A Redactable Blockchain-Assisted Application-Aware Authentication System for Internet of Drones

Cong Pu, Abdullah Bilal, Nohpill Park, Jongho Seol, and Kim-Kwang Raymond Choo

***Abstract*—The Internet of Drones (IoD) has latterly started to gear up its applications in diverse sectors of the society as a result of high adaptability and adjustability to new circumstances. Security, privacy, and storage issues still remain as major barriers for next-generation IoD systems to meet their general applicability requirements, even though many one-key-for-all static authentication and append-only blockchain assisted systems have been proposed by the IoD community. First, bearing channel bandwidth and drones' resource constraints in mind, authentication protocols with less computation and communication overhead are preferable. Second, the IoD drones might collect different types of data simultaneously, a unique secret session key for each type of data is needed to prevent data leakage from unauthorized parties. Third, the permanent storage of each drone's cryptographic and task information on the append-only blockchain raises significantly alert after a long period of operation and/or an exponential growth of drones. Motivated by the research challenges presented above, we propose a redactable blockchain-assisted application-aware authentication system, also referred to as $ReBAS$, for next-generation IoD applications, where the drones shuttle back and forth between different flying zones to collect diverse types of data. The Chebyshev polynomial, redactable consortium blockchain, and chameleon hash function are adopted to significantly minimize the computational, communication, and storage overheads of cryptography-related operations. According to the security verification, and formal and informal security analysis, the $ReBAS$ not only guarantees secure and dynamic authenticated key establishment, but also is in compliance with the security requirements of Canetti-Krawczyk adversarial framework. We also develop a rigorous simulation framework and conduct an extensive comparative study. The experimental results demonstrate that the $ReBAS$ can minimize the overheads in computation, communication, and storage while enhancing scalability.**

***Index Terms*—Redactable Blockchain, Application-Aware, Security and Privacy, Authentication, Internet of Drones**

I. Introduction

With the support of advanced technology from aerodynamics, carbon fibers, silicon chips, to flight control software, drones have grown into one of the most significant leading-edge technologies in the third decade of the 21st century. As the number of commercial and civilian drones is exponentially increasing, the Internet of Drones (IoD) [1] emerged as an architecture to coordinate drones to fly within airspace in an organized, controlled, and fair manner. Drawing on the successful experience of Internet of Things (IoT) commercialization, the IoD architecture will open up a Pandora's box of limitless terrestrial-aerial applications. For example, the global flying car market is expected to reach $1,600 billion by 2040 [2], and the world's first airport for flying cars just opened in the United Kingdom in 2022 [3]. As one of the implications of flying cars, air taxis have been an exciting but unattainable dream for years. Now they have a rollout schedule and the estimated arrival date is in 2028 [4]. Backed by the IoD architecture, the real-time information (e.g., pickup time, arrival time, route lines, etc.) can be seamlessly synchronized among the apps of air taxis, which in turn improve travel experience for passengers. Thus, the IoD architecture is envisioned to significantly promote the development and realization of air taxi commercialization (e.g., Archer Aviation's electric air taxis) in smart cities.

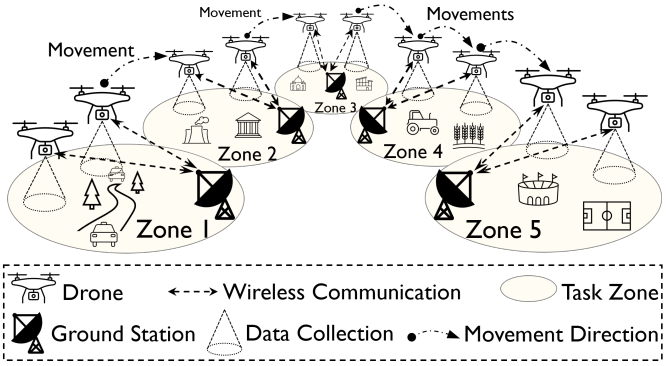


Fig. 1. IoD architecture and example applications. Zone 1: traffic surveillance; Zone 2: sport & entertainment; Zone 3: industrial plants monitoring; and Zone 4: precision agriculture. For simplicity, each application is shown in one zone. However, in the proposed IoD systems, multiple applications can be performed concurrently in one zone.

Fig. 1 presents a typical IoD architecture, where the terrestrial region along with its airspace above are divided into different task zones. The IoD drones fly back and forth between different task zones to collect various environmental data and/or IoT sensory data, and then submit the collected data to stationary networking systems for deep analysis. The IoD paradigm enables a myriad of drones seamlessly inter-

Manuscript received xxxx xx, xxxx; revised xxxx xx, xxxx. (Corresponding author: Cong Pu.) This research work was supported by the National Science Foundation (NSF) under SaTC Award No. 2333777.

C. Pu, A. Bilal, and N. Park are with Oklahoma State University, USA (e-mail: cong.pu@outlook.com, abdullah.bilal@okstate.edu, n.park@okstate.edu).

J. Seol is with Middle Georgia State University, USA (e-mail: jongho.seol@mga.edu).

K.K.R Choo is with The University of Texas at San Antonio, USA (e-mail: raymond.choo@fulbrightmail.org).

act with each other through ground stations without human intervention, which facilitates the creation of numerous IoD applications (e.g., Joby's electric flying car commercialization, Amazon Prime Air drone deliveries in the west valley phoenix metro area). In the era of smart city, drones are regarded as one of major elements that provide citizens and local government authorities with smart mobility. For example, the Joby's electric air taxis [5] can use their mounted cameras to not only find criminal suspects and missing children, but also observe road conditions, while delivering passengers to destinations. Many organizations can benefit from such a unified conceptual architecture in numerous ways in the near future - such that improving task efficiency, usability or safety, lowering the environmental impact of business, and automating operational processes. Whether you are already an IoD guru or barely a novice, the IoD technology will gradually increase its impact on our personal and professional lives.

After briefly describing the IoD advantages, we shall delve into more nebulous long-term issues to consider. First of all, the initial design of IoD architecture is not concerned with security and privacy requirements. As a result, an attacker is eager to use these design weaknesses to obtain unauthorized access to IoD systems and destroy, manipulate, or steal data. As the first line of defense, quite a few authentication protocols have emerged as a promising solution to protect IoD systems from attacks. Nevertheless, the state-of-the-art methods either incur high CPU time and execution time or demand a large number of communication messages for mutual authentication and key agreement negotiation. Second, in order to improve the utilization rate of drones, the IoD application platforms might assign different types of tasks (data) to drones to complete (collect). Using the same cryptographic key (i.e., secret session key) to encrypt different types of data collected by the drone and submit the encrypted data as a single packet to the ground station will cause potential data leakage, where one application might have access to the data which is meant for another application. Third, the IoD drones usually shuttle between different task zones, thus, it is necessary to establish the trusted communication among all task zones. To support cross-zone authentication, some studies integrate authentication with blockchain technology. However, the permanent storage of each drone's cryptographic and task information on the unprunable blockchain raises storage concerns after a long period of operation and/or an exponential growth of drones.

In this paper, we intend to demystify the research challenges presented above and clarify issues from security, privacy, and storage perspectives. The conception of security, privacy, and storage in the context of IoD is investigated, and specific IoD's security and performance requirements are documented. This paper proposes a cryptographic solution which features application-specific mutual authentication between the drone and the ground station in the IoD environment, where chaos-based cryptography, consortium blockchain, and trapdoor function are introduced to strengthen the security, privacy, and storage performance of IoD systems. In our proposed research, Chebyshev polynomial-based cryptography leverages the semi-group property of polynomials to provide good approximations with relatively few terms and create

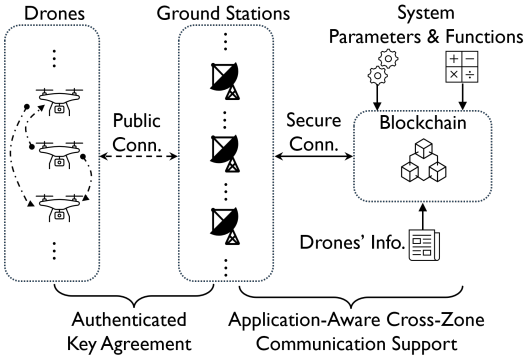


Fig. 2. The schematic diagram of the proposed $ReBAS$.

an efficient and secure cryptographic algorithm. In addition, redactable consortium blockchain and chameleon hash function are chosen to reduce the storage overhead of blockchain system which is used to store drone's cryptographic and task information In a few words, our major contribution is briefly summarized in twofold:

- We propose a redactable blockchain-assisted application-aware authentication system, also referred to as $ReBAS$, for next-generation IoD applications, where the Chebyshev polynomial, redactable consortium blockchain, and chameleon hash function are adopted to significantly reduce the computational, communication, and storage overheads of cryptography-related operations.
- We use the AVISPA tool [6] and Mao's BAN logic [7] to formally verify and analyze the $ReBAS$ in terms of security and privacy, respectively. In addition, we informally inspect the operations of $ReBAS$ from the perspective of the adversary. To promote collaboration and accessibility, the $ReBAS$ source codes and AVISPA verification programs are publicly available at the https://github.com/congpu/ReBAS.

Our approach $ReBAS$ is composed of two major components: (i) Authentication and Key Negotiation; and (ii) Redactable Blockchain System. The proposed authentication and key negotiation approach is original because it departs from the status quo of failing to distinguish between different types of data collected by drones during the authentication process. Moreover, the proposed redactable consortium blockchain system is novel because it adopts Hyperledger Fabric consortium blockchain and chameleon hash function to realize cross-domain support and enable service change flexibility while alleviating the storage overhead of IoD systems. The schematic diagram of the proposed $ReBAS$ is shown in Fig. 2. Regarding the expected outcomes, we anticipate that the proposed research solutions can be easily integrated with the existing IoD paradigm and enable IoD systems to achieve their application requirements. We build a rigorous simulation environment, implement the $ReBAS$ and benchmark schemes, and conduct extensive comparative experimental tests. The experimental results demonstrate that the $ReBAS$ achieves superior performance with better scalability and lower computational, communication, and storage overheads.

The research conducted in this article holds significant importance. To unlock the potential of IoD paradigm for

emerging commercial applications, IoD-specific research challenges such as data heterogeneity, communication security and privacy, cross-domain communication support, storage overhead, and service change flexibility need to be addressed properly. The data types become richer as the drones are being used in diverse IoD applications concurrently. But the existing authentication protocols do not distinguish between different types of data collected by the drones during the authentication process, which leads to potential data leakage where one application might have access to the data which is meant for another application. Usually, drones are scheduled to move from one place to another while providing a set of registered services. In order to enable secure communication between drones and ground stations which are located in different areas or comes from different domains, blockchain technology is regarded as one of feasible techniques. However, permanently storing each drone's identity, cryptographic, and registered service information on the unprunable blockchain significantly increases the storage overhead of IoD systems after a long period of operation and/or an exponential growth of drones. Third, drones might frequently change the registered services. Hence, the IoD systems shall possess service change flexibility without increasing storage overhead so that the change of registered services has minimal cost and performance implications. In summary, the proposed approach is anticipated to change the current situation of using unsuitable techniques as well as bridge the missing research gaps for secure communication protocol and redactable blockchain system in the IoD community. Moreover, this research is capable of paving the way for the development of secure aerial computing framework and providing design consideration for other open and interoperable platforms. In practical terms, the research outcomes would integrate with the IoD paradigm to provide a secure environment for emerging IoD applications.

The remainder of the paper is organized as follows. The state-of-the-art approaches are presented, analyzed, and compared in Section II. In Section III, we introduce the adopted techniques. We present the system and adversarial models as well as security and performance requirements in Section IV. After that, we demonstrate the design of the propose system in Section V. Section VI and VII are devoted to system analysis and evaluation, respectively. Finally, we conclude the paper in Section IX.

II. Literature Review

In the IoD networks, plenty of work has been done to secure IoD communication and data exchange. To protect IoD data from unauthorized access, Tanveer *et al.* [8] proposed an anonymous authentication solution in which the user and the drone execute chaotic, hash, and symmetric encryption functions to negotiate a secret session key after confirming each other's identity. Some other works [9], [10] have used resource-friendly operations such as hash and bitwise XOR operations to achieve mutual authentication between the user and the drone. However, in the solutions presented above, the secret session key is formalized through exchanging three (3) messages between the user and the drone, which inevitably incurs non-negligible computation and communication costs. The IoD drones are constrained in resources, and thus the solutions with less computation and communication overhead are preferable. Additionally, the type of data is not being considered in the authentication operations.

To create service-specific secret session keys in the IoD systems, El-Zawawy *et al.* [11] integrated data types into the process of authentication. By treating each type of data separately, a secret session key can be used to encrypt the specific type of data collected by a drone, and therefore it is likely to protect IoD systems from data leakage. However, El-Zawawy's approach cannot be applied to next-generation IoD applications due to the lack of support for cross-zone communication/authentication.

In another line of work, researchers use blockchain technology [12]–[14] to establish direct/indirect trust between entities from different zones in the IoD environment, however, none of them considered the types of data in the design process. García *et al.* [12] proposed a blockchain-based solution in which the drone embeds a unique message authentication code (MAC) into each message to guarantee that the message could not be manipulated by the adversary. Here, the timed efficient stream loss-tolerant authentication (Tesla) protocol is adopted to generate a distinct key for each MAC. Since the original Tesla protocol does not support cross-zone communication, blockchain technology is used to realize the coordinated authentication operations between ground stations from different task zones. Yu *et al.* [13] focused on a variant of IoD networks, flying ad hoc networks, and used blockchain technology to realize access control and data integrity. In [14], a public blockchain is utilized as an immutable database to store critical and confidential data collected by drones. Both works implicitly assume that all data fall under one category, and the data type is not considered as one of ingredients in the creation of secret session keys.

Immutability is one of the main driving forces for the wide adoption of blockchain technology, however, there are ways the immutability of the blockchain ledger can raise concerns. In [15], the authors use the blockchain network to store drone's registration information, collected data, and digital signatures. In [16], a private blockchain network is deployed to store the data collected by drones. In [17], the IoD ground stations verify digital signatures created by drones, aggregate all received digital signatures, and then upload them to the blockchain network. After the IoD applications have been running for a long time or experienced an exponential increase in the number of drones, however, the network administrators might be concerned with the storage cost or even performance issues due to the perpetual storage of blockchain transactions. In addition, erroneous data could be accidentally added into the blockchain ledger because of either the glitches of consensus programs or the software that interfaces with the blockchain system.

Finally, we investigate the above state-of-the-art works based on various security features and performance requirements, and highlight their differences in Table. I. Our solution $ReBAS$ is shown in the last column of Table. I. The IoD systems are regarded as open and integrated architecture of

TABLE I
COMPARISON OF EXISTING WORKS

Feature	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	$ReBAS$
MU	●	●	●	●	●	●	●	●	●	●	●
MI	●	●	●	●	●	●	●	●	●	○	●
DA	●	●	●	●	○	●	○	●	●	○	●
AS	○	○	○	●	○	○	○	○	○	○	●
CT	○	○	○	○	●	●	●	●	○	●	●
LO	○	●	●	●	●	●	○	●	○	○	●
IS	○	○	○	○	○	○	○	○	○	○	●

●: Provides ○: Does Not Provide
Features: **MU:** Mutual Authentication; **MI:** Message Integrity; **DA:** Drone Anonymity; **AS:** Application-Aware Session Key Establishment; **CT:** Cross-Zone Trust; **LO:** Lightweight Operations; **IS:** Impermanent Storage.

interoperable platforms, and their security approaches are multi-faceted and need to meet various types of security and performance needs. As different IoD security approaches offer different levels of security strength and performance efficiency, we decide to choose the most common, important, and representative security features (e.g., mutual authentication, message integrity, drone anonymity, application-aware session key establishment, and cross-zone trust) and performance requirements (e.g., lightweight operations and impermanent storage) to conduct the comparison between the proposed approach and the existing works. Through integrating Chebyshev polynomial, redactable consortium blockchain, and chameleon hash function, the $ReBAS$ is able to guarantee an authenticated and data type-aware key establishment anonymously between the drone and the ground station while providing decentralized and redactable management of drone relevant information via a redactable consortium blockchain. In addition, the cryptographic component of the $ReBAS$ is realized with the integration of advanced but lightweight technique Chebyshev polynomial to address the resource concerns of drones. Here, the 'lightweight' refer to cryptographic operations that are designed to be efficient and require minimal computational resources. Besides conducting comparison in Table. I, we also select two representative approaches, USAF-IoD [18] and PAF-IoD [19], and compare them with our solution $ReBAS$ in terms of several quantified metrics such as CPU time, energy consumption, authentication latency, authentication scalability index, communication cost in Section VII.

III. PRELIMINARY BACKGROUND

A. Chebyshev Polynomial for Cryptography

According to the Diffie-Hellman method [20], $(g^n)^m = (g^{mn}) = (g^m)^n \bmod p$, two unrecognized parties (e.g., Alice and Bob) can safely reach an agreement on a secret key over a public channel. Here, g is a random positive integer, p is a prime number, and m and n are the secret integer exponent for Alice and Bod, respectively. Alice first calculates $a = (g^m)$ *mod* p and sends it to Bod. Then, Bob calculates $b = (g^n)$ *mod* p and shares it with Alice. Finally, Alice calculates the secret key as $c = (b^m)$ *mod* p, while Bob calculates the secret key as $d = (a^m)$ *mod* p. Here, $c == d$ as $(g^n)^m = (g^{mn}) = (g^m)^n$ *mod* p according to the Diffie-Hellman method. The rationale is shown below:

$$\begin{aligned} c &= (b^m)\ mod\ p \\ &= ((g^n)\ mod\ p)^m\ mod\ p \\ &= (g^{nm})\ mod\ p \\ &= (g^{mn})\ mod\ p \\ &= ((g^m)\ mod\ p)^n\ mod\ p \\ &= (a^n)\ mod\ p \\ &= d \end{aligned}$$

Thus, the Diffie-Hellman method can help two unknown parties agree on their common secret key. Here, the calculation process of c == d is used to show that two unknown parties can securely establish a secret key using previously shared information over a public and unsecure channel.

Over time the Diffie-Hellman method has been generalized from modulo multiplication group to Chebyshev polynomial of the first kind [21]. Suppose that $(x^m)^n = (x^{mm}) = (x^n)^m$ is a polynomial identity, where x is an indeterminate. The following identity holds for Chebyshev polynomial of the first kind,

$$T_m(T_n(x)) = T_{mn}(x) = T_n(T_m(x)),$$

where $T_n(x) = cos(n\, arccos(x))$, all real number $x \in$ [-1,1]. The proof is as follows

$$\begin{aligned} T_m(T_n(x)) &= cos(m\, arccos(cos(n\, arccos(x)))) \\ &= cos(mn\, arccos(x)) \\ &= cos(nm\, arccos(x)) \\ &= cos(n\, arccos(cos(m\, arccos(x)))) \\ &= T_n(T_m(x)). \end{aligned}$$

B. Chameleon Hash Function

The chameleon hash function (C_H) was first proposed by Krawczyk *et al.* [22]. A C_H consists of four sub-algorithms: CKeyGen, CHash, CHashVer, and CHashCol; the definition of each sub-algorithm is given below:

- $(trk,\ puk) \leftarrow$ CKeyGen(1^λ): The CKeyGen takes a security parameter λ as input, and outputs a chameleon trapdoor key trk and a chameleon public key puk.
- $h \leftarrow$ CHash($puk,\ m,\ r$): The CHash is fed with the chameleon public key puk, a message m, and a random number r, and produces a hash value h.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3562857

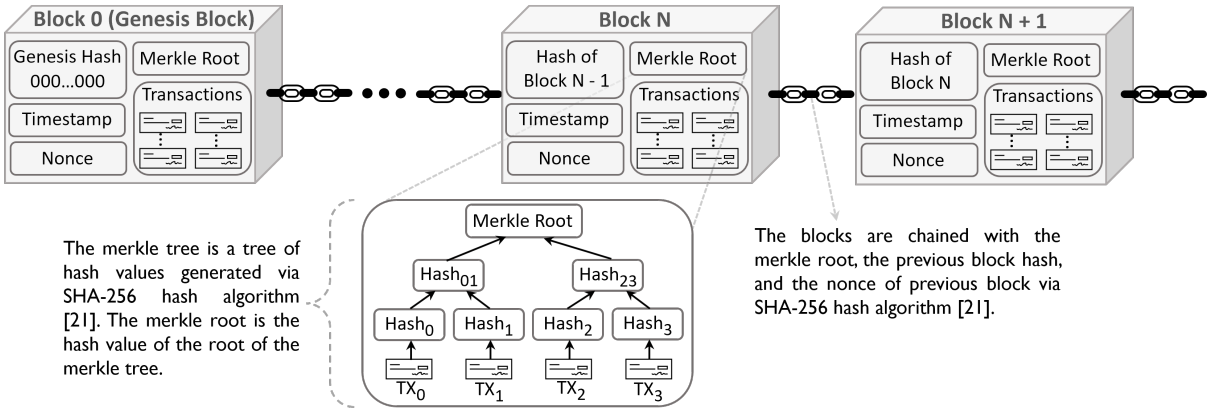


Fig. 3. The structure of traditional blockchain.

- $d \leftarrow$ CHashVer(puk, m, r, h): The CHashVer accepts the chameleon public key puk, a message m, a random number r, and a hash value h as input, and yields a boolean value d. If h is the hash value corresponding to the message m, d is true, otherwise d is false.
- $r^{'} \leftarrow$ CHashCol(trk, m, r, h, $m^{'}$): The CHashCol is provided with the chameleon trapdoor key trk, the old message m, the old random number r, the hash value h, and the new message $m^{'}$, and returns a new random number $r^{'}$ corresponding to the new message $m^{'}$ for the same hash value h.

The unique property of C_H is that a hash collision of the new message $m^{'}$ along with the corresponding random number $r^{'}$ can be easily calculated by the CHashCol when the chameleon trapdoor key trk is available. But for those who do not know trk, the CHashCol is collision resistant. In other words, the party who owns trk can decide whether others shall be able to equivocate the hash value by providing or withholding trk.

C. Overview of Traditional Blockchain

Blockchain technology is widely adopted as a decentralized storage system, where all participants cooperatively manage the data stored in the blockchain ledger. There are three major building components in the blockchain system: blockchain structure, cryptographic algorithm, and consensus algorithm. The basic structure of blockchain is shown in Fig. 3, where the blocks are chained in chronological order and each block contains five elements:

- Hash of Previous Block: The content of this element is the hash value of the previous block, which is used to chain blocks.
- Timestamp: The timestamp helps to establish the order of transactions and blocks in the blockchain ledger.
- Nonce: For blockchain systems utilizing mining mechanism, the value of nonce is generated by the publishing participant to solve the cryptographic puzzle. Other blockchain systems might or might not include this element or use it for other purposes.
- Transactions: The changes of blockchain's state are recorded through transactions.

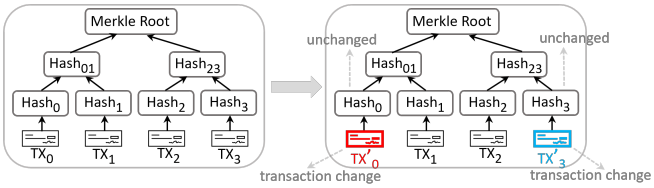


Fig. 4. Overview of redaction operations.

- Merkle Root: The value of merkle root is the hash value of the root of the merkle tree, where each transaction is matched with a unique transaction hash value.

In the following, we use the notations defined by Ye *et al.* [23] to explain the operations of constructing a blockchain. When the blockchain system is initialized, the first block, which is called the genesis block, is created to store the initial state of the system. In the future, other blocks can be added to the blockchain after the genesis block. The i^{th} $(0 < i)$ block is defined as a tuple $B_i := \{\alpha_i, \beta_i, \delta_i\}$. Here, α_i is the hash value of previous block, β_i is the value of merkle root, and δ_i is the nonce. Accordingly, the $i + 1^{th}$ block is defined as $B_{i+1} := \{\alpha_{i+1}, \beta_{i+1}, \delta_{i+1}\}$, where $\alpha_{i+1} = H\big(\delta_i, G(\alpha_i, \beta_i)\big)$. Here, H and G are called outside and inside hash function, respectively.

The consensus algorithms (e.g., proof-of-work, proof-of-stake, etc.) accept the responsibility for helping all blockchain participants maintain the consistency of transactions in the ledger. Speaking of the cryptographic algorithms, they are mainly dedicated to ensure the security of transactions and participants. Thanks to the underlying structure and techniques, a blockchain system is featured with anonymity, immutability, decentralization, and transparency.

D. Redactable Blockchain

One of the main features of blockchain is immutability, which is achieved through the collision-resistant hash function. However, in our proposed research, we decide to look at the immutability feature from a storage overhead perspective. In order to achieve cross-domain authentication, a blockchain system proves to be useful. However, the permanent storage

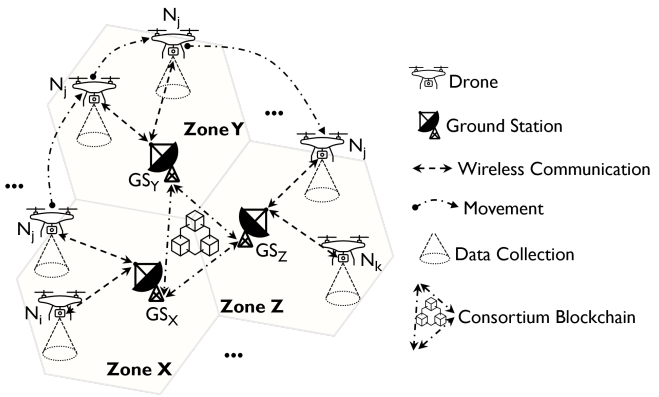


Fig. 5. System model.

of each drone's cryptographic and task information on the unprunable blockchain system raises storage concerns after a long period of operation and/or an exponential growth of drones. Thus, a redactable blockchain becomes an ideal candidate in our proposed research. In order to establish a redactable blockchain, Derler *et al.* [24] proposed the first solution that replaces the traditional blockchain's transaction hash function with the chameleon hash function. The fundamental idea is to discover a hash collision of the transaction so that changing the transaction does not impact the merkle root value.

As shown in Fig. 4, a hash representation of transactions is accomplished through a merkle tree. The merkle tree is a tree-like structure where the transaction is hashed and combined until there is a singular merkle root hash. Being knowledgeable about the trapdoor key, it is likely to discover a hash collision of a transaction. As a result, the original merkle tree is replaced with a new transaction, however, the block remains unchanged. To be specific, the transaction hash function is defined as h_i = CHash(puk_i, TX_i, r_i), where TX_i is the i^{th} transaction in the block, and puk_i and r_i are the parameters of C_H which are defined before. Suppose that the transaction TX_i needs to be edited without affecting the upper merkle tree structure. This can be achieved by computing a new random number $r^{'}$ corresponding to the new transaction $TX_i^{'}$ using the CHashCol, $r^{'}$ = CHashCol(trk, TX_i, r, h, $TX_i^{'}$), and adding the new random number $r^{'}$ and new transaction $TX_i^{'}$ into the block to replace old ones. After that, the new chain is broadcasted on the network. In compliance with previously-agreed redacting rules, every participant shall adopt the new chain, even if it stores a longer one.

IV. System and Adversary Models & Security and Performance Requirements

A. System Model

Fig. 5 presents an overview of the proposed system, where there are drones and ground stations distributing in multiple task zones. Each zone hosts at least one ground station and a set of drones, both of which are able to communicate wirelessly. In order to improve the utilization rate, each drone is assigned with multiple tasks (different types of data) which might be located in different task zones to complete (collect). For example, the drone N_j is an air taxi which is scheduled to deliver a passenger from Zone X to Zone Z. During the trip, the drone N_j not only posts real-time traveling information, but also collects IoT sensory data and observes traffic conditions in Zone Y. In addition, on the way to the destination, the drone N_j frequently submits each type of collected data which is encrypted using a unique secret session key to a nearby ground station. In our system, the ground stations are interconnected through wired Internet connection and collectively establish a redactable consortium blockchain network using Hyperledger Fabric [25] to store drones' identity and task information.

B. Adversary Model

In this paper, we adopt the Canetti-Krawczyk adversary framework [26] to mimic the behaviors of the adversary. The rationale behind this adoption is that the Canetti-Krawczyk adversary framework not only embraces the capabilities of attackers from the Bellare–Rogaway adversary model [27], but also endows the attackers with the ability to obtain the secret information stored in the drone's memory. To be specific, the attacker is able to dominate the IoD communication channels within a probabilistic polynomial time period through monitoring, manipulating, replaying the transmitted messages. In addition, it is assumed that the attacker could launch explicit attacks against the IoD drones to illegally gain access to the current secret session keys which are temporarily stored in the drones' memories. As a result, the on-going communication session between the drone and the ground station might be compromised.

C. Security & Performance Requirements

With the adoption of the Canetti-Krawczyk adversary framework, we specify the following security and performance requirements that the $ReBAS$ is supposed to meet. First, the ground station can validate whether the identity of drone is registered for specific tasks, while the drone shall authenticate the validity of the ground station's identity. Second, after the drone and the ground station finish the mutual authentication process, they can set up a data type specific secret session key. Third, the drone needs to use a pseudonym instead of its real identification to communicate with the ground station. Fourth, the $ReBAS$ should guarantee perfect forward secrecy so that the illegally-obtained session key does not affect either the communication sessions of other drones or the future communication sessions of the targeted drone. Finally, the $ReBAS$ should be invulnerable to well-known IoD security attacks such as targeted phishing attack, message fabrication attack, physical probing attack, message replay attack, and man-in-the-middle attack.

V. The Proposed System $ReBAS$

The $ReBAS$ consists of two sub-systems: (i) application-aware authentication system, which sets up data type-specific secret session keys between the legit drones and the ground stations; and (ii) chameleon hash based redactable blockchain system, which stores drones' identifiable information and assigned tasks in a modifiable manner. All notations and their meanings are summarized in Table. II.

TABLE II
NOTATIONS

Notation	Meaning
SA	System Administrator
GS_k	Ground station k
$T_{(n)}(x)$	Chebyshev polynomial
H	Secure hash function
m	Positive integer
PR_{GS_k}	Private key for GS_k
PU_{GS_k}	Public key for GS_k
C_H	Chameleon hash function
N_i	Drone i
ID_i	Real identification of drone i
cha_i	Drone i's PUF challenge
res_i	Drone i's PUF response
F_{puf}	PUF
PU_i	Public key of ID_i
PID_i	Pseudo identification
r, s, u	Random number
trk_i	Chameleon trapdoor key of ID_i
puk_i	Chameleon public key of ID_i
T_i	A set of tasks that ID_i needs to complete
pol_i^{PU}	Public Chebyshev polynomial calculated by ID_i
$H_{i.j}$	Hash value
req_i^{auth}	Authentication request message from ID_i
$pol_{GS_k}^{PU}$	Public Chebyshev polynomial calculated by GS_k
rep_j^{auth}	Authentication response message from GS_j
$SK_{j.i}$	Session key between participant j and i
$\|\|$	Concatenation
$\oplus$	Exclusive OR

A. Application-Aware Authentication System

The application-aware authentication system is composed of system initialization, drone enrollment, and authentication and key negotiation phases. During the system initialization phase, all system functions and parameters are selected and announced. In the next phase, the ground stations register all drones in their task zones for system-wide communications by adding drones' identifiable information and assigned tasks in the redactable blockchain ledger. In the last phase, the ground stations use the information stored in the ledger to authenticate drones and negotiate secret session keys with them.

System Initialization: In this phase, the system administrator *SA* chooses system parameters and functions, and records them in the genesis block.

1) *SA* determines a Chebyshev polynomial $T_{(n)}(x)$, where x is a real number within [1,-1] and n ($n \geq 1$) is the degree of Chebyshev polynomial.
2) *SA* selects a secure hash function H:$\{0,1\}^* \rightarrow \{0,1\}^m$, which converts a number of any length to a *m*-bit number.
3) *SA* chooses the private key PR_{GS_k} for each ground station GS_k ($k = 1, 2, \ldots$) and calculates the corresponding public key $PU_{GS_k} = T_{(PR_{GS_k})}(x)$. The private key PR_{GS_k} is sent to the ground station GS_k via a secure channel [28], while the public key PU_{GS_k} is added to the genesis block in the follow steps.
4) *SA* specifies a chameleon hash function C_H with four sub-algorithm CKeyGen, CHash, CHashVer, and CHashCol. Their meanings were previously defined in Section III.
5) *SA* adds all system parameters and functions, $\{T_{(n)}(x), H, C_H\}$ and $\{PU_{GS_1}, PU_{GS_2}, \ldots\}$, to the genesis block so that every participant can have access to them.

Drone Enrollment: In this phase, the drone N_i ($i = 1, 2, \ldots$) enrolls in a set of tasks (data types) to complete (collect) at the ground station GS_k.

1) N_i uses its media access control (MAC) address as the real identification ID_i. In addition, N_i randomly selects PUF challenge cha_i, and computes the corresponding PUF response $res_i = F_{puf}(cha_i)$.
2) N_i chooses res_i as the secret key and computes the corresponding public key $PU_i = T_{(res_i)}(x)$.
3) N_i calculates its pseudo identification $PID_i = H(ID_i \| PU_i \| r)$, where r is a random number.
4) N_i runs CKeyGen(res_i) algorithm to produce a chameleon trapdoor key trk_i and a chameleon public key puk_i.
5) N_i shares $\{ID_i, PID_i, PU_i, puk_i, r\}$ with the ground station GS_k through a secure channel.
6) GS_k assigns N_i with a set of tasks T_i = [$t_{i.1}$, $t_{i.2}$, $\cdots$, $t_{i.z}$, $\cdots$] to complete, and shares T_i with N_i via a secure channel. Here, $t_{i.z}$ indicates the type z of data that N_i is required to be collected.
7) GS_k adds the transaction $\{ID_i, PID_i, PU_i, puk_i, r, T_i\}$ to a block through invoking the CreateTrans() function of the smart contract.
8) N_i stores cha_i, r, and T_i in the memory, but deletes res_i and trk_i for security reasons. N_i is also free to cache PID_i, PU_i, and puk_i for rapid access. However, in this paper we assume that N_i chooses to delete them for saving memory space.

Authentication and Key Negotiation: In this phase, the ground station GS_j (GS_j and GS_j might be located in different task zones) retrieves the information of drone N_i stored in the ledger to verify its identification and task eligibility, and negotiates a secret session key with it. Here, we assume that GS_j is a nearby ground station that N_i can communicate with.

1) N_i calculates the following $res_i = F_{puf}(cha_i)$, $PU_i = T_{(res_i)}(x)$, and $PID_i = H(ID_i \| PU_i \| r)$.
2) N_i selects a random number s, and computes the public Chebyshev polynomial $pol_i^{PU} = T_{(s \cdot res_i \cdot ID_i)}(x)$.
3) N_i invokes the ReadTrans() function of the smart contract to retrieve the public key PU_{GS_j} of GS_j.
4) N_i computes the secret Chebyshev polynomial $pol_i^{PR} = T_{(s \cdot res_i \cdot ID_i)}(PU_{GS_j})$.
5) N_i calculates the following
$$H_{i.1} = ID_i \oplus H(PID_i \| PU_i \| pol_i^{PU}),$$
$$H_{i.2} = t_{i.z} \oplus H(PID_i \| PU_i \| pol_i^{PU} \| ID_i),$$
$$H_{i.3} = s \oplus H(PID_i \| PU_i \| pol_i^{PU} \| ID_i \| t_{i.z}),$$
$$H_{i.4} = H(PID_i \| PU_i \| pol_i^{PU} \| ID_i \| t_{i.z} \| s \| pol_i^{PR}),$$
and then sends an authentication request message $req_i^{auth} := \{PID_i, pol_i^{PU}, H_{i.1}, H_{i.2}, H_{i.3}, H_{i.4}\}$ to GS_j via open wireless channels.
6) GS_j invokes the ReadTrans() function of the smart contract to retrieve the corresponding $ID_i{}'$, $PU_i{}'$, and $T_i{}'$ of $PID_i{}'$, calculates $ID_i{}'' = H_{i.1}{}' \oplus H(PID_i{}' \| PU_i{}' \| pol_i^{PU'})$, and compares the retrieved $ID_i{}' \stackrel{?}{=}$

the restored $ID_i^{''}$. If the evaluation is false, req_i^{auth} is denied and discarded. Otherwise, GS_j calculates $t_{i.z}^{'} = H_{i.2}^{'} \oplus H(PID_i^{'} \parallel PU_i^{'} \parallel pol_i^{PU'} \parallel ID_i^{'})$. If $t_{i.z}^{'} \notin T_i^{'}$, req_i^{auth} is denied and discarded. Otherwise, GS_j calculates $s^{'} = H_{i.3}^{'} \oplus H(PID_i^{'} \parallel PU_i^{'} \parallel pol_i^{PU'} \parallel ID_i^{'} \parallel t_{i.z}^{'})$.

7) GS_j calculates the secret Chebyshev polynomial $pol_{GS_j}^{PR} = T_{(PR_{GS_j})}(pol_i^{PU'})$, where $pol_{GS_j}^{PR} == pol_i^{PR}$. The proof is as follows:

$$\begin{aligned} pol_{GS_j}^{PR} &= T_{(PR_{GS_j})}(pol_i^{PU'}) \\ &= T_{(PR_{GS_j})}(T_{(s \cdot res_i \cdot ID_i)}(x)) \\ &= T_{(s \cdot res_i \cdot ID_i)}(T_{(PR_{GS_j})}(x)) \\ &= T_{(s \cdot res_i \cdot ID_i)}(PU_{GS_j}) \\ &= pol_i^{PR}. \end{aligned}$$

8) GS_j calculates $H_{i.4}^{'} = H(PID_i^{'} \parallel PU_i^{'} \parallel pol_i^{PU'} \parallel ID_i^{'} \parallel t_{i.z}^{'} \parallel s^{'} \parallel pol_{GS_j}^{PR})$, and compares $H_{i.4}^{'} \stackrel{?}{=} H_{i.4}$. If the comparison is false, req_i^{auth} is denied and discarded. Otherwise, the following steps are continued.
9) GS_j selects a random number u and computes the public Chebyshev polynomial $pol_{GS_j}^{PU} = T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(PU_i)$.
10) GS_j computes $H_{j.1} = H(GS_j \parallel PID_i^{'} \parallel t_{i.z}^{'} \parallel s^{'} \parallel pol_{GS_j}^{PU})$, replies an authentication response message $rep_j^{auth} := \{GS_j, pol_{GS_j}^{PU}, H_{j.1}\}$, and then calculates the secret session key $SK_{j.i} = T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(pol_i^{PU'})$.
11) N_i calculates $H_{j.1}^{'} = H(GS_j \parallel PID_i \parallel t_{i.z} \parallel s \parallel pol_{GS_j}^{PU\,'})$ and compares $H_{j.1}^{'} \stackrel{?}{=} H_{j.1}$. If the comparison is false, rep_j^{auth} is denied and discarded. Otherwise, N_i computes the secret session key $SK_{i.j} = T_{(s \cdot ID_i)}(pol_{GS_j}^{PU\,'})$. $SK_{j.i} == SK_{i.j}$, and the proof is as follows:

$$\begin{aligned} SK_{i,j} &= T_{(s \cdot ID_i)}(pol_{GS_j}^{PU\,'}) \\ &= T_{(s \cdot ID_i)}(T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(PU_i)) \\ &= T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(T_{(s \cdot ID_i)}(PU_i)) \\ &= T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(T_{(s \cdot ID_i)}(T_{(res_i)}(x))) \\ &= T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(T_{(s \cdot res_i \cdot ID_i)}(x)) \\ &= T_{(u \cdot PR_{GS_j} \cdot GS_j \cdot t_{i.z}^{'})}(pol_i^{PU}) \\ &= SK_{j,i}. \end{aligned}$$

After completing the above operations, the ground station GS_j and the drone N_i have verified each other's identity and successfully negotiated a secret session key for the type $t_{i.j}$ (or $t_{j.i}$) of data. The communication sequence diagram of authentication and key negotiation phase is shown in Fig. 6.

Fig. 6. The communication sequence diagram of authentication and key negotiation phase.

B. Chameleon Hash based Redactable Blockchain System

Blockchains are promoted as decentralized and immutable digital ledger systems without governing authorities by the National Institute of Standards and Technology (NIST) [29]. In blockchain networks, mutual trust between all participants are no longer needed for interactions because all participants acknowledge the identical multi-party consensus protocol to bring all participants into agreement on transactions. In addition, with the assistance of various cryptographic techniques (e.g., hash and digital signature functions), the transactions are uneditable once they are added into the blockchain ledger. The immutability property is the building block of establishing unprunable blockchains. The rationale is that a slight change to any transaction in a block will substantially affect all subsequent blocks on the chain. However, the permanent storage of transactions on the unprunable blockchain also raises storage concerns after a long period of operation and/or an exponential growth of participants. Needless to say, the immutability property is not in compliance with laws and regulations (e.g., General Data Protection Regulation (GDPR) [30]) in certain countries and regions of the world. For instance, the European Union GDPR states that users have the ''right to be forgotten (RTBF)'' which entitles users to have their private information removed from searches engines and directories. Last but not least, in the next-generation IoD applications drones are usually involved in collaborative tasks/missions which spread over different zones. Moreover, the drones' assigned tasks/missions could also be changed frequently. Thus, the IoD drones need to shuttle between different mission zones intermittently while updating their assigned tasks/missions, which drives the need for trust establishment solutions to realize cross-zone communications. To address the above storage and communication concerns, in this paper we design a redactable blockchain

system utilizing Hyperledger Fabric consortium blockchain [25] and chameleon hash function [22].

Consortium Blockchain Design: In the $ReBAS$, the ground stations from different task zones collaboratively form a consortium blockchain network, where a replica of ledger is held in each task zone. For simplicity, we assume that each task zone has two ground stations: one serves as the peer, and the other acts as the order. It is worth clarifying that dense deployment of ground stations is feasible, however, this strategy incurs high deployment and operational costs [31]. A ground station deployment strategy [32] might help to address the issue of constrained ground station resources, however, this topic is outside the scope of this paper. As the proposed consortium blockchain system is permissioned, the identities of ground stations are explicitly assigned by the membership service provider (MSP) module. In addition, the MSP module specifies the ground stations' roles (e.g., peer and order) along with ledger access privileges. Here, the peer is responsible for maintaining the replica of ledger and a smart contract instance, while the order accepts the responsibility for establishing a total sequence of transactions, generating new blocks, and seeking consensus. The peer can also serve an additional role, called endorser. After receiving a transaction proposal from an application, the endorsing peer simulates the execution of transaction based on the smart contract instance, produces an endorsement piggybacked with simulation results and a cryptographic signature, and sends the endorsement back to the application. When the application has obtained enough endorsements from peers on the transaction proposal, it assembles a transaction and submits it to the order. Then the order establishes consensus on transactions, batches multiple transactions into a block, and broadcasts the block to all peers in the network. Lastly, each peer validates the new block, reflects the changes of state on the local copy of ledger, and appends the block to its local blockchain ledger.

Add New Transactions: During the drone enrollment phase, the ground station assembles the drone's identity and task information into a transaction and adds it to the blockchain ledger. The flow of adding a new transaction is as follows.

1) The ground station sends a transaction proposal, which contains its identification and digital signature, the transaction ID, and the drone's identity and task information, to the peer in the task zone.
2) The peer validates the digital signature of the transaction proposer and simulates the execution of the proposed transaction against the local key-value state through invoking the smart contract instance. After the simulation is complete, the peer constructs the read-set which summarizes the version numbers of keys viewed by the smart contract instance, and the write-set which records the pairs of key-value changed by the smart contract instance.
3) The peer creates an endorsement which contains the result of simulated execution as well as the read/write-set, and sends it back to the ground station. The endorsement is digitally signed by the peer using its cryptographic key.
4) After the ground station obtains the required number of valid endorsements as indicated in the endorsement policy, it assembles the transaction including the drone's identity and task information as well as the set of obtained endorsements, and sends it to the order.
5) Either when the order receives a maximum permissible number of transactions (indicated by block size) or when the transaction-reception window (known as block time-out) closes, the order organizes all pending transactions chronologically in a newly created block. After that, the order broadcasts the new block piggybacked with its digital signature to all peers in the blockchain network.
6) Each peer verifies the set of endorsements with the assistance of validation system chaincode and filters out invalid endorsements along with the corresponding transactions. For each transaction with the valid endorsement, the peer checks whether the version of read-set is consistent with the current state on the key-value store, and reflects the write-set to the local key-value store to complete the state transition. Finally, the new block is appended on the local blockchain ledger at the peer; the ground station is notified about the success of adding the transaction to the blockchain ledger.

Edit Existing Transactions: When the drone is assigned with new tasks (data types), the previous task information stored in the blockchain ledger need to be updated so that the data type specific secret session key can be negotiated. The operations of updating the task information is as follows.

1) N_i uses its PUF challenge cha_i to compute the corresponding PUF response $res_i = F_{puf}(cha_i)$ which will be used to further calculate the public key $PU_i = T_{(res_i)}(x)$.
2) N_i calculates its pseudo identification $PID_i = H(ID_i \parallel PU_i \parallel r)$, where ID_i is its MAC address.
3) N_i runs CKeyGen(res_i) algorithm to produce a chameleon trapdoor key trk_i and a chameleon public key puk_i.
4) N_i and GS_z negotiate a generic (non-data-type-specific) secret session key by following the steps presented in the authentication and key negotiation phase, where the data type field that is used to calculate the hash value (e.g., $H_{i.2}$) of the authentication request message is replaced with dummy data type or null.
5) N_i sends a task update request message $req_i^{upd^*} := \{PID_i, ID_i, r, T_i^{new}, trk_i\}$ to GS_z via open wireless channels. $req_i^{upd^*}$ is encrypted using the secret session key negotiated with GS_z in the previous step. Thus, GS_z is the only participant that can decrypt $req_i^{upd^*}$.
6) GS_z decrypts $req_i^{upd^*}$ to obtain $PID_i^{''}$, $ID_i^{''}$, $r^{''}$, $T_i^{new^{''}}$, and $trk_i^{''}$, invokes the ReadTrans() function of the smart contract to retrieve the corresponding $ID_i^{'}$, $r^{'}$, and $T_i^{'}$ of $PID_i^{''}$, and then compares the retrieved $ID_i^{'} \stackrel{?}{=}$ the decrypted $ID_i^{''}$ as well as the retrieved $r^{''} \stackrel{?}{=}$ the decrypted $r^{'}$. If either evaluation is false, $req_i^{upd^*}$ is denied and discarded. Otherwise, GS_z runs CHashCol($trk_i^{''}$, $T_i^{'}$, $r^{'}$, h, $T_i^{new^{''}}$) to generate a new random number v corresponding to the new set of data types $T_i^{new^{''}}$ for the same hash value h.
7) GS_z broadcasts the new blockchain ledger on the network. In compliance with previously-agreed redacting

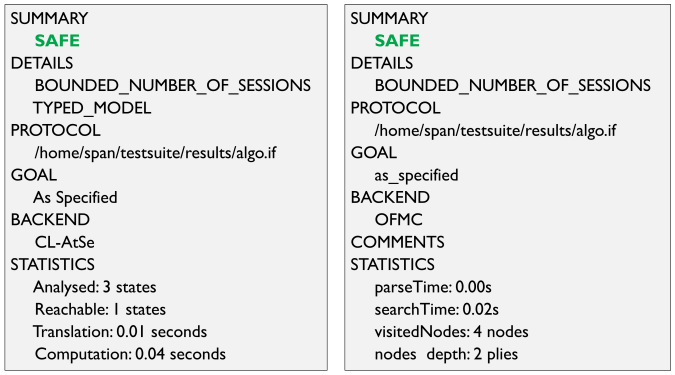


Fig. 7. Security verification results from AVISPA.

rules, every ground station shall adopt the new chain, even if it stores a longer one.

VI. Security Verification And Analysis

In this section, we concentrate on the security assessment of our protocol, aiming to validate its security features and operational integrity within a hostile setting through security verification, as well as formal and informal security analysis.

A. Security Verification

A widely recognized Internet security protocol verification framework, called AVISPA [6], is adopted to verify the safeness of our protocol. The goal of the security verification is to create an exhibition showing there are no security defects in our protocol which could be exploited by the adversary to attack IoD systems. In order to conduct the security verification on AVISPA, our protocol is implemented in HLPSL language [33]. In the HLPSL program, the drone and the ground station are assigned a role respectively between which message exchanges are performed. After that, the HLPSL program is executed using two major verification components of AVISPA, On-the-fly Model Checker (OFMC) and the Constraint-Logic-based Attack Searcher (CL-AtSe). Here, OFMC is particularly useful for examining the protocol's security characteristics such as authenticity, confidentiality, and integrity, while CL-AtSe is suitable for vulnerability assessment and threat modeling. The home operating system for the verification framework is Ubuntu 10.04, where AVISPA is installed and configured in Virtual Box. The results of AVISPA security verification are documented in Fig. 7, which demonstrates that our protocol is desired to achieve goals of robustness and security.

B. Formal Security Analysis

We also choose Mao's and Boyd's theoretical security analysis framework [7] to formally analyze our protocol in this subsection. The aim of formal security analysis is to exhibit that the drone (e.g., N_i) and the ground station (e.g., GS_j) are only two network entities who are granted to access the secret information, e.g., Chebyshev polynomial pol_i^{PR}. In other words, the secret information is impervious to unauthorized access, acquisition, or manipulation. The basic idea of formal security analysis is first introducing a set of inference rules for deductive reasoning about logical formulas, and then outlining

TABLE III
Formal Security Analysis Operators

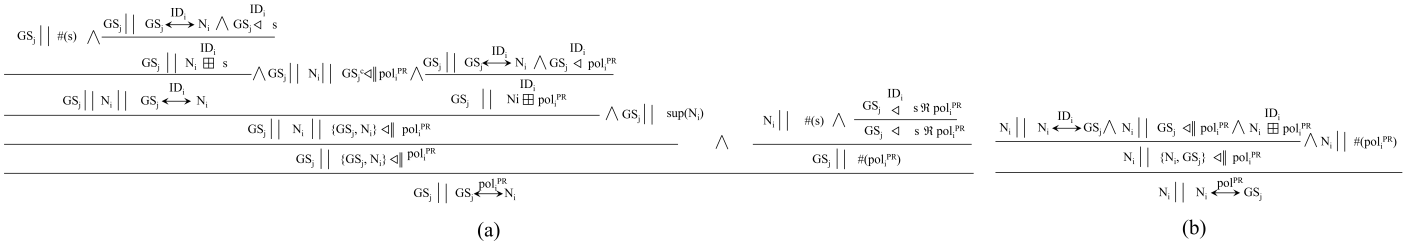
Operator	Meaning
$P \models X$	The principal P believe the statement X to be true
$P \triangleleft \parallel M$	The principal P cannot see the message M
#(M)	The message M is fresh
$P \overset{(K)}{\boxplus} M$	The principal P sees the message M by using the key K
$\bigwedge$	A boolean logical conjunctor

a series of foundational assumptions that constitute reasonable beliefs under which the essential communication activities are realized. The meanings of formal security analysis operations are summarized in Table III.

First of all, we make a set of initial belief statements as required by Mao's and Boyd's theoretical security analysis framework using the rules of inference.

1) $N_i \models N_i \overset{pol_i^{PU}}{\longleftrightarrow} GS_j$ and $GS_j \models GS_j \overset{pol_i^{PU}}{\longleftrightarrow} N_i$: The public Chebyshev polynomial (pol_i^{PU}) is shared between drone N_i and ground station GS_j through a secured channel during the initial stage.
2) $N_i \models N_i \overset{PID_i}{\longleftrightarrow} GS_j$ and $GS_j \models GS_j \overset{PID_i}{\longleftrightarrow} N_i$: The pseudonym PID_i is shared between drone N_i and ground station GS_j. PID_i is calculated by drone N_i using its real identification ID_i and public key PU_i along with a random number r.
3) $N_i \models GS_j \triangleleft \parallel ID_i$ and $GS_j \models N_i \models \{GS_j\} \triangleleft \parallel ID_i$: Drone N_i has a unique identification ID_i which is considered as drone N_i's secret information. During the communication, instead of using the real identification ID_i, drone N_i will calculate a pseudonym PID_i. As the trusted entity, ground station GS_j also have access to drone ID_i's ID_i.
4) $GS_j \models \text{sup}(N_i)$: Drone N_i is the super-principal to ground station GS_j.
5) $GS_j \models \#(u)$: Ground station GS_j generates a fresh random number u for each communication session.
6) $N_i \models \#(s)$: Drone N_i generates a fresh random number s for each communication session.
7) $GS_j \overset{(ID_i)}{\boxplus} pol_i^{PR}$: Ground station GS_j reads the message piggybacked with drone N_i's ID_i using drone N_i's private Chebyshev polynomial pol_i^{PR}.
8) $GS_j \overset{(Pol_i^{PR}, ID_i)}{\triangleleft} ID_i$: GS_j obtains drone N_i's Pol_i^{PR} through the calculation of its secret Chebyshev polynomial $pol_{GS_j}^{PR}$ using drone N_i's identification.
9) $GS_j \overset{(ID_i)}{\boxplus} s$: Ground station GS_j encrypts the message piggybacked with s using its public Chebyshev polynomial $pol_{GS_j}^{PU}$. Ground station GS_j obtains s from the encrypted messages from drone N_i.
10) $N_i \overset{(pol_i^{PR}, s)}{\triangleleft} t'_{i,z}$: Drone N_i decrypts the message using drone N_i's private Chebyshev polynomial Pol_i^{PR} and random number s.

In Fig. 8 we demonstrate that drone N_i and ground station GS_j are the only two communication entities who are involved in the communication and able to access the secret Chebyshev polynomial $pol_i^P R$. First, based on $N_i \models N_i \overset{pol_i^{PR}}{\longleftrightarrow} GS_j$ we



(a) (b)

Fig. 8. Formal security analysis. (a) The proof that pol_i^{PR} is believed to be confidential from the perspective of ground station GS_j. (b) The proof that pol_i^{PR} is believed to be confidential from drone N_i point of view.

apply the Good key rule to the fact that drone N_i's real identification is only known to drone N_i itself and ground station GS_j according to $N_i \models GS_j \triangleleft \parallel ID_i$. Moreover, drone N_i only uses its pseudonym PID_i in the communication messages which will be sent to ground station GS_j. Since drone N_i and ground station GS_j believe that drone N_i's real identification ID_i is a good shared secret between them, it can be inferred that the secret Chebyshev polynomial Pol_i^{PR} is a good shared secret as well based on the proved relationship among these system parameters. Then, we can apply the Confidentiality Rule to make the inference that the secret Chebyshev polynomial Pol_i^{PR} and drone N_i's real identification ID_i are confidential information based on the previously proved rule $GS_j \models \sup(N_i)$. Finally, since drone N_i's pseudonym PID_i is calculated using its real identification ID_i, public key PU_i, and random number r, it is impossible for an adversary to guess drone N_i's real identification ID_i. Thus, Pol_i^{PR} is a secure secret. If drone N_i is physically captured, the adversary still will not be able to have access to pol_i^{PR}. This is because the public key PU_i is calculated using the response of PUF which is resistant to any physical attacks.

C. Informal Security Analysis

Our protocol is designed based on several cryptographic primitives such as Chebyshev polynomials, PUFs, hash function, redactable blockchain, and chameleon hash function to achieve its desired security goals and objectives in the cyber-threat environment. In the following, we justify how our protocol can protect IoD systems against well-known cyberattacks.

Message Fabrication Attacks: Our protocol is secure against message fabrication attacks by utilizing a secure hash function to generate a hash value for the entire message, which is able to prevent message interception and tampering. For example, the drone N_i calculates a hash value $H_{i.4}$ using all to-be-shared parameters and piggybacks $H_{i.4}$ in the authentication request message which is sent to the ground station GS_j. After receiving the authentication request message, the ground station GS_j will verify the authenticity of $H_{i.4}$ and check whether the message has been fabricated or tampered. Additionally, recording drones' identity and task information into a blockchain transaction provides immutability to those cryptographic information. Based on the above discussion, it is impossible for attackers to fabricate the messages or cryptographic information in our protocol.

Physical Tampering Attacks: Even if a drone is physically captured by an attacker, our protocol still can guarantee its security and privacy because of the adoption of PUFs. For example, the drone N_i will generates a PUF response res_i using a PUF challenge cha_i, which is regarded as a hardware-bounded secret information that is impossible to predict or clone. Moreover, PUFs are completely designed based on the physical characteristics of drone's integrated circuit. When the attacker is probing the integrated circuit of drone to retrieve any cryptographic information stored in the memory, the PUF will be destroyed and the original PUF response cannot be reproduced. In summary, our protocol can help drones defend against physical tampering attacks.

Replay Attacks: In the replay attacks, the attackers attempt to intercept and retransmit messages to trick the receivers in the IoD systems. In our protocol, a random number is piggybacked in each message to prevent IoD systems from replay attacks. For example, when the drone N_i plans to send the authentication request message to the ground station GS_j, it will calculate $H_{i.3}$ which includes a random number s. When the ground station GS_j receives the authentication request message, it will verify the freshness of the message and discards the message if it is a replayed message.

Known Session Key Attacks: Known session key attacks occur when an attacker gains access to the session keys of past communication sessions. In our protocol, each session key is uniquely created using Chebyshev polynomial and random number. For example, the session key $SK_{j.i}$ is calculated as $T(u \cdot PR_{GS_j} \cdot GS_j \cdot t'i.z)(pol_i^{PU'})$, which includes unique session-specific parameters, making it infeasible for an attacker to use previously-known session keys to decrypt new messages.

Impersonation Attacks: In the impersonation attacks, an attacker pretends to be either a legitimate drone or a ground station. However, our protocol can mitigate such attacks in the IoD systems. This is because the drone N_i and the ground station GS_j will verify the harsh values (e.g., $H_{i.4}$ and $H_{j.1}$) using their respective keys and pseudo identities. Since the keys and pseudo identities are derived from their secret keys and unique identifications, only legitimate entities can produce valid harsh values, which can successfully prevent impersonation attacks.

Eavesdropping Attacks: Our protocol can protect IoD systems from eavesdropping attacks by securely encrypting messages with session keys derived from Chebyshev polynomials and random numbers. Since the session keys are computed using unique and session-specific information, an attacker cannot decrypt the captured messages without knowing the private keys and random numbers used in the Chebyshev

Simulation Setting

Entry	Value/Name
Programming Language	Python 3
Simulation Platform	Apple MacBook Pro Laptop
Laptop Processor	Apple M3 Pro chip
Number of Drones	20 to 100
Blockchain System	Hyperledger Fabric
Number of Task Transactions	20 to 100
Benchmark Schemes	USAF-IoD and PAF-IoD

polynomial calculations.

In summary, our protocol meets all stringent security and privacy requirements through incorporating various cryptographic primitives, redactable consortium blockchain, and chameleon hash function. Through these combined techniques, our protocol provides a robust framework for secure communication in the IoD systems.

VII. Performance Evaluation

This section presents a comprehensive performance evaluation of our approach $ReBAS$ which is specifically designed for next-generation IoD applications. The $ReBAS$ can not only achieve data type aware authentication between drones and ground stations but also provide the cross-domain communication feature. In addition, the $ReBAS$ is realized with the integration of advanced techniques such as Chebyshev polynomial, redactable consortium blockchain, and chameleon hash function to significantly reduce the computation, communication, and storage overheads of cryptography-related operations. In order to show the superior performance of our approach $ReBAS$, we implement other two state-of-the-art approaches, e.g., USAF-IoD [18] and PAF-IoD [19], and compare them with the $ReBAS$. All three approaches are implemented using Python 3 and executed on Apple MacBook Pro laptop (Apple M3 Pro chip; 11-Core CPU, 14-Core GPU, 18GB Unified Memory). The simulation parameters are summarized in Table VII.

In order to evaluate the performance of $ReBAS$, USAF-IoD, and PAF-IoD, we choose several quantified performance metrics such as CPU time, energy consumption, authentication latency, authentication scalability index, communication cost, blockchain execution time, and blockchain storage cost. The definition of each performance metric is provided below.

- The CPU time is the amount of time the processor spends actively processing the operations of the algorithm.
- The energy consumption refers to the amount of energy (Joules) used by the processor to execute the algorithm.
- The authentication latency is measured as the amount of time elapsed from when the drone initiates the authentication process to when the authentication process is completed by the ground station.
- The authentication scalability index (ASI) is calculated as the variation of drone authentication time divided by the variation of drone authentication requests.
- The communication cost refers to the size of transmitted messages (KB) during the authentication process.
- The blockchain execution time is defined as the duration it takes for a transaction to be processed.

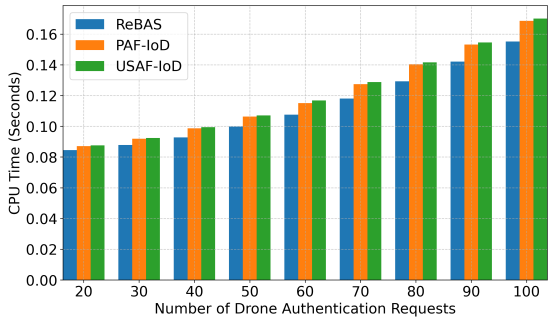


Fig. 9. The performance of CPU time against the number of drone authentication requests.

- The blockchain storage cost indicates the amount of storage required to store drone's cryptographic and task information on a blockchain network.

In addition, the fundamental ideas and achievements of USAF-IoD [18] and PAF-IoD [19] are summarized below:

- The USAF-IoD proposes a key agreement framework for the IoD environment, where the drones, the users, and the server can mutually authenticate each other to defend against cyberattacks such as physical capture, impersonation, and replay attacks. The major techniques which are used to achieve the security objectives of USAF-IoD are authenticated encryption, hash functions, XOR operations, and physical unclonable functions (PUFs). The USAF-IoD seems not to suffer from common security vulnerabilities based on the security analysis. An experimental study has also been conducted to show that the USAF-IoD can achieve lower computation and communication overhead compared to the benchmark schemes.
- The PAF-IoD introduces a lightweight authentication system for the IoD networks. By incorporating PUFs into the authentication process, the PAF-IoD is able to guarantee the uniqueness and tamper-resistance of drones, which offers robust security against physical attacks. As the PUFs are not stable in the harsh environment, the PAF-IoD also incorporates fuzzy extractors into PUFs to improve their reliability. The PAF-IoD demonstrates its security and robustness against impersonation and man-in-the-middle attacks by means of security analysis. In addition, the PAF-IoD seems to be more efficient in terms of computation and communication costs while comparing with other approaches.

First, we measure and analyze the performance of $ReBAS$, USAF-IoD, and PAF-IoD in terms of CPU time and energy consumption with a varying number of drone authentication requests in Fig. 9. Overall, the $ReBAS$ demonstrates a clear CPU time performance advantage over USAF-IoD and PAF-IoD with a varying number of drone authentication requests. As the number of drone authentication requests is increased from 20 to 100 in the system, the CPU time of all three approaches increase linearly. Nevertheless, our approach $ReBAS$ still shows a significantly lower CPU time compared to USAF-IoD and PAF-IoD. This is because the $ReBAS$ adopts a cost-effective cryptographic primitive Chebyshev

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3562857

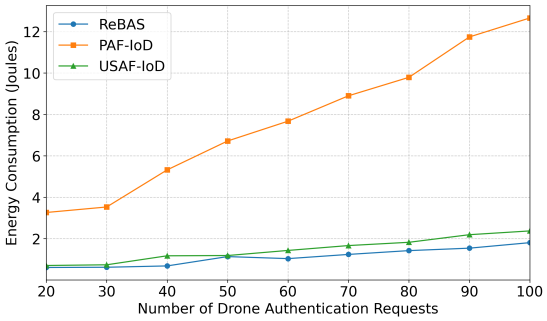


Fig. 10. The performance of energy consumption against the number of drone authentication requests.

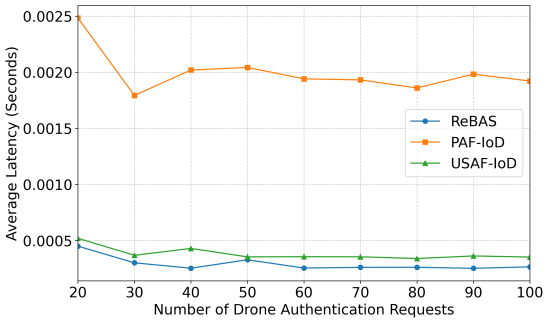


Fig. 11. The performance of average authentication latency against the number of drone authentication requests.

polynomial to enable mutual authentication. In addition, the CPU time differential between the $ReBAS$ and other two approaches becomes significant as the number of drone authentications scales. When the number of drone authentications reaches up to 100, the $ReBAS$ takes up 0.171 seconds of CPU time, in contrast to 0.185 seconds of CPU time which is consumed by the USAF-IoD. This translates to a 7.57% improvement in CPU efficiency by our approach $ReBAS$. In a nutshell, the $ReBAS$ can elevate the performance of CPU time distinctly across numerous drone authentication scenarios, and the performance advantage becomes more pronounced under substantial drone authentication load.

Second, we obtain the energy consumption of $ReBAS$, USAF-IoD, and PAF-IoD in Fig. 10, where the number of drone authentication requests is changed between 20 and 100. Here, we use PyRAPL [34] to measure the energy consumption during code execution. Fig. 10 clearly illustrates that the energy consumption patterns vary significantly among the three approaches. However, the $ReBAS$ exhibits superior energy efficiency with different number of drone authentication requests, where USAF-IoD showing moderate energy efficiency and PAF-IoD exhibiting the highest energy demand due to its complex computations. Most importantly, the energy efficiency differential between the $ReBAS$ and the PAF-IoD widens as the number of drone authentication requests increases. With 100 drone authentication requests, the $ReBAS$ consumes approximately 1.8 joules, compared to 2.3 joules and 10.5 joules consumed by the USAF-IoD and PAF-IoD, respectively.

Third, the average authentication latency results of $ReBAS$, USAF-IoD, and PAF-IoD are shown in Fig. 11. Overall,

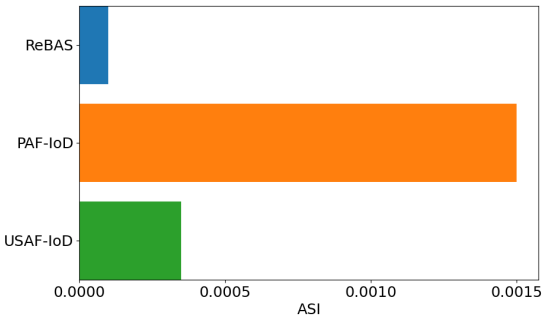


Fig. 12. The performance of authentication scalability index against the number of drone authentication requests.

the authentication latency plays a vital role in real-time IoD systems, directly influencing system responsiveness and overall performance. To put it plainly, the authentication latency indicates how long it takes for a drone to be authenticated and securely communicate with the ground station. The average authentication latency is calculated as the total authentication latency divided by the number of drone authentication requests. In Fig. 11, as the number of drone authentication requests increases from 20 to 100, all three approaches exhibit a relatively stable performance in the average authentication latency. The average authentication latency tends to stabilize as the number of drones increases because of the concept of the law of large numbers in statistics. Essentially, as the sample size grows, the effects of random variations diminish, and the average converges closer to the true population mean. This happens because extreme authentication latency (e.g., very high and very low ones) get balanced out by the larger number of typical authentication latency. In simpler terms, with more drones, the collective performance becomes more representative of the entire group, reducing the influence of outliers. This is why the average authentication latency does not fluctuate much when a larger group of drones is analyzed. The best latency performance belongs to our approach $ReBAS$ which maintains an average authentication latency below 0.3 milliseconds even though the number of drone authentication requests increases to 100. The authentication latency of PAF-IoD is significantly higher than that of $ReBAS$ and USAF-IoD, averaging around 0.0016 seconds, which is approximately 5.3 times higher than the $ReBAS$. The USAF-IoD performs better than the PAF-IoD but still shows higher latency compared to the $ReBAS$, with an average around 0.35 milliseconds. In summary, the low authentication latency demonstrated by our approach $ReBAS$ represents a significant advancement in the field of IoD authentication. To be specific, the quicker authentication processes can make IoD systems facilitate more responsive system interactions and handle more drone authentication requests within the given time frame. This performance improvement is particularly valuable in scenarios which require rapid authentication of multiple drones in the large-scale or high-traffic urban environments.

Fourth, we measure the authentication scalability index (ASI) of $ReBAS$, USAF-IoD, and PAF-IoD in Fig. 12. In this article, the ASI indicates how well the approach maintains the authentication latency as the number of drone authentication

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3562857

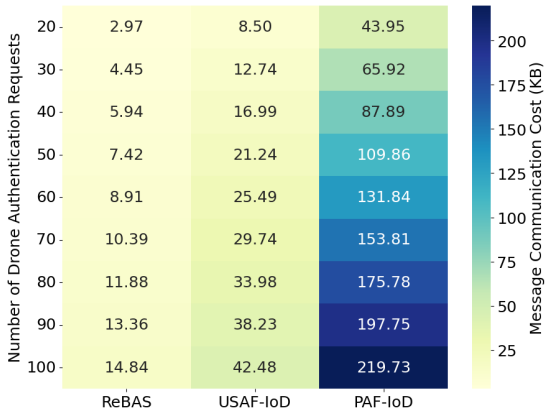


Fig. 13. The performance of communication cost against the number of drone authentication requests.

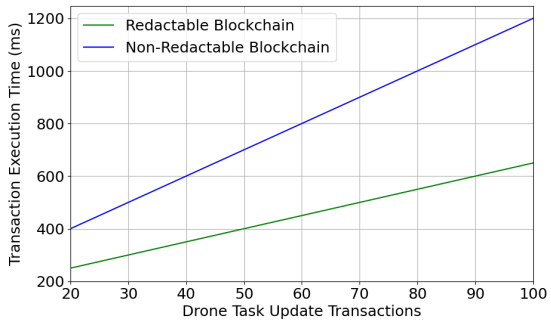


Fig. 14. The performance of blockchain execution time against the number of task update transactions.

requests increases. In other words, the ASI represents an increase in authentication time for each additional drone. The lower the ASI is, the better scalability the approach has. As shown in Fig. 12, our approach $ReBAS$ achieves the lowest ASI value of 0.0003 compared to that of PAF-IoD and USAF-IoD. The PAF-IoD exhibits the highest ASI value of 0.0018, which is 6 times higher than $ReBAS$. With an ASI value of 0.0004, the USAF-IoD shows better scalability than the PAF-IoD but still falls short in terms of the scalability demonstrated by the $ReBAS$. In brief, the $ReBAS$ is able to accommodate a larger number of drone authentication requests without compromising the efficiency of authentication approach. This is crucial for efficiently supporting large-scale IoD operations in the urban environment. Moreover, the superior scalability of $ReBAS$ indicate more efficient use of existing computational resources, which in turn can reduce the number of ground stations deployed in the aera.

Fifth, we obtain the communication cost results of all three approaches by changing the number of drone authentication requests in Fig. 13. The horizontal axis shows three approaches (e.g., $ReBAS$, USAF-IoD, and PAF-IoD), while the vertical axis represents the number of drones in the network, ranging from 20 to 100. The color gradient provides a visual indication of communication cost, lighter colors signifying lower communication cost and darker colors indicating higher communication overhead. As shown in Fig. 13, our approach $ReBAS$ consistently incurs the lowest communication cost with different number of drones in the network. For example, when there 20 drones in the network, the $ReBAS$ incurs a communication overhead of 2.97 KB, which slowly increases to 14.84 KB when the number of drones reaches 100. The lower communication cost achieved by our approach $ReBAS$ will bring many benefits to the IoD applications. For instance, it helps reduce bandwidth usage and ensures that the application can efficiently scale to accommodate larger IoD networks. In contrast, the USAF-IoD shows moderate performance in terms of communication cost. It starts with a communication overhead of 8.50 KB for 20 drones, and increases to 42.48 KB for 100 drones. Although the USAF-IoD performs better than the PAF-IoD, its communication cost is still considerably higher than that of $ReBAS$, particularly as the IoD system scales. The PAF-IoD, on the other hand, exhibits the highest communication cost across all scenarios. For 20 drones, the communication overhead is already quite high at 43.95 KB, and this number increases dramatically to 219.73 KB when there 100 drones in the network. The dark shading of the heatmap cells visually illustrates the significant communication overhead associated with the PAF-IoD. This high communication cost suggests that the PAF-IoD may not be suitable for large-scale IoD applications. This is because the excessive bandwidth consumption could lead to network congestion, delays, and inefficiencies in real-time IoD operations.

Sixth, the performance of blockchain execution time is presented in Fig. 14, where the number of task update transactions is changed between 20 and 100. It is clearly shown that our approach integrated with redactable blockchain demonstrates the superior computational performance compared to non-redactable blockchain. With 100 task update transactions, the redactable blockchain completes all task update requests in approximately 500 milliseconds, while the non-redactable blockchain requires about 1500 milliseconds. Thus, our approach $ReBAS$ can achieve a 66.7% reduction in blockchain execution time. This significant blockchain execution time reduction is attributed to the elimination of cascading hash recalculations typically required in traditional blockchain architectures. Finally, we show the performance of blockchain storage cost against the number of task update transactions in Fig. 15. As the number of drone task update increases, the storage cost for the redactable blockchain grows at a substantially slower rate compared to its non-redactable counterpart. With 100 drone task update requests, the $ReBAS$ consumes only 50 KB of storage, while the non-redactable blockchain requires 140 KB storage. In other words, our approach $ReBAS$ will save 64.3% of storage. This remarkable storage reduction is achieved through our innovative approach to data management within the redactable blockchain structure, allowing for in-place updates that minimize redundant data storage.

VIII. Discussion

In the proposed Internet of Drones (IoD) framework, the ground region and its overlying airspace are partitioned into designated task zones. IoD drones traverse these zones, gathering environmental and Internet of Things (IoT) sensory data, which is subsequently delivered to stationary networking

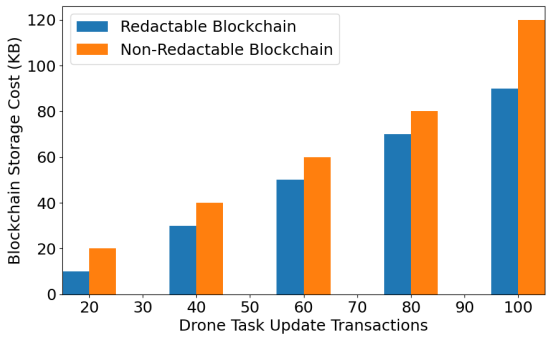


Fig. 15. The performance of blockchain storage cost against the number of task update transactions.

systems for comprehensive analysis. If a large group of IoD drones communicates with the ground station for authentication and key establishment simultaneously, a severe authentication signaling congestion will occur at the ground station. As a result, the IoD drones might be faced with authentication failure or even suffer denial of service, and the overall quality of service (QoS) is adversely affected. To properly address the authentication signaling congestion issue, a group or aggregated authentication protocol might be needed. As the primary focus of this manuscript is a redactable blockchain-assisted application-aware authentication system for IoD systems, the group or aggregated authentication protocols fall outside the scope of this manuscript and will be proposed as a topic for future research.

IX. Conclusion

This paper introduced the $ReBAS$, a novel redactable blockchain-assisted application-aware authentication system, to address critical security, privacy, and storage challenges for next-generation Internet of Drones (IoD) systems. By integrating Chebyshev polynomial, redactable consortium blockchain, and chameleon hash function, the $ReBAS$ can significantly reduce computation, communication, and storage overheads associated with cryptography-related operations. In addition, the $ReBAS$ offer several crucial benefits to the IoD systems. First, the $ReBAS$ distinguishes data types during the authentication process, which allows the IoD systems to establish unique secret session keys for different types of data, effectively mitigating potential data leakage risks. Second, the integration of a redactable blockchain system in the $ReBAS$ can enable efficient updates of drone task information without causing an increase in the storage cost. Through comprehensive security verification using the AVISPA tool and formal security analysis based on Mao's BAN logic, we demonstrated the robustness of $ReBAS$ against various cyber threats, ensuring its compliance with the requirements of Canetti-Krawczyk security framework. Moreover, extensive performance evaluations revealed $ReBAS$'s superior performance in terms of CPU time, energy consumption, authentication latency, scalability, and storage cost compared to state-of-the-art approaches. As the IoD landscape continues to evolve, the $ReBAS$ will play a crucial role in enabling secure, privacy-preserving, and resource-efficient drone operations across diverse sectors, from urban air mobility to large-scale environmental monitoring. Future research directions may include exploring the integration of $ReBAS$ with emerging IoD applications and investigating its performance in heterogeneous IoD environments, further extending its applicability and impact on the field of IoD security and privacy.

References

[1] S. Krishnan and M. Murugappan, *Internet of Drones: Applications, Opportunities, and Challenges*. CRC Press, 2023.
[2] *Flying Cars Market Size Worth USD 1533.471 Billion, Globally, by 2040 at 58.01% CAGR*, https://www.fortunebusinessinsights.com/flying-cars-market-105378 (Accessed: Nov 23, 2023).
[3] *The world's first airport for flying cars and drones has just landed*, https://www.weforum.org/agenda/2022/04/urban-airport-flying-cars-drones (Accessed: Nov 23, 2023).
[4] *Air Taxis, Hyped for Years, May Finally Take Off*, https://www.nytimes.com/2023/07/18/business/air-taxi-faa.html (Accessed: Nov 23, 2023).
[5] Q. Long, J. Ma, F. Jiang, and C. Webster, "Demand analysis in urban air mobility: A literature review," *Journal of Air Transport Management*, vol. 112, p. 102436, 2023.
[6] *Automated Validation of Internet Security Protocols and Applications*, https://www.ercim.eu/publication/Ercim_News/enw64/armando.html (Accessed: May 28, 2024).
[7] W. Mao and C. Boyd, "Towards Formal Analysis of Security Protocols," in *Proc. IEEE CSFW*, 1993, pp. 147–158.
[8] M. Tanveer, H. Alasmary, N. Kumar, and A. Nayak, "SAAF-IoD: Secure and Anonymous Authentication Framework for the Internet of Drones," *IEEE Transactions on Vehicular Technology (Early Access)*, pp. 1–13, 2023.
[9] D. Chaudhary, T. Soni, K. Vasudev, and K. Saleem, "A modified lightweight authenticated key agreement protocol for Internet of Drones," *Internet of Things*, vol. 21, p. 100669, 2023.
[10] T. Lee, D. Lou, and C. Chang, "Enhancing lightweight authenticated key agreement with privacy protection using dynamic identities for Internet of Drones," *Internet of Things*, vol. 23, p. 100877, 2023.
[11] M. El-Zawawy, A. Brighente, and M. Conti, "SETCAP: Service-based energy-efficient temporal credential authentication protocol for Internet of Drones," *Computer Networks*, vol. 206, p. 108804, 2022.
[12] J. García, A. Benslimane, A. Braeken, and Z. Su, "μTesla-based Authentication for Reliable and Secure Broadcast Communications in IoD using Blockchain," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18 400–18 413, 2023.
[13] S. Yu, J. Lee, A. Sutrala, A. Das, and Y. Park, "LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad-hoc networks," *Computer Networks*, vol. 224, p. 109612, 2023.
[14] N. Jadav, T. Rathod, R. Gupta, S. Tanwar, N. Kumar, R. Iqbal, S. Atalla, H. Mohammad, and S. Al-Rubaye, "Blockchain-Based Secure and Intelligent Data Dissemination Framework for UAVs in Battlefield Applications," *IEEE Communications Standards Magazine*, vol. 7, no. 3, pp. 16–23, 2023.
[15] M. Akram, H. Ahmad, A. Mian, A. Jurcut, and S. Kumari, "Blockchain-based privacy-preserving authentication protocol for UAV networks," *Computer Networks*, vol. 224, p. 109638, 2023.
[16] A. Berini, M. Ferrag, B. Farou, and H. Seridi, "HCALA: Hyperelliptic curve-based anonymous lightweight authentication scheme for Internet of Drones," *Pervasive and Mobile Computing*, vol. 92, p. 101798, 2023.
[17] P. Bagchi, R. Maheshwari, B. Bera, A. Das, Y. Park, P. Lorenz, and D. Yau, "Public Blockchain-Envisioned Security Scheme Using Post Quantum Lattice-Based Aggregate Signature for Internet of Drones Applications," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, pp. 10 393–10 408, 2023.
[18] A. Badshah, G. Abbas, M. Waqas, S. Tu, Z. Abbas, F. Muhammad, and S. Chen, "USAF-IoD: Ultralightweight and Secure Authenticated Key Agreement Framework for Internet of Drones Environment," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 10 963–10 977, 2024.
[19] M. Tanveer, A. Aldosary, S. Khokhar, A. Das, S. Aldossari, and S. Chaudhry, "PAF-IoD: PUF-Enabled Authentication Framework for the Internet of Drones," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 9560–9574, 2024.
[20] E. Rescorla, *RFC2631: Diffie-Hellman Key Agreement Method*, 1999.

[21] E. Weisstein, *Chebyshev Polynomial of the First Kind*, 2003, https://mathworld.wolfram.com/ChebyshevPolynomialoftheFirstKind.html (Accessed: Nov 28, 2023).
[22] H. Krawczyk and T. Rabin, "Chameleon Hashing and Signatures," in *Cryptology ePrint Archive*, 1998.
[23] T. Ye, M. Luo, Y. Yang, K. Choo, and D. He, "A Survey on Redactable Blockchain: Challenges and Opportunities," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1669–1683, 2023.
[24] D. Derler, K. Samelin, D. Slamanig, and C. Striecks, "Fine-grained and controlled rewriting in blockchains: Chameleon-hashing gone attribute-based," in *Cryptology ePrint Archive*, 2019.
[25] E. Androulaki, C. Cachin, C. Ferris, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, and C. Stathakopoulou, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proc. ACM EuroSys*, 2018, pp. 1–15.
[26] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," in *Proc. Springer EUROCRYPT*, 2001, pp. 453–474.
[27] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution," in *Proc. Springer CRYPTO*, 1993, pp. 232–249.
[28] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm," in *The Internet Engineering Task Force, RFC 6238*, 2011.
[29] D. Yaga, P. Mell, N. Roby, and K. Scarfone, *Blockchain Technology Overview*, 2019, https://doi.org/10.6028/NIST.IR.8202 (Accessed: Dec 12, 2023).
[30] *General Data Protection Regulation (GDPR)*, 2018, https://gdpr-info.eu/ (Accessed: Dec 12, 2023).
[31] C. Pu and L. Carpenter, "Psched: A Priority-Based Service Scheduling Scheme for the Internet of Drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4230–4239, 2021.
[32] B. Liang, W. Lu, and B. Ran, "Deploying Roadside Unit Efficiently in VANETs: A Multi-Objective Delay-Based Optimization Strategy Using Lagrangian Relaxation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2023.
[33] D. V. Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. APPSEM Workshop*, 2005, pp. 1–17.
[34] *PyRAPL*, https://pypi.org/project/pyRAPL/ (Accessed: March 21, 2025).