx}$). Then, GSS_k sends the information of the partial block to the connected CS_l through the established session key SK_{GSS_k,CS_l}. After receiving PB_{GSS_k} from GSS_k, CS_l

creates the full block FB_{CS_l} from it. FB_{CS_l} contains information like the block's identity BID_{CS_l}, the hash of this block $HASH_{FB_{CS_l}}$, the hash of previous block $HASH_{FB_{CS_l}-1}$, the timestamp value $TS_{FB_{CS_l}}$, random nonce value $RN_{FB_{CS_l}}$, OW_{GSS_k}, Q_{GSS_k}, TR_{X_i}, and the signature of this block (i.e., $SG_{FB_{CS_l}}$ using a standard algorithm like "Elliptic Curve Digital Signature Algorithm (ECDSA))". Then, CS_l broadcasts it to its peer-to-peer cloud server network (P2PCS). Then, the miners (i.e., authorized cloud servers of the P2PCS network) execute a consensus process using some standard algorithm (i.e., practical byzantine fault tolerance (pBFT) [27,28]). After the steps of the consensus process have successfully been completed, the new block FB_{CS_l} is added to the blockchain BC_{DRN}. Using machine learning algorithms, the data saved in the blockchain can be used to make predictions and analyze things like traffic on a certain street in a city or the weather in a certain area.

4.6. Big Data Analytics Phase

This phase is used to perform big data analytics over the stored data in blockchain BC_{DRN}. For this task, the authorized cloud server, i.e., CS_l performs the standard steps of big data analytics [29]. The details of all steps are given below.

- **Secure data collection and processing:** Data collection takes on various forms throughout organizations. For example, the data are collected at the CS_l in a secure way through the established session key SK_{GSS_k,CS_l}, which is further processed and stored in the blockchain BC_{DRN}. Here, it is important to mention that the data stored in BC_{DRN} are protected against various information security-related attacks due to the inherent mechanism of blockchain.
- **Cleaning of data:** It is vital to clean the data to improve the findings and raise the bar for the data quality maintained in BC_{DRN}. In order to accomplish this, all of the data need to be presented appropriately, and any redundant or irrelevant material must either be removed or accounted for. Incorrect data can distort the picture and give the wrong impression, which ultimately leads to incorrect insights [29].
- **Secure data analysis:** This takes time to transform massive amounts of data into a form that is usable. When it is ready, advanced analytic techniques are able to turn massive amounts of data into insightful conclusions, i.e., the data maintained in BC_{DRN}. The following strategies can be used for analyzing huge data, i.e., "data mining, predictive analysis, and deep learning [30]". Here, data mining is the process of searching through huge datasets to find patterns and relationships. This is accomplished by locating outliers and forming data clusters. Furthermore, an organization's historical data are used in predictive analytics to create forecasts about the organization's future and detect emerging dangers and opportunities. Furthermore, deep learning is a type of earning method that imitates how humans learn by layering algorithms and combining artificial intelligence and machine learning to uncover patterns in the most complex and abstract data [29].

Remark 1 (User authentication process in SBBDA-IoD)**.** *The SBBDA-IoD includes a provision stating that a legitimate user may access the data of the network by following the procedures of a standard user authentication protocol (also known as the "secure key management and user authentication scheme" given in [31]). This provision is included in the document. Following the effective completion of each step in this protocol, both the legitimate user (i.e., U_m) and the cloud server (CS_l) will be able to generate a session key SK_{U_m,CS_l} in order to gain access to the data in a safe manner. Therefore, U_m is able to obtain the data of the network in a safe manner and can utilize it in accordance with their needs.*

Remark 2 (Secrecy of the data is maintained in SBBDA-IoD). *In the proposed SBBDA-IoD, the messages like* $msg_1 = \{M_1, M_2, T_1\}$, $msg_2 = \{M_3, M_4, T_2\}$ *and* $msg_3 = \{M_5, T_3\}$ *are transmitted between* DR_i *and* DR_j. *Similarly, messages like* $MSG_1 = \{TOT_{DR_i}, m_1, m_2, t_1\}$, $MSG_2 = \{m_3, m_4, m_5, t_2\}$, *and* $MSG_3 = \{m_{SKV}, t_3\}$ *are exchanged between* DR_i *and* GSS_k. *These messages are constructed through concatenation, bitwise XOR, and cryptographic one-way hash operations. Through these operations, we protect sensitive data during the transmission of these messages. Furthermore, the data which are stored over the cloud servers is maintained in the form of certain blocks of the blockchain. This blockchain is formed through a certain number of blocks, and each block contains sensitive data in the form of encrypted transactions (i.e.,* $TR_{X_i} | i = 1, 2, \cdots, num_{tx}$*). Therefore, the secrecy of the data are maintained both in transit as well as in storage.*

To provide the readers with a better understanding of the paper, we have provided a block diagram of the proposed SBBDA-IoD in Figure 2. This provides an illustration of all phases of the proposed SBBDA-IoD.

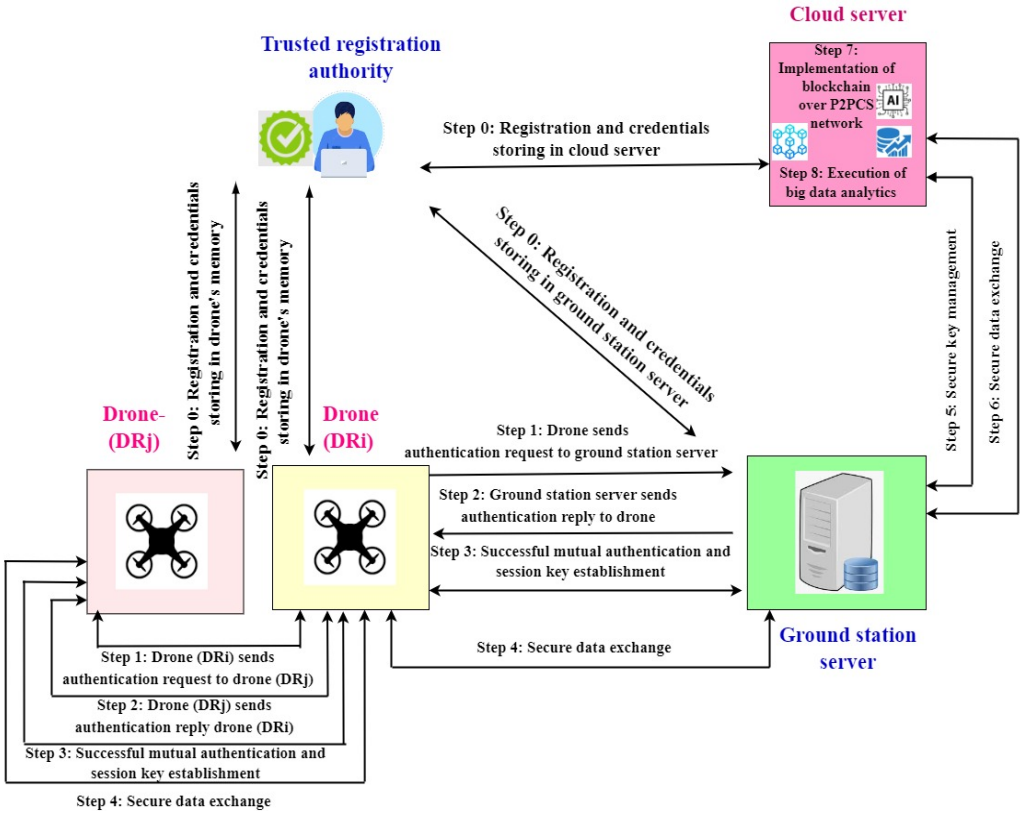

Figure 2. Block diagram of the proposed SBBDA-IoD.

5. Security Analysis of SBBDA-IoD

The security analysis of the SBBDA-IoD is covered here. The security analysis of the proposed framework has been conducted in an informal way through mathematical assumptions and equations and through formal security verification using the Scyther tool (Section 6). Through the conducted security analysis, it has been observed that the proposed framework is secured against various potential attacks, i.e., replay attack, man-in-the-middle (MiTM) attack, impersonation attacks, privileged insider attack, stolen verifier attack, physical drone capture attack, ephemeral secret leakage (ESL) attack, secret data leakage attack, etc. Herein, the specifics are laid down.

5.1. SBBDA-IoD Prevents the Replay Attack

Several distinct timestamp values, such as T_1, T_2, T_3, t_1, t_2, and t_3, are utilized and then validated on the opposite end of the communication. If the verification of the timestamp is successful, the recipient will accept the message; if it is not, the message will be denied. The SBBDA-IoD has the capability to prevent replay attacks thanks to the utilization of this condition checking. As a result, the SBBDA-IoD protocol is protected from replay attack.

5.2. SBBDA-IoD Prevents Man-in-the-Middle (MiTM) and Impersonation Attacks

We use a variety of proprietary factors, such as (k_{DR_i}, k_{CRA}, k_{GSS_k}, and k_{CS_l}), in the process of computing the messages that are being broadcast. $\mathcal{A}$ does not know these confidential values. Under those circumstances, $\mathcal{A}$ cannot make any changes to the delivered or received communications. In addition, $\mathcal{A}$ is unable to generate completely novel messages in the forms in which they were initially sent. Because of this, the SBBDA-IoD that is being described provides protection against man-in-the-middle (MiTM) attacks as well as impersonation attacks.

5.3. SBBDA-IoD Has Resilience against the Privileged Insider Attack

After registration, CRA deletes the secret values of the entities (k_{DR_i}, n_{DR_i}, k_{CRA}, K_{GSS_k}, RTS_{DR_i}, RTS_{GSS_k} and k_{CS_l}) from its database, making it inaccessible to the privileged insider user (i.e., $\mathcal{A}$) who plans to harm the entities (i.e., by some attacks). As a result, the SBBDA-IoD is protected from privileged attacks as well as other linked threats such as MiTM attacks, efforts to impersonate someone else, unauthorized session key computations, and so on. As a consequence, the SBBDA-IoD possesses the ability to reduce the impact of the privileged insider attack.

5.4. SBBDA-IoD Is Protected from Stolen Verifier Attack

In the secure section of the cloud server database, we maintain records of the parameters that various servers and devices register (i.e., their secret information). Multiple levels of security are used to secure that. Under these circumstances, $\mathcal{A}$ is unable to gain access to the entities' secret values. As long as this mechanism is in place, the AKM-BDSF cannot be attacked using the stolen verifier attack or other related attacks. Thus, the AKM-BDSF is secured against the stolen verifier attack.

5.5. SBBDA-IoD Prevents Physical Drone Capture Attack

The SBBDA-IoD prevents sensitive information being saved in its unencrypted form in the memory of the drones. In addition, if $\mathcal{A}$ physically steals a drone and then attempts to use a sophisticated power analysis attack to retrieve critical data from the drone's memory, this is a very dangerous scenario [25]. $\mathcal{A}$ would only be able to obtain the drone's session key and registration data under those conditions, but not the secret data of any other drones. Each session key is unique in the SBBDA-IoD. Since each one is computed using a different set of parameters, both of these are kept a secret. It is impossible to figure out the session key for other drones utilizing the deduced session key. As a direct consequence of this, the remaining sections of the transmission are kept secure. Hence, SBBDA-IoD is resistant to an attack that involves the physical acquisition of drones.

5.6. SBBDA-IoD Supports Anonymity and Untraceability Properties of Communication

In the SBBDA-IoD, no identifying information is transmitted in plaintext. This ensures that everyone's privacy is preserved. All of the information exchanged is derived from freshly generated timestamp values and randomly generated secret values. As a result of this mechanism's implementation, we obtain different messages in separate sessions. Hence, SBBDA-IoD boasts characteristics like anonymity and untraceability.

5.7. SBBDA-IoD Is Secured against Ephemeral Secret Leakage (ESL) Attack under the CK-Adversary Model

The mechanism that makes up the SBBDA-IoD is responsible for computing the session key by making use of both short-term information, such as random nonce metrics, and long-term information, such as secret keys and identities. In SBBDA-IoD, session key between DR_i and DR_j and DR_i and GSS_k are calculated as $SK_{DR_i,DR_j} = h(h(rv_{DR_i}||\ RID_{DR_i})||\ h(rv_{DR_j}||\ RID_{DR_j})||\ RIN_{DR_i}||\ RIN_{DR_j}||\ N_{DR_i}||\ N_{DR_j}||\ T_1||\ T_2)$ and $SK_{DR_i,GSS_k} = h(h(rs_1||\ ERP_{DR_i}||\ t_1)||h(rs_2||\ ERP_{GSS_k}||\ t_2)||\ RID_{DR_i}||\ t_1||\ t_2)$. The session keys accommodate both the random secrets (rv_{DR_i}, rv_{DR_j}, rs_1, and rs_2) as short-term secret values and secret keys (k_{DR_i}, k_{DR_j}, k_{CRA}, and k_{GSS_k} and several identities (RID_{DR_i}, RID_{DR_j}, and RID_{GSS_k}) as long-term secret values. A fresh session key is generated at the beginning of each session. In this particular scenario, $\mathcal{A}$ does not possess the necessary skills to unearth the long-term and short-term secrets, both of which are vital components for precisely establishing the value of the session key. As a direct consequence of this, an adversary will be unable to correctly guess the session key. Therefore, according to the CK-adversary model, SBBDA-IoD is resilient enough to withstand an ESL attack.

6. Formal Security Verification of the SBBDA-IoD Using Scyther Tool

This section discusses the formal security verification performed for the SBBDA-IoD. It is possible to utilize the Scyther tool in order to validate the formal security of the SBBDA-IoD [32–34]. It is an upgraded and more effective tool for judging, verifying, and analyzing the specified security protocol than other verification tools that are currently accessible, such as ProVerif and AVISPA. The most advanced cryptographic assumptions were used to develop Scyther. This ensures that an adversary is unable to decrypt the information in the given scheme if they do not have access to the secret key. It does this by imitating user-defined security protocols through the use of a language called security protocol descriptive language (SPDL). Each communication party (entity) is shown as a separate role within the context of the SPDL standard. These roles are able to carry out a variety of tasks, including sending and receiving of messages, the providing the necessary security claims, and the management of events. For example, "send" refers to the act of transmitting a message from one entity to another, while "recv" refers to the act of "receiving" a message from one entity to another [35].

The Scyther tool adheres to the guidelines established by the Dolev–Yao (DY) model in addition to nine additional adversarial models, including the e Canetti–Krawczyk's (eCK) model and the CK model. Scyther claims that the tests it offers can verify several aspects of security, including agreement, synchronization, poor agreement, and secrecy. In order to simulate the authentication and key agreement phase in the proposed system, we take into consideration the two fundamental responsibilities of DR (which refers to a drone) and GSS (which refers to a ground station server). After that, the SPDL code is utilized in order to put the suggested plan into action. Figures 3 and 4 exhibit the SPDL code snippets necessary to play the roles of a ground station server (GSS_k) and a drone (DR_i), respectively. In conclusion, the analysis and implementation results performed with the help of the Scyther tool are displayed in Figure 5 (under the claim, status, and comments items). Further investigation of the claims indicated that the SBBDA-IoD has protection under the claims covered in the preceding section. As a result, according to the findings of the formal security verification, which has been carried out, it is determined that the suggested scheme is protected against the myriad of possible attacks.

```
hashfunction h;
const xor:Function;
const cat:Function;
protocol drones(DR,GSS)
{
 role DR
 {
   fresh rs1:Nonce;
   const TOTDRi,TOTDRin,RIDDRi,ERPDRi,RIDGSSk,ERPGSSk,t1,t2,
   t3,rs1,rs2;
   var rs2:Nonce;
   macro m1=xor(h(cat(rs1,ERPDRi,t1)),h(cat(RIDDRi,t1)));
   macro m2=h(cat(h(cat(rs1,ERPDRi,t1)),RIDDRi,t1));
   macro SKDRiGSSk=h(h(cat(rs1,ERPDRi,t1)),h(cat(rs2,ERPGSSk,
   t2)),RIDDRi,t1,t2);
   macro mSKV=h(cat(SKDRiGSSk,t3));
   send_1(DR,GSS,cat(TOTDRi,m1,m2,t1));
   recv_2(GSS,DR,cat(xor(h(cat(rs2,ERPGSSk,t2)),h(cat(RIDDRi,t2)
   )),h(cat(h(h(cat(rs1,ERPDRi,t1)),h(cat(rs2,ERPGSSk,t2)),RIDDRi,
   t1,t2),h(rs2,ERPGSSk,t2),RIDDRi,t2)),xor((TOTDRin),h(cat(h(cat(
   rs1,ERPDRi,t1))),RIDDRi,t2)),t2));
   send_3(DR,GSS,cat(mSKV,t3));
   claim_DR1(DR,Secret,(rs1));
   claim_DR2(DR,Secret,(RIDDRi));
   claim_DR3(DR,Secret,(ERPDRi));
   claim_DR4(DR,Niagree);
   claim_DR5(DR,Nisynch);
   claim_DR86(DR,Secret,(SKDRiGSSk));
   claim_DR7(DR,Weakagree);
   claim_DR8(DR,Alive);
 }
```

Figure 3. SPDL snippet for the role of a drone DR_i.

```
 role GSS
 {
   fresh rs2:Nonce;
   const TOTDRi,TOTDRin,RIDDRi,ERPDRi,RIDGSSk,ERPGSSk,t1,t2,t3,
   rs1,rs2;
   var rs1:Nonce;
   macro m3=xor(h(cat(rs2,ERPGSSk,t2)),h(cat(RIDDRi,t2)));
   macro SKGSSkDRi=h(h(cat(rs1,ERPDRi,t1)),h(cat(rs2,ERPGSSk,
   t2)),RIDDRi,t1,t2);
   macro m4=h(cat(SKGSSkDRi,h(rs2,ERPGSSk,t2),RIDDRi,t2));
   macro m5=xor((TOTDRin),h(cat(h(cat(rs1,ERPDRi,t1))),RIDDRi,t2));
   recv_1(DR,GSS,cat(TOTDRi,xor(h(cat(rs1,ERPDRi,t1)),h(cat(RIDDRi,
   t1))),h(cat(h(cat(rs1,ERPDRi,t1)),RIDDRi,t1)),t1));
   send_2(GSS,DR,cat(m3,m4,m5,t2));
   recv_3(DR,GSS,cat(h(cat(SKGSSkDRi,t3)),t3));
   claim_GSS1(GSS,Secret,(rs2));
   claim_GSS2(GSS,Secret,(RIDGSSk));
   claim_GSS3(GSS,Secret,(ERPGSSk));
   claim_GSS4(GSS,Niagree);
   claim_GSS5(GSS,Nisynch);
   claim_GSS6(GSS,Secret,(SKGSSkDRi));
   claim_GSS7(GSS,Weakagree);
   claim_GSS8(GSS,Alive);
 }
}
```

Figure 4. SPDL snippet for the role of a ground station server GSS_k.

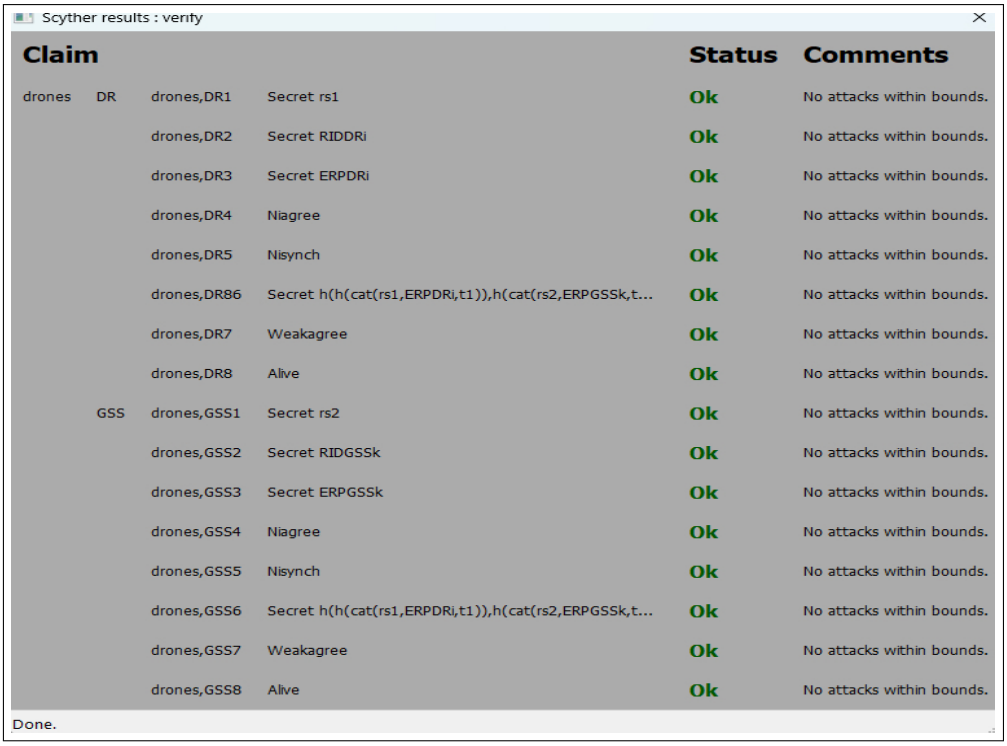

Figure 5. Results of security verification using Scyther tool.

7. Performance Comparison

In this section, we provide the details of conducted comparisons. We compare the SBBDA-IoD with relevant existing protocols, such as the schemes of Ali et al. [13], Rodrigues et al. [14], Ever [15], and Bera et al. [16]. The costs of communication, computation, and critical security and functionality characteristics were compared. The details are given below.

7.1. Comparison of Computation Costs

The computation costs of the SBBDA-IoD and the schemes of Ali et al. [13], Rodrigues et al. [14], Ever [15], and Bera et al. [16] are compared. For the comparison, we consider a few notations like T_h, T_{ecm}, T_{eca}, T_{exp}, T_{bp}, T_{fe}, $T_{senc/sdec}$, and T_{mtp} which are "one-way hash function using SHA-256 [25]", "elliptic curve multiplication", "elliptic curve addition", "modular exponentiation", "bilinear pairing", "fuzzy extractor operation" and "symmetric encryption/decryption using advanced encryption standard (AES)-128 [36]", and "map-to-point", respectively. To estimate the rough computation time (in milliseconds), we use the experimental results given in [37]: "$T_{ecm} \approx 13.405$ ms, $T_{eca} \approx 0.081$ ms, $T_h \approx 0.056$ ms, $T_{bp} \approx 32.713$ ms, $T_{exp} \approx 2.249$ ms". It is also considered that $T_{fe} \approx T_{ecm}$, $T_{senc/sdec} \approx T_h$ and $T_{mtp} \approx T_{exp}$. Table 3 provides the different computation cost values. As can be seen from the data presented in Table 3, the SBBDA-IoD has a significantly lower computation cost than the other schemes already in existence.

7.2. Comparison of Communication Costs

We use the following assumptions to calculate the communication costs of different schemes: "identity takes 160 bits", a "random number needs 160 bits", the "hash output when SHA-256 technique takes [38] is 256 bits", and a "timestamp needs 32 bits". For a point on an elliptic curve, there is an assumption $P = (P_x, P_y)$, where P_x and P_y are the x and y coordinates, needing $(160 + 160) = 320$ bits. This is considered as per the fact that "the security fulfills by a 160-bit elliptic curve cryptography (ECC) is almost same as that for a 1024-bit RSA-based public key cryptographic technique [39]". The various communication costs are broken down and compared in Table 4. The information shown

in Table 4 makes it abundantly clear that the SBBDA-IoD has fewer costs associated with communication than any of the other approaches that are already in use.

Table 3. Comparison of various computation cost values.

Protocol	Smart Device/ Drone	GSS/ Server
Ali et al. [13]	$18T_h + T_{fe} + T_{senc}$ $\approx$ 14.469 ms	$7T_h + 3T_{senc/sdec}$ $\approx$ 0.56 ms
Rodrigues et al. [14]	$9T_h + 6T_{ecm}$ $\approx$ 80.934 ms	$9T_h + 2T_{ecm}$ $\approx$ 27.314 ms
Ever [15]	$9T_h + 2T_{bp}$ $+ 2T_{mtp} + 3T_{ecm}$ $\approx$ 110.643 ms	$6T_h + 3T_{bp}$ $+ 2T_{mtp} + 3T_{ecm}$ $\approx$ 143.67 ms
Bera et al. [16]	$9T_h + 2T_{senc/sdec}$ $+2T_{ecm} + T_{eca}$ $\approx$ 27.507 ms	$9T_h + 2T_{senc/sdec}$ $+2T_{ecm} + T_{eca}$ $\approx$ 27.507 ms
SBBDA-IoD	$9T_h$ $\approx$ 0.504 ms	$7T_h$ $\approx$ 0.392 ms

Table 4. Comparison of communication costs.

Protocol	No. of Messages	Total Cost (in Bits)
Ali et al. [13]	3	3424
Rodrigues et al. [14]	4	3456
Ever [15]	6	5344
Bera et al. [16]	3	2368
SBBDA-IoD	3	1792

7.3. Comparison of Security and Functionality Features

In Table 5, we examine the various systems' levels of security as well as their functional capabilities. The information shown in Table 5 makes it abundantly clear that the SBBDA-IoD provides more functionality features in addition to higher levels of security than each of the existing schemes. Like most of the schemes, however, it does not have important features, i.e., formal security verification using the Scyther tool, ephemeral secret leakage (ESL) attack possibility, presence of dynamic drone/device addition, availability of blockchain-based solution, availability of anonymity and untraceability, and support for big data analytics.

Table 5. A juxtaposition of security and functionality components.

Feature	Ali et al. [13]	Rodrigues et al. [14]	Ever [15]	Bera et al. [16]	SBBDA-IoD
SnF_1	✓	✓	✓	✓	✓
SnF_2	✓	✓	✓	✓	✓
SnF_3	✓	✓	✓	✓	✓
SnF_4	✓	✓	✓	✓	✓
SnF_5	✓	✓	✓	✓	✓
SnF_6	✓	✓	✓	✓	✓
SnF_7	✓	✓	✓	✓	✓
SnF_8	✓	✓	✓	✓	✓

Table 5. *Cont.*

Feature	Ali et al. [13]	Rodrigues et al. [14]	Ever [15]	Bera et al. [16]	SBBDA-IoD
SnF_9	×	×	×	×	✓
SnF_{10}	×	×	×	✓	✓
SnF_{11}	✓	×	×	✓	✓
SnF_{12}	×	×	×	✓	✓
SnF_{13}	×	✓	✓	✓	✓
SnF_{14}	×	✓	×	✓	✓
SnF_{15}	×	×	×	×	✓

SnF_1: "replay attack"; SnF_2: "man-in-the-middle attack"; SnF_3: "mutual authentication"; SnF_4: "key agreement"; SnF_5: "device/drone impersonation attack"; SnF_6: "GSS/server impersonation attack"; SnF_7: "malicious device deployment attack"; SnF_8: "resilience against drone/device physical capture attack"; SnF_9: "formal security verification using Scyther tool"; SnF_{10}: "ephemeral secret leakage (ESL) attack under the CK-adversary model"; SnF_{11}: "support dynamic drone/device addition phase"; SnF_{12}: "support blockchain-based solution"; SnF_{13}: "free from design flaws"; SnF_{14}: "anonymity and untraceability"; SnF_{15}: "support big data analytics"; ✓: "a scheme is secure or it supports a functionality feature"; ×: "a scheme is insecure or it does not support a functionality feature"; *N/A*: "not applicable in a scheme".

8. Practical Implementation

In this section, we provide the details of the practical implementation of the proposed SBBDA-IoD. For the big data analytics procedure, we have taken the "SDOT Collisions All Years" dataset [40]. We then identified the chances of the collision of vehicles across various streets in a city. As we know, flying drones can also perform the same task and help us make predictions about the possibility of colliding with different vehicles. In this way, we can save people's lives and protect vehicles from damage.

8.1. Implementation Settings and Environment

The details of the various simulation parameters are given in Table 6. We used a 2X Intel(R) Xeon(R) processor with 2.20 GHz. The Google Colab environment was considered as the platform over Ubuntu 18.04.5 LTS operating system. The graphics processing unit (GPU) was 12 GB NVIDIA Tesla K80 along with the 13.34 GB random access memory (RAM). The number of cloud servers deployed was 2. The libraries like Shap, Tensorflow, SKLEARN, and Pandas were utilized. Furthermore, the "SDOT Collisions All Years" dataset [40] used utilized.

Table 6. Details of the simulation parameters.

Parameter	Value
Processor	2X Intel(R) Xeon(R) CPU @2.20 GHz
Platform used	Google Colab environment
Operating system	Ubuntu 18.04.5 LTS
GPU	12 GB NVIDIA Tesla K80
Random access memory (RAM) size	13.34 GB
Number of cloud servers deployed	2 (not interlinked but elastic)
Libraries utilized	Shap, Tensorflow, SKLEARN, Pandas
Used dataset	"SDOT Collisions All Years" dataset [40]

8.2. Obtained Results

During the implementation, the following results were obtained.

8.2.1. Accuracy Values

Accuracy values for possibilities of different vehicles colliding were calculated for different machine learning algorithms, i.e., logistic regression, random forest, XGBoost, and extra trees. We obtained accuracy values of 98.86, 99.25, 99.68, and 99.83, with the logistic regression, random forest, XGBoost, and extra trees algorithms, respectively. Here, it is important to mention that extra trees provided the highest value of accuracy as it suits these data and the organization of the dataset. Therefore, we obtained the highest accuracy value in this case. Similar results are reported in Figure 6.

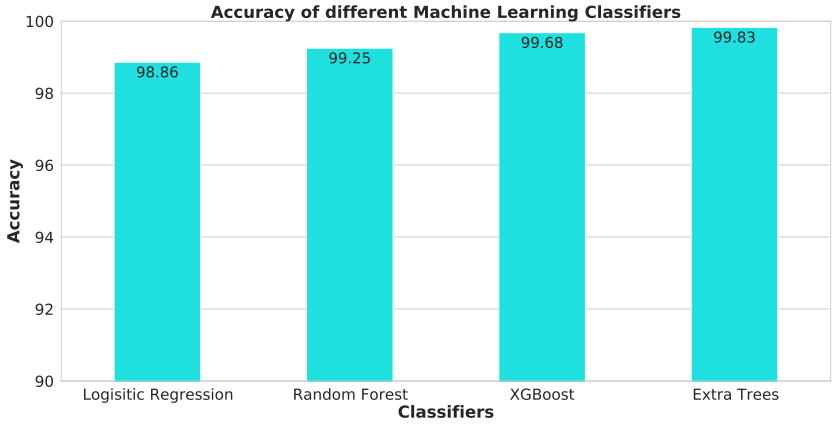

Figure 6. Accuracy values for the possibilities of different vehicles colliding.

8.2.2. F1-Score Values

F1-Score values for possibilities of colliding of different vehicles were calculated for different machine learning algorithms, i.e., logistic regression, random forest, XGBoost, and extra trees. We obtained F1-Score values 0.9850, 0.9914, 0.9968, and 0.9986 with the logistic regression, random forest, XGBoost, and extra trees algorithms, respectively. Here, it is important to mention that extra trees provided the highest value of F1-Score as the extra trees algorithm works well with the organization of the used dataset and makes predictions in a better way. Therefore, we obtained the highest F1-Score value in this case. Similar results are reported in Figure 7.

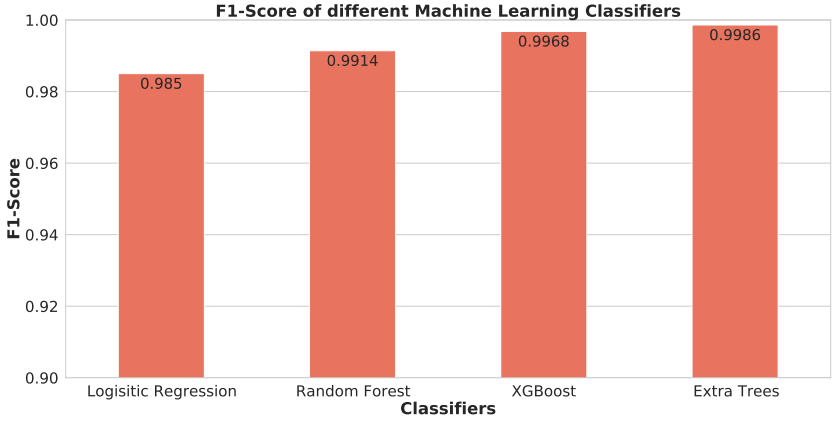

Figure 7. F1-Score values for possibilities of different vehicles colliding.

8.2.3. Severity Distribution

Figure 8 gives the severity distribution for different scenarios. We have different types of collision cases, i.e., injury collision, property damage only collision, fatality collision, and serious injury collision. In the given Figure 8, we can see that we have more than 100,000 cases of "property damage only collision" and more than 40,000 cases of "injury collision". However, other cases of injuries are very few, as compared to these cases.

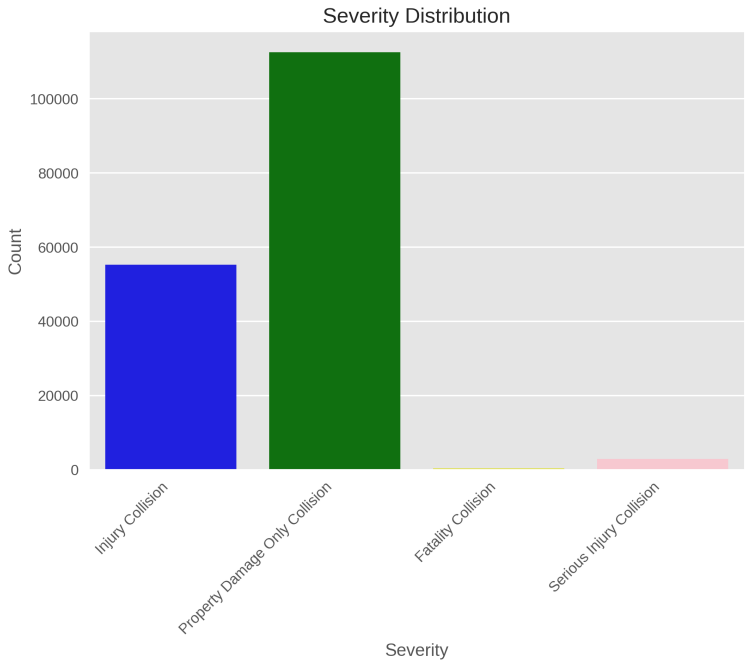

Figure 8. Severity distribution for different scenarios.

8.2.4. Scatter Plots

The different scatter plots for various cases are given in Figure 9 (person count versus pedestrian count), Figure 10 (injuries count versus vehicle count) and Figure 11 (person count versus vehicle count). From these scatter plots, it is clear that when we have vehicles between 0 and 10, then we have a high possibility of injuries. Furthermore, we have the highest person count when we have five vehicles. Hence, from this discussion, it is clear that we have more injuries when the number of vehicles lies between 0 and 10.

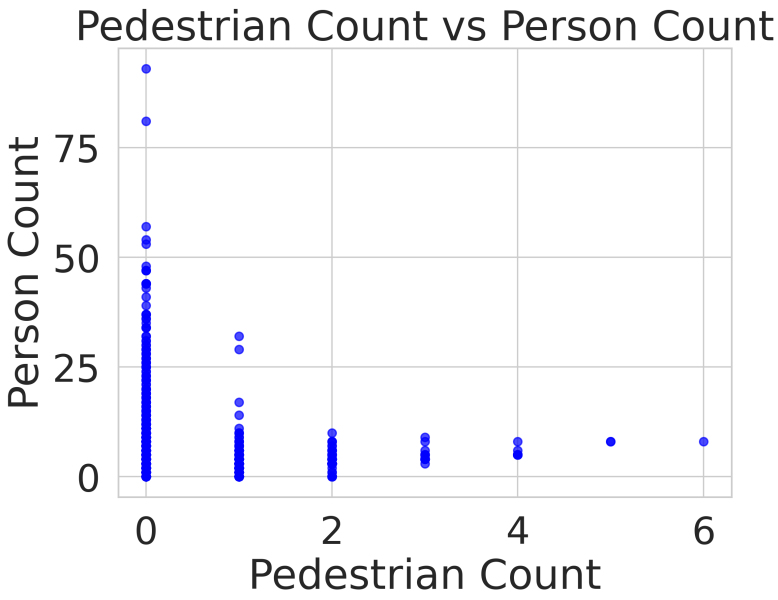

Figure 9. Scatter plot-1.

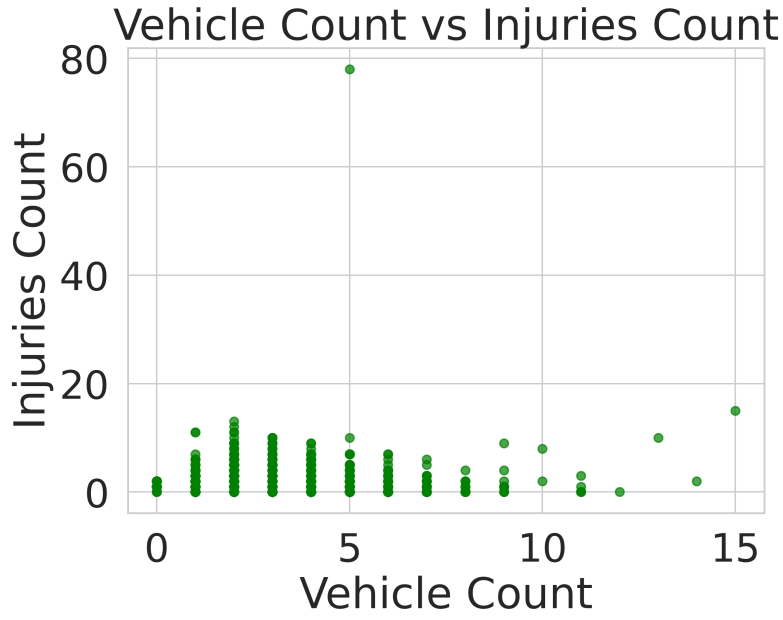

Figure 10. Scatter plot-2.

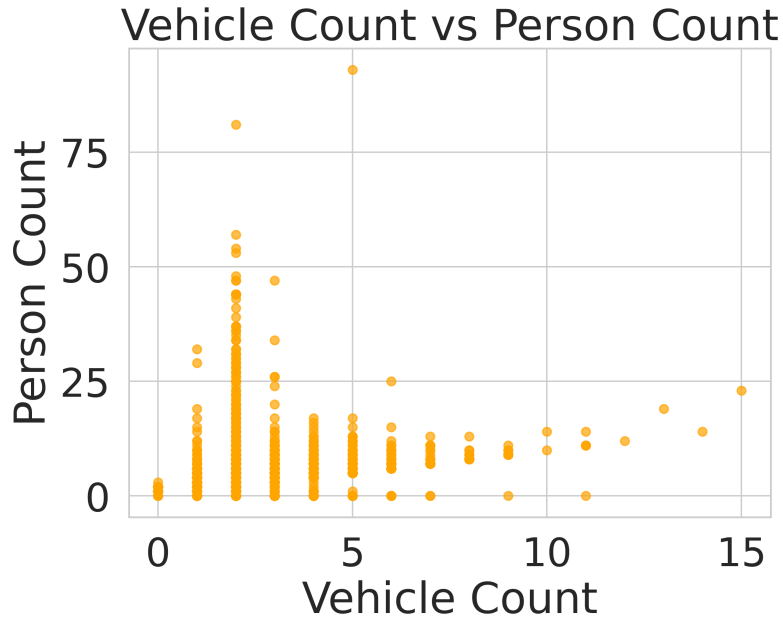

Figure 11. Scatter plot-3.

9. Conclusions

Security and privacy are the main concerns with communication in an IoD network. A secure blockchain-enabled authentication key management framework with big data analytics for drones in networks beyond 5G applications (SBBDA-IoD) was presented that relied on the authenticated key management paradigm. The detailed network model and the associated threat for SBBDA-IoD were provided. The detailed security analysis, i.e., informal security analysis through mathematical assumptions and equations and formal security verification using the Scyther tool, proved the security of SBBDA-IoD. The comparative analysis shows that SBBDA-IoD outperformed the other schemes regarding superior security and efficiency. The real-world implementation of SBBDA-IoD was also performed to evaluate its effect on several important measures for performance.

One of our goals in the future is to enhance the usefulness of SBBDA-IoD, which is currently being offered with additional capabilities. In addition, we wish to incorporate deep learning into SBBDA-IoD so that we can perform superior data analysis and make more accurate predictions.

Author Contributions: Conceptualization, A.K.M. and M.W.; methodology, A.K.M., M.W., J.S. and A.K.D.; software, A.K.M., M.W. and J.S.; validation, A.K.M., M.W., D.P.S., J.S., A.K.D. and A.V.V.; formal analysis, M.W. and A.K.D.; investigation, M.W., D.P.S., A.K.D. and A.V.V.; resources, M.W. and A.K.D.; data curation, A.K.M., M.W., J.S. and A.K.D.; writing—original draft preparation, A.K.M., M.W. and J.S.; writing—review and editing, A.K.M., M.W., D.P.S., J.S., A.K.D. and A.V.V.; supervision, M.W., D.P.S. and A.K.D.; project administration, M.W., A.K.D. and A.V.V.; funding acquisition, A.V.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Gharibi, M.; Boutaba, R.; Waslander, S.L. Internet of Drones. *IEEE Access* **2016**, *4*, 1148–1162.
2. Abdelmaboud, A. The Internet of Drones: Requirements, Taxonomy, Recent Advances, and Challenges of Research Trends. *Sensors* **2021**, *21*, 5718. [CrossRef]
3. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J.P.C. Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment. *IEEE Internet Things J.* **2019**, *6*, 3572–3584.
4. Mozaffari, M.; Taleb Zadeh Kasgari, A.; Saad, W.; Bennis, M.; Debbah, M. Beyond 5G With UAVs: Foundations of a 3D Wireless Cellular Network. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 357–372.
5. Wu, Q.; Xu, J.; Zeng, Y.; Ng, D.W.K.; Al-Dhahir, N.; Schober, R.; Swindlehurst, A.L. A Comprehensive Overview on 5G-and-Beyond Networks With UAVs: From Communications to Sensing and Intelligence. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2912–2945. [CrossRef]
6. Yahuza, M.; Idris, M.Y.I.; Ahmedy, I.B.; Wahab, A.W.A.; Nandy, T.; Noor, N.M.; Bala, A. Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges. *IEEE Access* **2021**, *9*, 57243–57270.
7. Gupta, R.; Bhattacharya, P.; Tanwar, S.; Kumar, N.; Zeadally, S. GaRuDa: A Blockchain-Based Delivery Scheme Using Drones for Healthcare 5.0 Applications. *IEEE Internet Things Mag.* **2021**, *4*, 60–66. [CrossRef]
8. Bera, B.; Wazid, M.; Das, A.K.; Rodrigues, J.J.P.C. Securing Internet of Drones Networks Using AI-Envisioned Smart-Contract-Based Blockchain. *IEEE Internet Things Mag.* **2021**, *4*, 68–73. [CrossRef]
9. Jangirala, S.; Das, A.K.; Vasilakos, A.V. Designing Secure Lightweight Blockchain-Enabled RFID-Based Authentication Protocol for Supply Chains in 5G Mobile Edge Computing Environment. *IEEE Trans. Ind. Inform.* **2020**, *16*, 7081–7093.
10. Dibaei, M.; Zheng, X.; Xia, Y.; Xu, X.; Jolfaei, A.; Bashir, A.K.; Tariq, U.; Yu, D.; Vasilakos, A.V. Investigating the Prospect of Leveraging Blockchain and Machine Learning to Secure Vehicular Networks: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 683–700.
11. Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; Aledhari, M. Enabling Drones in the Internet of Things With Decentralized Blockchain-Based Security. *IEEE Internet Things J.* **2021**, *8*, 6406–6415. [CrossRef]
12. Singh, M.P.; Aujla, G.S.; Bali, R.S. Blockchain for the Internet of Drones: Applications, Challenges, and Future Directions. *IEEE Internet Things Mag.* **2021**, *4*, 47–53. [CrossRef]
13. Ali, Z.; Chaudhry, S.A.; Ramzan, M.S.; Al-Turjman, F. Securing Smart City Surveillance: A Lightweight Authentication Mechanism for Unmanned Vehicles. *IEEE Access* **2020**, *8*, 43711–43724. [CrossRef]
14. Rodrigues, M.; Amaro, J.; Osório, F.S.; Branco Kalinka, R.L.J.C. Authentication Methods for UAV Communication. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 29 June–3 July 2019; pp. 1210–1215.
15. Kirsal Ever, Y. A Secure Authentication Scheme Framework for Mobile-Sinks used in the Internet of Drones Applications. *Comput. Commun.* **2020**, *155*, 143–149. [CrossRef]
16. Bera, B.; Das, A.K.; Sutrala, A.K. Private Blockchain-Based Access Control Mechanism for Unauthorized UAV Detection and Mitigation in Internet of Drones Environment. *Comput. Commun.* **2021**, *166*, 91–109. [CrossRef]
17. Feng, C.; Liu, B.; Guo, Z.; Yu, K.; Qin, Z.; Choo, K.K.R. Blockchain-Based Cross-Domain Authentication for Intelligent 5G-Enabled Internet of Drones. *IEEE Internet Things J.* **2022**, *9*, 6224–6238. [CrossRef]
18. Lwin, K.K.; Sekimoto, Y.; Takeuchi, W.; Zettsu, K. City Geospatial Dashboard: IoT and Big Data Analytics for Geospatial Solutions Provider in Disaster Management. In Proceedings of the IEEE International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), Paris, France, 18–20 December 2019; pp. 1–4. [CrossRef]

19. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. Applications, Deployments, and Integration of Internet of Drones (IoD): A Review. *IEEE Sens. J.* **2021**, *21*, 25532–25546. [CrossRef]
20. Pu, C.; Wall, A.; Choo, K.K.R.; Ahmed, I.; Lim, S. A Lightweight and Privacy-Preserving Mutual Authentication and Key Agreement Protocol for Internet of Drones Environment. *IEEE Internet Things J.* **2022**, *9*, 9918–9933. [CrossRef]
21. Krichen, M.; Lahami, M.; Al–Haija, Q.A. Formal Methods for the Verification of Smart Contracts: A Review. In Proceedings of the 2022 15th International Conference on Security of Information and Networks (SIN), Sousse, Tunisia, 11–13 November 2022; pp. 1–8. [CrossRef]
22. Abdellatif, T.; Brousmiche, K.L. Formal Verification of Smart Contracts Based on Users and Blockchain Behaviors Models. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 2018; pp. 1–5. [CrossRef]
23. Dolev, D.; Yao, A. On the Security of Public Key Protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
24. Canetti, R.; Krawczyk, H. Universally Composable Notions of Key Exchange and Secure Channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques—Advances in Cryptology (EUROCRYPT 2002), Amsterdam, The Netherlands, 28 April–2 May 2002; pp. 337–351.
25. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Trans. Comput.* **2002**, *51*, 541–552. [CrossRef]
26. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment. *IEEE Trans. Depend. Secur. Comput.* **2020**, *17*, 391–406. [CrossRef]
27. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst.* **2002**, *20*, 398–461. [CrossRef]
28. Wazid, M.; Bera, B.; Das, A.K.; Mohanty, S.P.; Jo, M. Fortifying Smart Transportation Security Through Public Blockchain. *IEEE Internet Things J.* **2022**, *9*, 16532–16545. [CrossRef]
29. Tableau. Big Data Analytics: What It Is, How It Works, Benefits, and Challenges. 2023. Available online: https://www.tableau.com/learn/articles/big-data-analytics (accessed on 5 June 2023).
30. Tsai, C.W.; Lai, C.F.; Chao, H.C.; Vasilakos, A.V. Big Data Analytics: A Survey. *J. Big Data* **2015**, *2*, 21. [CrossRef]
31. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V. Design of Secure Key Management and User Authentication Scheme for Fog Computing Services. *Future Gener. Comput. Syst.* **2019**, *91*, 475–492. [CrossRef]
32. Khadem, B.; Suteh, A.M.; Ahmad, M.; Alkhayyat, A.; Farash, M.S.; Khalifa, H.S. An Improved WBSN Key-Agreement Protocol Based on Static Parameters and Hash Functions. *IEEE Access* **2021**, *9*, 78463–78473. [CrossRef]
33. Cremers, C.J.F. Scyther: Semantics and Verification of Security Protocols. 2006. Available online: https://pure.tue.nl/ws/files/2425555/200612074.pdf (accessed on 16 November 2022).
34. Tanveer, M.; Zahid, A.H.; Ahmad, M.; Baz, A.; Alhakami, H. LAKE-IoD: Lightweight Authenticated Key Exchange Protocol for the Internet of Drone Environment. *IEEE Access* **2020**, *8*, 155645–155659. [CrossRef]
35. Adeli, M.; Bagheri, N.; Meimani, H.R. On the Designing a Secure Biometric-Based Remote Patient Authentication Scheme for Mobile Healthcare Environments. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 3075–3089. [CrossRef]
36. NIST. Advanced Encryption Standard, 2001, FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce. 2001. Available online: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (accessed on 6 April 2023).
37. Wu, L.; Wang, J.; Choo, K.K.R.; He, D. Secure Key Agreement and Key Protection for Mobile Device User Authentication. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 319–330. [CrossRef]
38. Vangala, A.; Roy, S.; Das, A.K. Blockchain-Based Lightweight Authentication Protocol for IoT-Enabled Smart Agriculture. In Proceedings of the International Conference on Cyber-Physical Social Intelligence (ICCSI), Nanjing, China, 18–21 November 2022; pp. 110–115.
39. Barker, E. Recommendation for Key Management, 2014. Special Publication 800-57 Part 1 Rev. 4, NIST, 01/2016. Available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf (accessed on 6 April 2023).
40. Seattle. SDOT Collisions All Years. 2023. Available online: https://data.seattle.gov/dataset/SDOT-Collisions-All-Years/79xi-y524 (accessed on 9 June 2023).