

Article

Decentralized Blockchain-Based Authentication and Interplanetary File System-Based Data Management Protocol for Internet of Things Using Ascon

Hiba Belfqih *  and Abderrahim Abdellaoui

Engineering Sciences Laboratory, Ibn Tofail University, Kenitra 14000, Morocco; abderrahim@uit.ac.ma

* Correspondence: hibabelfqih2000@gmail.com

Abstract: The increasing interconnectivity of devices on the Internet of Things (IoT) introduces significant security challenges, particularly around authentication and data management. Traditional centralized approaches are not sufficient to address these risks, requiring more robust and decentralized solutions. This paper presents a decentralized authentication protocol leveraging blockchain technology and the IPFS data management framework to provide secure and real-time communication between IoT devices. Using the Ethereum blockchain, smart contracts, elliptic curve cryptography, and ASCON encryption, the proposed protocol ensures the confidentiality, integrity, and availability of sensitive IoT data. The mutual authentication process involves the use of asymmetric key pairs, public key registration on the blockchain, and the Diffie–Hellman key exchange algorithm to establish a shared secret that, combined with a unique identifier, enables secure device verification. Additionally, IPFS is used for secure data storage, with the content identifier (CID) encrypted using ASCON and integrated into the blockchain for traceability and authentication. This integrated approach addresses current IoT security challenges and provides a solid foundation for future applications in decentralized IoT environments.

Keywords: IoT security; decentralized authentication; blockchain; interplanetary file system (IPFS); ASCON encryption; smart contracts; decentralized data management; elliptic curve cryptography (ECC)



Received: 9 March 2025

Revised: 9 April 2025

Accepted: 15 April 2025

Published: 23 April 2025

Citation: Belfqih, H.; Abdellaoui, A. Decentralized Blockchain-Based Authentication and Interplanetary File System-Based Data Management Protocol for Internet of Things Using Ascon. *J. Cybersecur. Priv.* **2025**, *5*, 16. <https://doi.org/10.3390/jcp5020016>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) has witnessed unprecedented growth in recent years, with billions of interconnected devices transforming various domains such as smart homes, healthcare systems, industrial automation, and urban infrastructures [1]. These devices facilitate seamless data collection, processing, and sharing, enabling smarter and more efficient systems. However, this pervasive interconnectivity introduces a host of security challenges, particularly in the domains of authentication, data integrity, and confidentiality [2,3]. The highly distributed and resource-constrained nature of IoT environments makes them especially susceptible to cyber threats such as data breaches, impersonation attacks, and denial-of-service attacks [4].

Traditional centralized security mechanisms, including cloud-based authentication systems, have been widely adopted to address these challenges [5]. However, their reliance on central authority introduces several vulnerabilities. They suffer from single points of failure, limited scalability, and latency, making them inadequate for real-time and large-scale IoT deployments. As the complexity of IoT ecosystems grows, these limitations become increasingly critical, necessitating innovative and decentralized approaches [6].

Blockchain technology has emerged as a powerful solution for addressing IoT security concerns. Its decentralized architecture eliminates single points of failure, while its inherent properties of transparency, immutability, and distributed consensus provide enhanced security and resilience [7,8]. Despite its potential, IoT devices are often constrained by limited computational power, energy, and bandwidth, making the direct application of blockchain-based solutions complex. Furthermore, secure key management, efficient data storage, and real-time data sharing remain significant hurdles in realizing scalable and effective blockchain-based IoT systems [9–11].

In this paper, we propose a decentralized authentication protocol tailored to the unique requirements of IoT environments. Our approach integrates key technologies, including the Ethereum blockchain, the interplanetary file system (IPFS), elliptic curve cryptography (ECC), and the ASCON lightweight encryption algorithm. This protocol achieves mutual authentication between IoT devices, ensures that sensitive data are accessible only to authorized entities, and provides traceability for all communications. It is designed to address the resource limitations of IoT devices while maintaining high security standards.

The key contributions of this approach are as follows:

- **Decentralized Authentication Framework:** The proposed protocol employs asymmetric cryptography and the Diffie–Hellman key exchange to establish a shared secret, enabling secure and verifiable mutual authentication between devices.
- **Advanced Data Security:** By leveraging ASCON encryption, the protocol ensures the confidentiality and integrity of IoT data during transmission and storage.
- **Decentralized Data Management:** The integration of IPFS allows for secure, decentralized data storage, reducing dependency on centralized servers and improving data availability.
- **Traceability through Blockchain:** Encrypted content identifiers (CIDs) are stored on the Ethereum blockchain, providing a transparent and immutable audit trail for data exchanges.
- **Scalability and Efficiency:** The protocol is optimized for resource-constrained IoT environments, ensuring minimal computational and communication overhead while supporting large-scale deployments.

The rest of this paper is structured as follows: Section 2 reviews related works. Section 3 presents the background necessary to understand the proposed approach, including foundational concepts and technologies. Section 4 provides a detailed description of the proposed protocol's architecture and design. Section 5 evaluates the protocol's performance and security, highlighting its effectiveness and addressing potential challenges. Finally, Section 6 concludes this paper and outlines potential directions for future research.

2. Related Work

This section discusses the recent state-of-the-art articles under three categories, namely decentralized storage solutions, hybrid authentication mechanisms, and lightweight encryption in the IoT.

2.1. Decentralized Storage Solutions

Research in the field of decentralized storage solutions explores mechanisms to overcome the limitations of traditional centralized storage systems, such as vulnerability to censorship and risks of data breaches. Recent studies have examined various approaches and technologies aimed at enhancing the security, efficiency, and resilience of blockchain-based storage solutions.

Article [12] provides an overview of centralized and decentralized storage methods, focusing on technologies such as IPFS, Swarm, and Sia. A comparative analysis revealed

that each solution had specific strengths and weaknesses depending on the requirements of various blockchain applications. The authors emphasize that decentralized systems offer greater fault tolerance, enhanced privacy, and improved scalability compared to centralized solutions.

In [13] the author focuses on Filecoin, a blockchain-based storage system built on IPFS. It analyzes the unintended centralization introduced by its reward mechanism, which favors large storage miners capable of mobilizing significant resources. Although Filecoin aims for full decentralization, the results show that large enterprises with excess storage capacities dominate block creation, limiting the competitiveness of smaller nodes.

The author in [14] provides a technical overview of next-generation peer-to-peer data networks such as IPFS, Swarm, and Arweave. IPFS is highlighted as a groundbreaking solution due to its content-based addressing and deduplication capabilities. These features enable secure and efficient data sharing while facilitating integration with blockchains for various applications, including IoT and content distribution.

2.2. Hybrid Authentication Mechanisms

Hybrid authentication mechanisms are garnering increasing interest due to their ability to combine the strengths of multiple technologies to address modern security challenges. By integrating centralized systems with decentralized approaches such as blockchain, these mechanisms achieve a balance between scalability, efficiency, and robust security. Recent advancements in this field highlight the use of hybrid architectures and innovative techniques such as machine learning and lightweight cryptography to enhance authentication in various environments, including IoT systems and distributed networks.

The article in [15] explores a novel authentication method leveraging the Ethereum blockchain to replace traditional one-time passwords (OTPs) in two-factor authentication (2FA). Using smart contracts, it demonstrates how blockchain-based 2FA strengthens security against common vulnerabilities such as phishing and brute-force attacks. This method's applicability is illustrated through scenarios including registration, login, and password recovery, showcasing its potential to counter modern cyber threats.

The research in [16] presents a hybrid architecture integrating centralized edge servers with blockchain technology to achieve efficient and scalable IoT system authentication. The architecture employs lightweight cryptographic methods to reduce computational load and energy consumption, making it suitable for resource-constrained IoT devices. Performance evaluations demonstrate its superiority in scalability, response time, and energy efficiency compared to existing solutions.

In [17], the authors examine the integration of hybrid consensus algorithms with machine learning techniques to enhance blockchain security. By combining methods such as Delegated Proof-of-Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), and Proof-of-Work (PoW), they address challenges like scalability, energy efficiency, and resistance to 51% attacks. Their study highlights how machine learning improves anomaly detection, threat prevention, and adaptive decision-making, ensuring a robust and scalable blockchain ecosystem.

2.3. Lightweight Encryption in the IoT

IoT devices, characterized by limited energy, memory, and computational power, require lightweight encryption algorithms that balance efficiency and security. Numerous studies have explored these algorithms, including ChaCha20 and AES, and their application in IoT environments.

The article in [18] presents an optimized implementation of ChaCha20-Poly1305, tailored for high-speed embedded IoT systems. The authors demonstrate that this algorithm

provides robust security while consuming less memory and CPU, making it ideal for resource-constrained devices.

The study in [19] explores the integration of various algorithms, including ChaCha20, into the MAVLink protocol used for communication between drones and control stations. The authors show that ChaCha20 offers superior performance in terms of memory and CPU consumption while ensuring message confidentiality and integrity, making it suitable for low-power IoT systems.

In [20] the author compares three lightweight algorithms, AES-128, SPECK, and ASCON, on resource-limited IoT platforms such as Arduino Nano and Micro. SPECK stands out for its efficiency and simplicity, making it ideal for highly constrained devices. ASCON provides robust security with a trade-off in memory consumption. This study aids in selecting algorithms that balance performance and security for the IoT.

According to Table 1, summarizing this Section, the proposed protocol uniquely integrates decentralized storage, hybrid authentication, and lightweight encryption, addressing gaps in previous research. By combining IPFS, blockchain, and Ascon, it offers a scalable and secure solution tailored for IoT environments.

Table 1. Summary of all articles in the Related Work section.

Article	Decentralized Storage	Hybrid Authentication Mechanisms	Lightweight Encryption for the IoT	Description
[12]	✓	×	×	Focuses on decentralized storage solutions but lacks authentication and encryption aspects.
[13]	✓	×	×	Highlights blockchain-based storage with some centralization issues but no mention of encryption or authentication.
[14]	✓	×	×	IPFS is emphasized for scalability and content-based addressing, but no hybrid mechanisms or encryption is discussed.
[15]	×	✓	×	Proposes blockchain-based 2FA for securing traditional systems but lacks storage or encryption components.
[16]	×	✓	✓	Combines IoT authentication with lightweight cryptography for better efficiency and energy use.
[17]	×	✓	×	Uses hybrid consensus algorithms and AI for blockchain security but does not address storage or lightweight encryption.
[18]	×	×	✓	Optimizes implementation of ChaCha20 for IoT embedded systems with high efficiency but no storage or authentication focus.
[19]	×	×	✓	Is integrated for secure drone communication in IoT but lacks storage or authentication mechanisms.
[20]	×	×	✓	Comparative analysis for selecting encryption algorithms suitable for resource-constrained IoT devices.

3. Background

In this section, we provide a detailed overview of the core technologies utilized in the proposed protocol. These technologies are fundamental to ensuring security, efficiency, and decentralization.

3.1. Blockchain

Blockchain is a decentralized, distributed ledger technology designed to securely record transactions without the need for a central authority [21,22]. Unlike traditional, centralized databases, blockchain operates on a peer-to-peer network where each participant maintains a copy of the ledger. Transactions are verified through cryptographic mechanisms and consensus protocols, ensuring integrity and transparency [23].

Structure of Blockchain:

As shown in Figure 1, each block in the blockchain consists of several key components [24]:

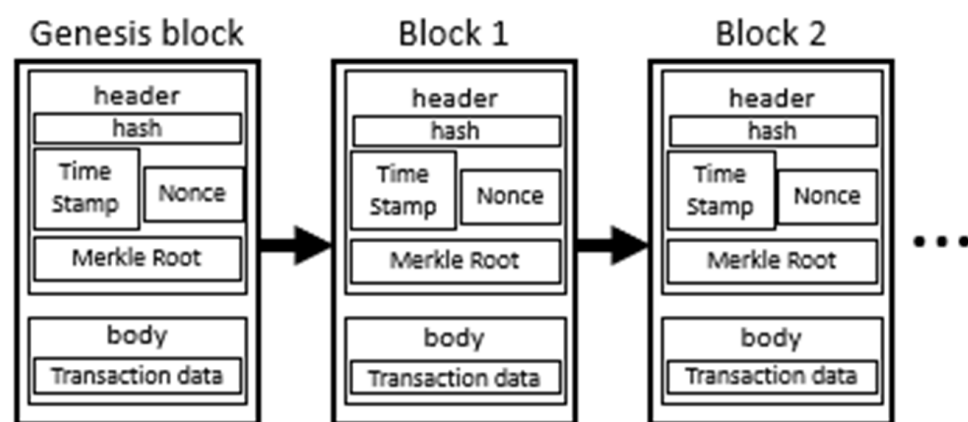


Figure 1. Blockchain structure.

- **Block Header:** This contains metadata, including a timestamp, the hash of the previous block, the Merkle root of transactions, and, in Proof-of-Work systems, a difficulty target. These elements ensure that each block is uniquely identified and linked to the previous one.
- **Block Body:** This stores a list of validated transactions. The transactions are structured in a Merkle tree, which allows for the efficient verification of the integrity and consistency of transaction records.
- **Nonce:** This is a randomly generated number used in mining to adjust the difficulty level, ensuring that block creation remains computationally feasible yet resistant to attacks.
- **Genesis Block:** This is the first block in the blockchain, which initializes the chain and serves as the foundation upon which subsequent blocks are built. Each new block references the hash of its predecessor, ensuring data integrity and immutability.

Choice of Blockchain:

The choice of Ethereum for this decentralized authentication protocol is based on several key technical factors that are crucial for ensuring the security, transparency, and robustness of the system. Although blockchains like IoTA offer superior performance in terms of transaction speed and energy efficiency, Ethereum has specific advantages for our application.

Security and Maturity: Ethereum is a well-established blockchain, widely used for smart contracts and the development of decentralized applications. It offers a high level of

security through its Proof-of-Work (PoW) consensus mechanism, and with the transition to Proof-of-Stake (PoS) in Ethereum 2.0, improvements in transaction speed and energy consumption are expected. This makes it suitable for an environment that requires strong guarantees regarding data integrity.

Support for Smart Contracts: Ethereum allows the development of powerful and flexible smart contracts, which are essential for securely automating the authentication of IoT devices. These contracts enable the enforcement of precise rules, ensure transaction integrity, and securely manage key generation.

Scalability Forecast: Although Ethereum currently has relatively slow transaction times and higher energy costs compared to some other blockchains, future developments (such as Ethereum 2.0 and sharding) are expected to significantly improve its scalability and performance. These advancements will better meet the needs of large-scale IoT environments.

However, it is important to note that alternatives like IoTA, which does not rely on a traditional blockchain, offer faster transaction times and reduced energy costs, making them attractive for IoT applications where responsiveness and energy efficiency are paramount. IoTA, for instance, uses a Tangle to validate transactions more quickly and efficiently than traditional blockchain mechanisms. Nevertheless, Ethereum remains the optimal choice for this project due to its strong security, broad adoption, and mature development ecosystem.

Security Mechanisms in Blockchain

Blockchain security is reinforced through several mechanisms:

- **Public Key Infrastructure (PKI):** Blockchain employs asymmetric cryptography [25], where transactions are signed using a private key and verified using a corresponding public key, ensuring authenticity and non-repudiation.
- **Decentralization and Distributed Ledger:** as shown in Figure 2 the decentralized nature of blockchain eliminates single points of failure by distributing data across multiple nodes [26]. This ensures that no single entity can alter the ledger.

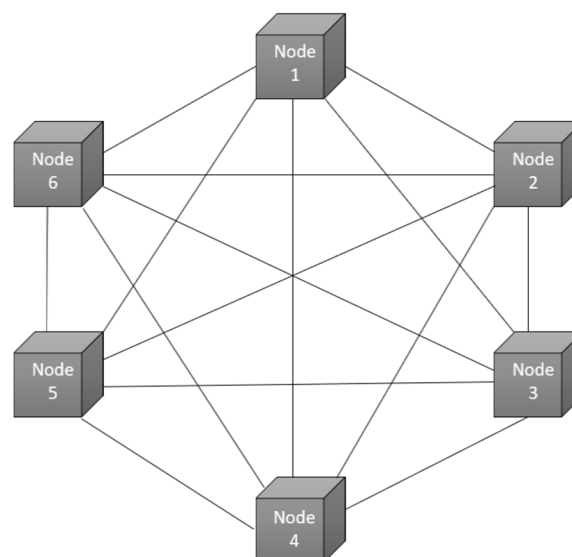


Figure 2. Decentralized network.

- **Consensus Mechanisms:**
 - **Proof-of-Work (PoW):** Requires computational effort to validate transactions, as seen in Bitcoin, ensuring that adding new blocks is costly and difficult to manipulate.

- Proof-of-Stake (PoS): Selects validators based on their stake in the network, reducing energy consumption and incentivizing honest participation.
- Delegated Proof-of-Stake (DPoS): Introduces a voting mechanism to elect trusted validators, enhancing scalability and transaction speed.
- Transparency and Immutability: Transactions recorded on the blockchain are visible to authorized participants and cannot be altered once recorded, preventing fraud and enhancing trust [27].

Applications of Blockchain

Blockchain technology finds applications in various domains:

- Decentralized Identity Management: Blockchain enables the efficient management of digital identities without relying on centralized authorities. In a decentralized or “self-sovereign” identity framework, individuals retain full control over their personal data, thereby minimizing the risks of identity theft and unauthorized access. Recent research has explored the advantages and challenges of blockchain-based identity systems [28].
- Supply Chain Tracking: In supply chain management, blockchain’s transparency and immutability offer an effective means to track products from origin to destination. Every transaction or transfer is permanently recorded, which enhances traceability, helps prevent fraud or counterfeiting, and improves overall operational efficiency [29].
- Smart Contracts: Smart contracts are self-executing programs that automatically enforce the terms of an agreement once predefined conditions are met. By removing the need for intermediaries, they lower transaction costs and reduce the chances of human error or fraud [30].

3.2. IPFS (Interplanetary File System)

The interplanetary file system (IPFS) is a decentralized, peer-to-peer file system that enhances the security and efficiency of data storage and sharing. Unlike traditional web protocols that rely on location-based addressing, such as URLs, IPFS employs content-based addressing. This approach ensures data authenticity and availability, as files are identified by a unique cryptographic hash derived from their content.

How IPFS Works

IPFS organizes data using a Merkle Directed Acyclic Graph (DAG), as shown in Figure 3, a sophisticated data structure where each block of data is identified and linked through a unique cryptographic hash. In this system, every data block references other blocks by their hash values, creating an interconnected web of content-addressed nodes [31, 32]. This design inherently enhances data integrity—any modification to a data block will result in a completely different hash, thereby immediately signaling a change or potential tampering.

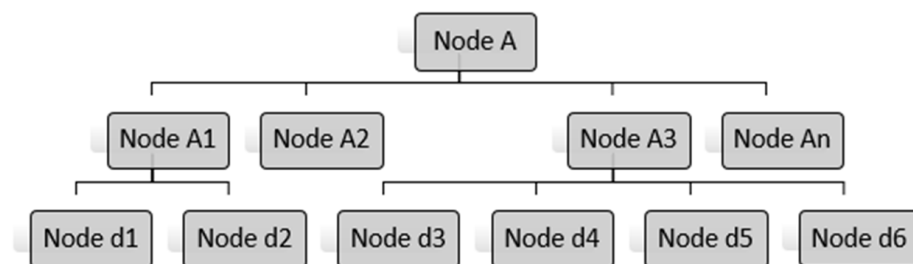


Figure 3. Merkle DAG.

Beyond ensuring data integrity, the Merkle DAG enables efficient versioning and data deduplication. Much like how Git manages version control, IPFS can represent changes to files over time by linking different versions within the DAG. This not only allows users to retrieve historical versions of files but also minimizes storage redundancy when multiple files share common data blocks.

3.3. ASCON

The ASCON algorithm is a lightweight cryptographic algorithm designed for constrained environments such as those of IoT devices, blockchain networks, and decentralized applications. Originally developed for the CAESAR competition, ASCON was later standardized by NIST as the lightweight cryptography standard [33,34]. It is designed to provide strong security while maintaining minimal computational overhead.

Features of ASCON

ASCON employs an Authenticated Encryption with Associated Data (AEAD) scheme [35], which ensures both confidentiality and integrity in a single step. It utilizes a sponge construction, a permutation-based processing technique that enhances efficiency in encryption and hashing operations. ASCON is optimized for fast execution with low resource consumption, making it suitable for environments with limited processing power. Additionally, it is resistant to side-channel attacks [36], such as timing- and power-based attacks, which commonly target cryptographic implementations.

Security Mechanisms of ASCON

ASCON integrates encryption and authentication into one streamlined operation, reducing the computational burden compared to traditional AES-based schemes that require separate authentication mechanisms [37]. It is designed with nonce misuse resistance, preventing security vulnerabilities that arise from nonce repetition. Its low memory footprint ensures efficient performance on resource-constrained devices, such as embedded systems and decentralized applications [34,38].

By integrating blockchain, IPFS, and ASCON, the proposed protocol ensures robust security, decentralization, and high efficiency. Blockchain provides an immutable and transparent record-keeping system, IPFS offers distributed data storage with verifiable integrity, and ASCON guarantees lightweight cryptographic protection. This combination makes the protocol particularly well suited for decentralized authentication and secure data management in resource-constrained environments.

4. Proposed Work

The main objective of this paper is to propose a decentralized blockchain-based authentication and IPFS-based data management protocol for smart systems that is suitable for resource-constrained smart devices. Thus, the time consumed for message transmission and the issue of network latency are not discussed in this study as they depend on the blockchain network, specifically the consensus protocol.

The proposed protocol leverages the following emerging technologies: the Ethereum blockchain and smart contracts for secure data management; IPFS (interplanetary file system) for decentralized storage; ECC (elliptic curve cryptography) for lightweight key management; SAS (Secure Authentication System) for robust mutual authentication; and an ASCON cipher for advanced IoT data encryption. This combination ensures the confidentiality, integrity, and availability of critical IoT application data, providing an efficient and robust infrastructure. Figure 1 illustrates the overall architecture of the proposed protocol, showcasing its essential elements.

The secure IoT protocol leverages blockchain and IPFS technologies, and consists of two phases: mutual authentication and secure data sharing. Each device possesses the following:

- Personal account;
- Corresponding device account;
- Shared ID.

Phase 1: Mutual Authentication

- Key Generation: IoT devices (IoT1 and IoT2) generate asymmetric key pairs (public and private).
- Public Key Registration: Public keys are stored on the blockchain for authentication.
- Shared Key Calculation: Devices calculate a shared key using the Diffie–Hellman protocol.
- Secure Authentication System (SAS) Generation: Devices generate a SAS by combining their shared key and shared ID.
- SAS Storage: The SAS is stored on the blockchain.
- SAS Comparison: Devices compare SASs for mutual authentication validation.

Phase 2: Secure Data Sharing

- Data Storage: IoT1 stores data (images, etc.) in IPFS.
- Content ID Retrieval: IoT1 retrieves data CID.
- CID Encryption: CID is encrypted using ASCON algorithm and shared key.
- Encrypted CID Storage: Encrypted CID is stored on blockchain.
- Encrypted CID Retrieval: IoT2 retrieves encrypted CID.
- CID Decryption: IoT2 decrypts CID using shared key.
- Data Retrieval: IoT2 retrieves data from IPFS using decrypted CID.

Tools Used

- Blockchain (authentication, key storage, and CID encrypted storage);
- IPFS (data storage);
- Asymmetric encryption algorithms (key generation);
- ASCON algorithm (CID encryption and decryption);
- Diffie–Hellman protocol (shared key generation).

4.1. Smart Contract

The smart contract used in this protocol (Algorithm 1) is a set of functions facilitating key exchange and encrypted data storage between IoT devices. Deployed on the blockchain, it utilizes the Solidity language. It enables device registration, public and shared key storage, and the management of encrypted data (CID). Its key functions include the following: device registration (`registerDevice`), public key retrieval (`getPublicKey`), shared key definition (`setSaS`), and encrypted data storage (`setEncryptedData`) and retrieval (`getEncryptedData`). These operations ensure secure, transparent, and confidential data exchanges between IoT devices.

Algorithm 1: Smart contract

```

contract IoTmanagement {
// Contract owner's address
address public owner;
// Mapping of device addresses to their public keys
mapping(address => string) private publicKeys;
// Mapping of device addresses to their shared keys
mapping(address => string) private sharedKeys;
// Structure to store encrypted CID, nonce, and authTag
struct EncryptedData {
string encryptedCID;
string nonce;
string authTag;}
// Mapping of device addresses to their encrypted data
mapping(address => EncryptedData) private encryptedDeviceData;
// Event emitted when a device is registered
event DeviceRegistered(address indexed device, string publicKey);
// Event emitted when a shared key is set
event SharedKeySet(address indexed device, string sharedKey);
// Event emitted when encrypted data are set
event EncryptedDataSet(address indexed device, string encryptedCID);
// Constructor: sets the contract owner
constructor() {
owner = msg.sender; // Set the deployer as the owner}
// Function to register a device and its public key
function registerDevice(address _deviceAddress, string memory _publicKey) public {
publicKeys[_deviceAddress] = _publicKey; // Store the public key
emit DeviceRegistered(_deviceAddress, _publicKey); // Emit registration event}
// Function to retrieve a device's public key
function getPublicKey(address _deviceAddress) public view returns (string memory) {
return publicKeys[_deviceAddress]; // Return the public key}
// Function to set a shared key for a device
function setSaS(address _deviceAddress, string memory _sharedKey) public {
sharedKeys[_deviceAddress] = _sharedKey; // Store the shared key
emit SharedKeySet(_deviceAddress, _sharedKey); // Emit shared key event}
// Function to retrieve a device's shared key
function getSaS(address _deviceAddress) public view returns (string memory) {
return sharedKeys[_deviceAddress]; // Return the shared key}
// Function to store encrypted data (CID) for a device
function setEncryptedData(address _deviceAddress, string memory _encryptedCID,
string memory _nonce, string memory _authTag) public {
// Only the owner or the device itself can call this function
encryptedDeviceData[_deviceAddress] = EncryptedData(_encryptedCID, _nonce,
_authTag); // Store encrypted data
emit EncryptedDataSet(_deviceAddress, _encryptedCID, _nonce, _authTag); // Emit
encrypted data event}
// Function to retrieve encrypted data for a device
function getEncryptedData(address _deviceAddress) public view returns
(EncryptedData memory) {
return encryptedDeviceData[_deviceAddress]; // Return encrypted data}}

```

4.2. Mutual Authentication

In Figure 4 the Phase 1 mutual authentication process in the secure IoT communication protocol involves several key steps. First, each IoT device generates an asymmetric key pair consisting of a public key and a private key. The public keys are then registered on the blockchain for authentication. Each device retrieves the other device's public key.

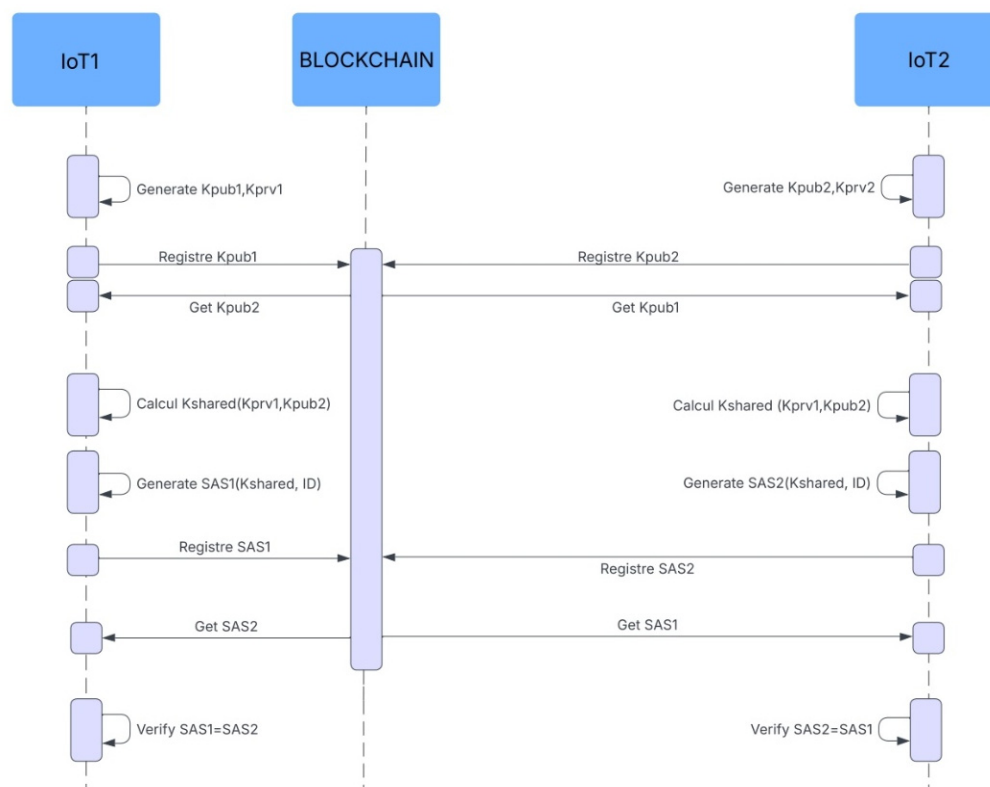


Figure 4. Sequence diagram of the mutual authentication phase.

Using its private key and the other device's public key, each device calculates a shared key via the Diffie–Hellman protocol. This shared key is combined with the common ID to generate a Secure Authentication System (SAS). Each device stores its SAS on the blockchain and then retrieves and compares the other device's SAS. If the SAS matches ($SAS1 = SAS2$), communication is authorized, ensuring secure mutual authentication between the IoT devices. This process guarantees the confidentiality, integrity, and security of the exchanged data.

4.3. Secure Data Sharing

As shown in Figure 5 the Phase 2 secure data sharing protocol in IoT communication involves multiple steps. Initially, IoT device 1 stores data (images, files, etc.) in the decentralized interplanetary file system (IPFS) and retrieves a unique Content ID (CID). The CID is then encrypted using the ASCON algorithm with the shared key from Phase 1 to ensure confidentiality. The encrypted CID is stored on the blockchain, enabling authentication and transaction traceability. IoT device 2 retrieves the encrypted CID, decrypts it using ASCON, and uses the decrypted CID to retrieve data from IPFS. This process ensures secure, confidential, and traceable data sharing between IoT devices, guaranteeing data integrity and security.

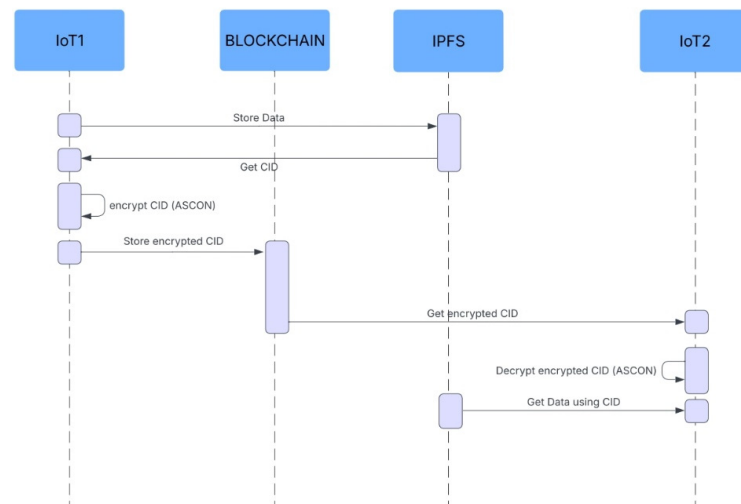


Figure 5. Sequence diagram of the secure data sharing phase.

4.4. Global Architecture

For a comprehensive understanding of the secure IoT communication protocol, the Figure 6 presents a complete scenario illustrating the key interaction steps. This schema details the secure communication process, from mutual authentication to confidential data sharing, between two IoT devices, including encryption and transaction traceability.

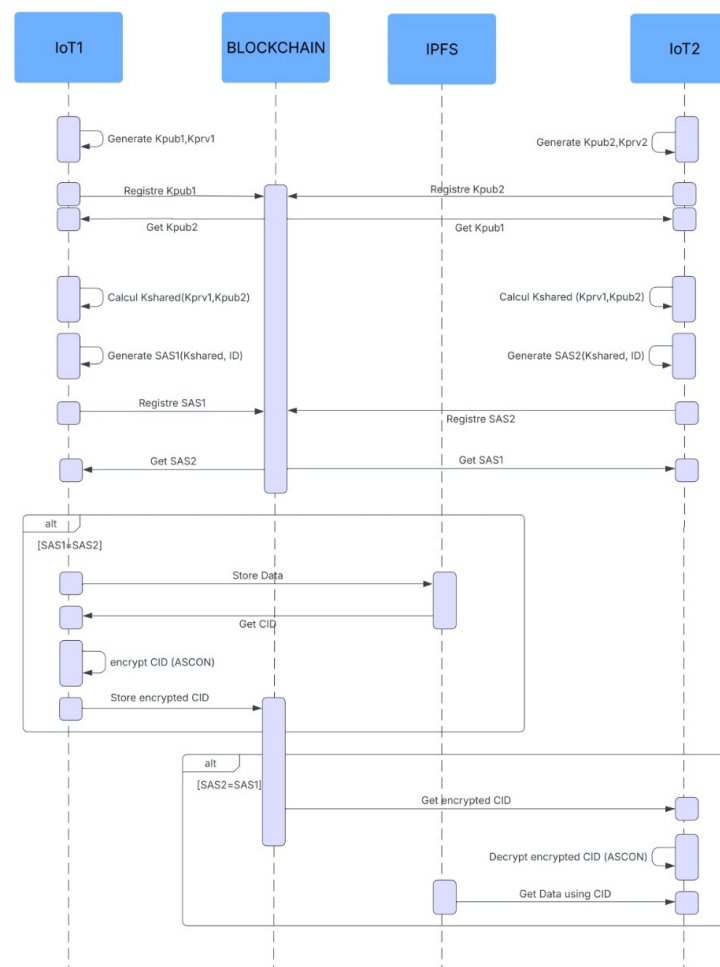


Figure 6. Sequence diagram of the global architecture.

4.5. Lightweight Cryptography: ASCON

In the proposed protocol, the lightweight cryptography algorithm ASCON-128 plays a central role in securing the content IDs (CIDs) generated by IPFS. These CIDs, which are unique identifiers used to locate data on the IPFS network, are encrypted using ASCON-128 before being stored on the blockchain.

Encrypting the CIDs with ASCON-128 ensures that only authorized parties can access these identifiers. Indeed, the encrypted CIDs are stored on the blockchain, and their decryption can only be performed with the appropriate shared key. This guarantees restricted access to the data, even though the blockchain is public.

The CIDs generated by IPFS have a standardized size of approximately 34 bytes (or 46 characters in base58), which is a significant advantage for the use of ASCON-128, an algorithm designed for efficient encryption management. During the encryption of the CIDs, the process unfolds as follows:

1. **Padding the data:** The 34-byte CID is padded with 2 additional bytes, bringing the total size to 36 bytes. This size is a multiple of 16 bytes, meeting the requirements of ASCON-128, which encrypts blocks of 128 bits (or 16 bytes).
2. **Encryption with ASCON-128:** This 36-byte block is then encrypted in a single operation using ASCON-128. The result of the encryption is also a 36-byte block, containing the encrypted data as well as an authentication tag. This tag ensures the integrity of the information and detects any attempts to modify the CIDs.

Figure 7 illustrates the process, detailing each step from padding a CID to encrypting it with ASCON-128 and producing the final encrypted output.

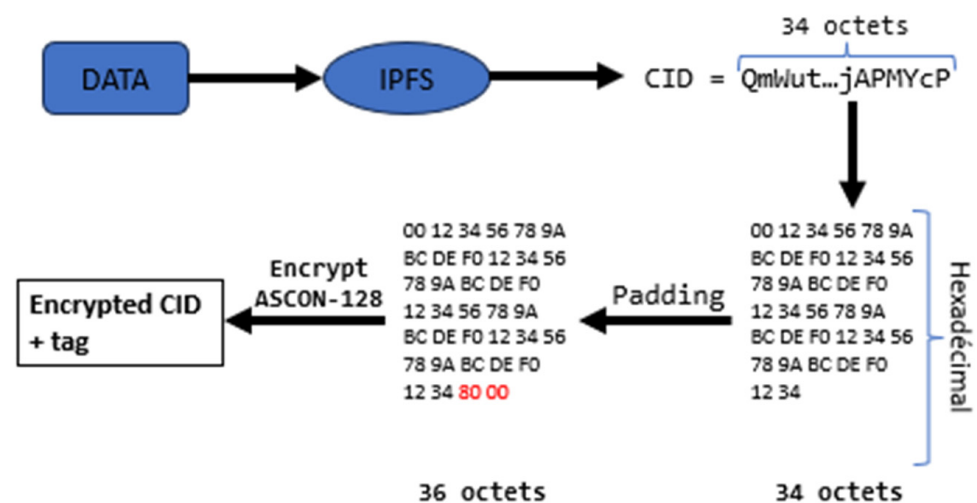


Figure 7. CID padding in encryption phase with ASCON.

As shown in Table 2, the data inserted into the system can take different forms, such as images, files, or text. Regardless of their nature, they are first stored on IPFS, which generates a unique 34-byte CID (content identifier). Before encryption with ASCON-128, this CID is converted to a hexadecimal format and then padded to reach the required size of 36 bytes. The table below illustrates this process for different types of data.

Table 2. CID conversion and padding for ASCON-128 encryption by data type.

Data Type	CID (34 Bytes)	Hexadecimal (34 Bytes)	Padding (36 Bytes)
Image	QmA1bD4...eW3xy7	5A 67 89 10 4C 7F 3B E1 92 D5 46 77 3B 5C 93 2F A1 D2 0F 57 3D C8 71 89 2F D0 5A 12 61 74 90 8D 56 34 2A	5A 67 89 10 4C 7F 3B E1 92 D5 46 77 3B 5C 93 2F A1 D2 0F 57 3D C8 71 89 2F D0 5A 12 61 74 90 8D 56 34 2A 80 00
File	QmX7cM2V...kT4pQ1	12 39 47 A8 C3 9F 7B D0 51 67 84 1A 95 C8 3F 9A 12 57 76 6D 43 B0 81 45 9E C7 54 31 F0 D4 C3 29 D8 7E	12 39 47 A8 C3 9F 7B D0 51 67 84 1A 95 C8 3F 9A 12 57 76 6D 43 B0 81 45 9E C7 54 31 F0 D4 C3 29 D8 7E 80 00
Text	QmK7hL3A...rV9zQ2	8B 15 49 33 5D F2 98 8E 44 77 5F 26 1D 38 92 A9 5C 10 6E 3D A1 D8 44 F7 33 18 74 2B A1 91 D2 D5 7C 8A	8B 15 49 33 5D F2 98 8E 44 77 5F 26 1D 38 92 A9 5C 10 6E 3D A1 D8 44 F7 33 18 74 2B A1 91 D2 D5 7C 8A 80 00

5. Implementation

The main objective of this section is to describe the implementation of the “decentralized blockchain-based authentication and IPFS-based data management protocol using Ascon”. The implementation was carried out in a simulated environment to evaluate the protocol’s efficiency in terms of resource consumption and performance.

5.1. Experimental Environment

The experiment was conducted by setting up a simulated network integrating blockchain, IPFS, and secure authentication and data management mechanisms. The hardware and software components used are listed below (Table 3):

- Two laptops (blockchain and IPFS nodes);
- Two machines simulating IoT devices;
- A local server hosting the main IPFS node.

Table 3. Nodes used in the experimentation.

Category	Components	Specifications
Hardware	Two computers (blockchain and IPFS nodes)	Model: DELL CPU: Intel i5 or equivalent RAM: 8 GB Storage: 256 GB SSD OS: Windows 10 or Linux (Ubuntu recommended)
	Two machines simulating IoT devices	Model: Raspberry Pi 4 or equivalent CPU: Quad-core ARM Cortex-A72 RAM: 2 GB or 4 GB OS: Raspbian or any Linux-based OS
	A local server hosting the main IPFS node	Model: HP CPU: Intel i3 or equivalent RAM: 4 GB Storage: 500 GB HDD or SSD OS: Ubuntu 20.04 LTS or equivalent
Software and Tools	Remix IDE, Ganache, Web3.js	Ensure the software is up to date
	IPFS Desktop	0.42.0
	Node-RED	4.0.9

Software and tools

- Remix IDE, Ganache, Web3.js;
- IPFS Desktop;
- Node-RED.

The hardware infrastructure consisted of two laptops acting as blockchain and IPFS nodes, allowing local testing of the blockchain and decentralized storage. Additionally, two machines were used to simulate IoT devices with limited computational resources, ensuring realistic constraints similar to real-world IoT environments. These devices were configured with a restricted CPU and RAM to better reflect the processing limitations of embedded systems. A local server was deployed to host the main IPFS node, facilitating file storage and retrieval. However, for a more extensive simulation, a distributed IPFS network with multiple nodes would be required.

The software environment included Remix IDE, Ganache, and Web3.js for smart contract development and blockchain interaction. IPFS Desktop was utilized for decentralized data storage, enabling content-based addressing and file sharing. Node-RED was employed to orchestrate IoT data flow and automate interactions between different components of the system.

5.2. Estimation of Resource Consumption

There are three important factors to estimate during the experiment: The time required by a smart device to perform each function, the RAM required for each function, and the CPU consumption.

A- Time consumption

The first important factor is the time required for a smart device or user to perform the smart contract functions mentioned above. Table 3 shows the maximum, and average time required over the 100 runs of the experiment. We chose 100 repetitions to ensure the robustness and statistical significance of the results. This number, recommended in the literature, helps reduce the impact of random variations and anomalies while providing an accurate estimate of performance under varied conditions. By repeating the experiment 100 times, we were able to obtain an average of the measurements, minimizing biases and occasional anomalies while balancing precision and the available resources for the experiment.

As shown in Table 4, the key pair generation took a maximum time of 0.015 s and an average of 0.001 s. Similarly, SAS verification by an IoT node required a maximum time of 0.07 s and an average of 0.03 s. These relatively low results demonstrate that the protocol is suitable for IoT environments, as it guarantees fast authentication with minimal impact on device performance, thus promoting reduced energy consumption and better system scalability.

Table 4. Summary Table of function time.

Function	Max Time	Average Time
KeypairGeneration	0.015	0.001
VerifieSAS	0.07	0.03
CIDEncryption	0.01	0.003
AddCID	0.11	0.04
GetCID	0.06	0.02
AddData	0.12	0.05
GetData	0.06	0.02

B- CPU consumption

The second important factor to discuss is the energy consumption required of a network node to perform the functions mentioned above. Figure 8 shows a comparison between the average power consumed to perform each function, in which the power consumed by the phase of creating a certificate is the maximum value, which is 25%.

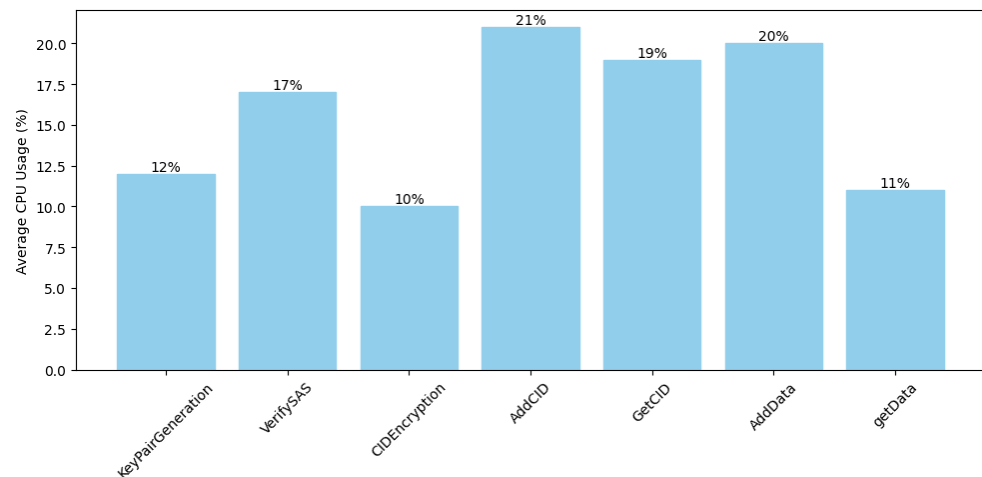


Figure 8. CPU consumption.

C- RAM consumption

The third factor is RAM consumption, which is also an important criterion. Figure 9 shows the results obtained over the 100 execution times, and demonstrates the low memory consumption of our approach. This low resource usage makes our solution particularly suitable for IoT environments, where memory and energy optimization are essential.

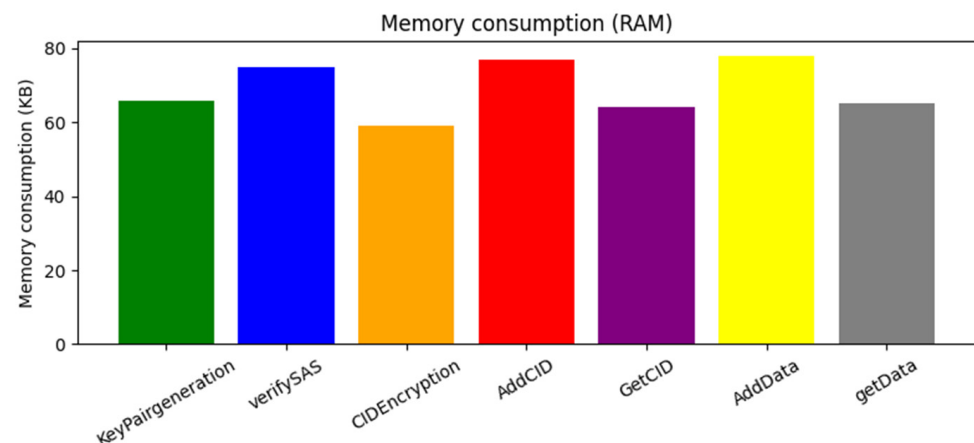


Figure 9. RAM consumption.

6. Evaluation and Security

This Section describes in detail the security mechanisms integrated into the proposed protocol and outlines the strategies employed to mitigate potential threats. By combining authenticated encryption, blockchain immutability, and the distributed nature of IPFS, this protocol is designed to protect sensitive data and ensure the integrity, availability, and proper access control of stored resources.

6.1. Security Analysis

The protocol achieves robust security primarily through the use of authenticated encryption with the Ascon algorithm. This method guarantees data confidentiality by ensuring that all sensitive information is encrypted before transmission. Even if an adversary manages to intercept the communications, the encrypted data remain unintelligible without the corresponding decryption key. This approach is crucial for preventing unauthorized access, particularly in environments where data are exchanged over open networks.

Data integrity is secured by leveraging two key mechanisms. First, the blockchain's inherent immutability ensures that all metadata—such as transaction records and access logs—cannot be altered once recorded. This provides a reliable audit trail and prevents any unauthorized modifications. Second, IPFS employs content-based addressing, where every file is identified by its cryptographic hash. Any attempt to alter a file results in a different hash value, thereby making tampering immediately detectable.

Access control is enforced using smart contracts deployed on the blockchain. These contracts are programmed to execute strict permission rules, ensuring that only authorized users can access or modify resources stored in the system. The combination of these measures creates a robust barrier against unauthorized access while maintaining a transparent and verifiable record of all interactions.

Furthermore, the decentralized architecture of both blockchain and IPFS contributes significantly to the protocol's resilience. By eliminating single points of failure, the system remains operational even if parts of the network are compromised or subjected to distributed denial-of-service (DDoS) attacks. This distributed design not only improves reliability but also enhances the overall security posture of the protocol.

6.2. Threat Mitigation

The proposed protocol incorporates a variety of targeted mechanisms to neutralize potential threats and ensure the security of the system. One of the primary concerns is the risk of replay attacks, where an attacker might capture and resend valid messages to illicitly repeat transactions or commands. To counter this, each authentication attempt within the protocol is tagged with a unique nonce and an accompanying session token. These elements ensure that every transaction is unique; even if an attacker intercepts a message, it cannot be reused because the system will recognize it as a duplicate and subsequently reject it.

Another significant threat is posed by Sybil attacks, where a malicious actor creates multiple fake identities to overwhelm or manipulate the network. The protocol defends against this by leveraging the inherent properties of blockchain consensus mechanisms. These mechanisms, such as Proof-of-Stake or other decentralized algorithms, make it computationally and economically prohibitive for an attacker to flood the network with fraudulent identities. As a result, the network remains resilient and resistant to domination by any single entity attempting to subvert the system.

Insider threats, which involve malicious actions by authorized users or compromised accounts, are addressed through robust access control measures. The protocol employs Role-Based Access Control (RBAC) to ensure that every user is granted only the permissions necessary for their role. This minimizes the potential damage that could be caused by an insider. Furthermore, regular security audits and the continuous monitoring of user activities help detect any anomalous behavior early on, allowing for prompt intervention to prevent potential security breaches.

Finally, to protect against data theft, the protocol utilizes the Ascon encryption algorithm to secure all sensitive data during transmission. This means that even if an attacker manages to intercept the communication channel, the captured data remain encrypted

and indecipherable without their corresponding decryption key. By ensuring that data are always encrypted while in transit, the protocol effectively safeguards sensitive information from being extracted or misused by unauthorized parties.

6.3. Protection Against Specific Attacks

The proposed protocol is designed to counter a variety of specific attack vectors commonly encountered in distributed environments. Each of these potential threats is addressed through tailored mechanisms that work in concert to secure communications and data.

Man-in-the-Middle (MitM) Attacks:

Man-in-the-Middle attacks involve an adversary intercepting communications between two parties with the aim of eavesdropping or altering the transmitted messages. In our protocol, robust cryptographic keys and advanced encryption mechanisms—specifically using the Ascon algorithm—serve as the first line of defense. By encrypting all communications end-to-end, the protocol ensures that any intercepted messages remain unreadable without the appropriate decryption key. Additionally, blockchain-based authentication guarantees that both parties are verified before any exchange occurs, making it exceedingly difficult for an attacker to successfully impersonate either party or inject fraudulent data into the communication stream.

Brute Force Attacks:

Brute force attacks attempt to decipher encrypted data by systematically testing every possible key until the correct one is found. The protocol mitigates this threat by relying on the inherent strength and high computational complexity of the Ascon algorithm. As Ascon is designed with a large key space and strong cryptographic primitives, it renders any brute force attempt computationally prohibitive, effectively ensuring that decryption without the correct key remains virtually impossible.

Phishing Attacks:

Phishing attacks typically target user credentials by tricking users into disclosing sensitive information. To counteract this threat, the protocol enforces strict authentication through smart contracts and the use of unique session tokens. This multi-factor approach ensures that even if an attacker manages to compromise a user's login details, the additional security layer provided by session tokens and the smart contract authentication checks will prevent unauthorized access to system resources.

Denial-of-Service (DoS) Attacks:

Denial-of-service attacks aim to overwhelm a system with excessive traffic, thereby rendering it unavailable to legitimate users. The decentralized architecture of our protocol, which leverages both blockchain and IPFS, inherently mitigates this risk. The distributed nature of the system eliminates single points of failure, ensuring that even if certain nodes or segments of the network are targeted and become temporarily inoperative, the overall system remains operational. This redundancy and distribution of workload significantly reduce the impact of DoS attacks.

Data Corruption:

Ensuring data integrity is critical in any secure system. Our protocol defends against data corruption by combining the immutability of the blockchain with IPFS's content-based addressing. In this system, any unauthorized modification of data results in a discrepancy between the stored cryptographic hash and the actual content. Such a mismatch is immediately detectable, ensuring that any attempts to corrupt the data are promptly identified and can be addressed before further harm is carried out.

Eavesdropping:

Eavesdropping involves the unauthorized interception of data during transmission. To guard against this, all communications within the protocol are encrypted using the strong, lightweight encryption provided by Ascon. This ensures that even if an attacker is able to tap into the communication channel, the intercepted data will remain encrypted and indecipherable without the correct decryption key, thereby preserving the confidentiality of the transmitted information.

6.4. Evaluation of Existing IoT Security Protocols

In order to better understand the advantages of our decentralized authentication protocol based on blockchain and IPFS, we compared several scientific articles in the field of IoT security. The following table presents a comparison of several relevant articles based on different security criteria, such as authenticated encryption, replay protection, smart contract access, content integrity (IPFS), DoS resilience, and lightweight for the IoT. This analysis highlights the strengths of our approach in comparison with existing solutions.

As shown in Table 5, our protocol offers several advantages over existing IoT security solutions. In particular, it stands out for its ability to provide authenticated encryption and effective replay protection, making it more resilient to malicious attacks. Furthermore, unlike many approaches, our solution integrates smart contracts to manage access to resources in a decentralized and transparent manner.

Table 5. Security feature comparison of IoT protocols.

Article Title	Authenticated Encryption	Replay Protection	Smart Contract Access	Content Integrity (IPFS)	DoS Resilience	Lightweight for IoT
[39]	✓	✓	✓	×	×	✓
[40]	✓	✓	×	×	✓	✓
[41]	✓	✓	✓	×	×	×
Our approach	✓	✓	✓	✓	✓	✓

Regarding content integrity, our use of IPFS ensures data protection while providing fast and reliable access, even in the event of distributed attacks. We have also considered DDoS resilience, a critical issue in IoT environments, by incorporating mechanisms that reduce the system's vulnerability to such attacks.

Finally, our solution is specifically designed to be lightweight, making it suitable for resource-constrained IoT devices, which are often limited in terms of hardware and energy resources. This feature allows our protocol to function efficiently in large-scale IoT environments while maintaining a high level of security.

7. Conclusions

The rapid growth of IoT devices has brought both opportunities and security challenges, particularly in authentication, data management, and communication. This paper proposes a decentralized authentication protocol using the Ethereum blockchain, IPFS, elliptic curve cryptography (ECC), and the ASCON lightweight encryption algorithm. This protocol addresses IoT security challenges by enabling mutual authentication, ensuring data confidentiality, and providing traceability via blockchain-integrated encrypted content identifiers (CIDs). By utilizing IPFS for decentralized storage and lightweight cryptography, this solution meets IoT device constraints while maintaining high security.

Our experimental analysis shows that the protocol achieves its objectives, making it scalable and suitable for real-time, secure IoT applications.

Future work will focus on improving the energy efficiency and scalability of blockchain implementations in the IoT, as well as exploring adaptive authentication methods and DNA-based cryptographic techniques to enhance system security and usability.

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

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study shall be made available upon request.

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

Abbreviations

The following abbreviations are used in this manuscript:

IOT	Internet of Things
IPFS	Interplanetary File System
ECC	Elliptic Curve Cryptography
ASCON	Authenticated Secure Construction

References

1. Mouha, R.A. Internet of Things (IoT). *J. Data Anal. Inf. Process.* **2021**, *9*, 77–101. [\[CrossRef\]](#)
2. Panahi Rizi, M.H.; Hoseini Seno, S.A. A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city. *Internet Things* **2022**, *20*, 100584. [\[CrossRef\]](#)
3. Steneck, N.H. Fostering integrity in research: Definitions, current knowledge, and future directions. *Sci. Eng. Ethics* **2006**, *12*, 53–74. [\[CrossRef\]](#)
4. Gabriel, T.; Cornel-Cristian, A.; Arhip-Calin, M.; Zamfirescu, A. Cloud storage: A comparison between centralized solutions versus decentralized cloud storage solutions using blockchain technology. In Proceedings of the 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, 3–6 September 2019; pp. 1–5. [\[CrossRef\]](#)
5. Wu, F.; Li, X.; Xu, L.; Kumari, S.; Karuppiah, M.; Shen, J. A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server. *Comput. Electr. Eng.* **2017**, *63*, 168–181. [\[CrossRef\]](#)
6. Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I. Internet of things (IoT) security: Current status, challenges and prospective measures. In Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 14–16 December 2015; pp. 336–341. [\[CrossRef\]](#)
7. Xu, L.; Lu, Y.; Li, L. Embedding blockchain technology into IoT for security: A survey. *IEEE Internet Things J.* **2021**, *8*, 10452–10473. [\[CrossRef\]](#)
8. Gugueoth, V.; Safavat, S.; Shetty, S.; Rawat, D. A review of IoT security and privacy using decentralized blockchain techniques. *Comput. Sci. Rev.* **2023**, *50*, 100585. [\[CrossRef\]](#)
9. Lao, L.; Li, Z.; Hou, S.; Xiao, B.; Guo, S.; Yang, Y. A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus, and Traffic Modeling. *ACM Comput. Surv.* **2020**, *53*, 18. [\[CrossRef\]](#)
10. Abdelmaboud, A.; Ahmed, A.I.A.; Abaker, M.; Eisa, T.A.E.; Albasheer, H.; Ghorashi, S.A.; Karim, F.K. Blockchain for IoT applications: Taxonomy, platforms, recent advances, challenges and future research directions. *Electronics* **2022**, *11*, 630. [\[CrossRef\]](#)
11. Rahimi, P.; Khan, N.D.; Chrysostomou, C.; Vassiliou, V.; Nazir, B. A secure communication for maritime IoT applications using blockchain technology. In Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, 25–27 May 2020; pp. 244–251. [\[CrossRef\]](#)
12. Badhe, G.; Arjunwadkar, M. Decentralised Storage Systems for Blockchain Powered Applications. *Int. J. Eng. Technol. Manag. Sci.* **2024**, *7*, 324–326.
13. Guidi, B.; Michienzi, A.; Ricci, L. Evaluating the Decentralisation of Filecoin. In Proceedings of the 3rd International Workshop on Distributed Infrastructure for the Common Good (DICG '22), Quebec City, QC, Canada, 7 November 2022; ACM: New York, NY, USA, 2022; pp. 1–6. [\[CrossRef\]](#)

14. Daniel, E.; Tschorsch, F. IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 1176–1205. [\[CrossRef\]](#)
15. McCabe, C.; Mohideen, A.I.C.; Singh, R. A Blockchain-Based Authentication Mechanism for Enhanced Security. *Sensors* **2024**, *24*, 5830. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Khashan, O.A.; Khafajah, N.M. Efficient Hybrid Centralized and Blockchain-Based Authentication Architecture for Heterogeneous IoT Systems. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 726–739. [\[CrossRef\]](#)
17. Venkatesan, K.; Rahayu, S.B. Blockchain Security Enhancement: An Approach Towards Hybrid Consensus Algorithms and Machine Learning Techniques. *Sci. Rep.* **2024**, *14*, 1149. [\[CrossRef\]](#)
18. De Santis, F.; Schauer, A.; Sigl, G. ChaCha20-Poly1305 Authenticated Encryption for High-Speed Embedded IoT Applications. In Proceedings of the 2017 Design, Automation and Test in Europe Conference (DATE), Lausanne, Switzerland, 27–31 March 2017; IEEE: New York, NY, USA, 2017; pp. 692–694. [\[CrossRef\]](#)
19. Allouch, A.; Cheikhrouhou, O.; Koubâa, A.; Khalgui, M.; Abbes, T. MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; IEEE: New York, NY, USA, 2019; pp. 621–628. [\[CrossRef\]](#)
20. Radhakrishnan, I.; Jadon, S.; Honnavalli, P.B. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors* **2024**, *24*, 4008. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Mistry, I.; Tanwar, S.; Tyagi, S.; Kumar, N. Blockchain for 5G-Enabled IoT for Industrial Automation: A Systematic Review, Solutions, and Challenges. *Mech. Syst. Signal Process.* **2019**, *135*, 106382. [\[CrossRef\]](#)
22. Kabra, N.; Bhattacharya, P.; Tanwar, S.; Tyagi, S. MudraChain: Blockchain-Based Framework for Automated Cheque Clearance in Financial Institutions. *Future Gener. Comput. Syst.* **2020**, *102*, 574–587. [\[CrossRef\]](#)
23. Bashir, I. *Mastering Blockchain: Deeper Insights into Decentralization, Cryptography, Bitcoin, and Popular Blockchain Frameworks*; Packt Publishing: Birmingham, UK, 2017.
24. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE 6th International Congress on Big Data, Honolulu, HI, USA, 25–30 June 2017. [\[CrossRef\]](#)
25. Crosby, M.; Nachiappan, P.; Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain Technology: Beyond Bitcoin. *Appl. Innov. Rev.* **2016**, *2*, 6–19. Available online: <https://scet.berkeley.edu/wp-content/uploads/AIR-2016-Blockchain.pdf> (accessed on 9 March 2025).
26. Yli-Huumo, J.; Ko, D.; Choi, S.; Park, S.; Smolander, K. Where Is Current Research on Blockchain Technology?—A Systematic Review. *PLoS ONE* **2016**, *11*, e0163477. [\[CrossRef\]](#)
27. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A Survey on the Security of Blockchain Systems. *Future Gener. Comput. Syst.* **2020**, *107*, 841–853. [\[CrossRef\]](#)
28. Ahmed, M.R.; Saeed, S.; Khan, M.K.; Han, J.; Han, K. Blockchain-Based Identity Management System and Self-Sovereign Identity Ecosystem: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 113436–113481. [\[CrossRef\]](#)
29. Sylvestre, C.E. The Role of Blockchain in Enhancing Supply Chain Transparency. *J. Bus. Logist.* **2021**, *42*, 71–87. [\[CrossRef\]](#)
30. Hewa, T.; Ylianttila, M.; Liyanage, M. Survey on blockchain based smart contracts: Applications, opportunities and challenges. *J. Netw. Comput. Appl.* **2021**, *177*, 102857. [\[CrossRef\]](#)
31. Jayabalan, J.; Jeyanthi, N. Scalable Blockchain Model Using Off-Chain IPFS Storage for Healthcare Data Security and Privacy. *J. Parallel Distrib. Comput.* **2022**, *164*, 152–167. [\[CrossRef\]](#)
32. Kang, P.; Yang, W.; Zheng, J. Blockchain Private File Storage-Sharing Method Based on IPFS. *Sensors* **2022**, *22*, 5100. [\[CrossRef\]](#)
33. Cagua, G.; Gauthier-Umaña, V.; Lozano-Garzon, C. Implementation and Performance of Lightweight Authentication Encryption ASCON on IoT Devices. *IEEE Access* **2025**, *13*, 16671. [\[CrossRef\]](#)
34. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *J. Cryptol.* **2021**, *34*, 1–42. [\[CrossRef\]](#)
35. Baraiya, H. Comparative Analysis and Benchmarking of ASCON and Grain128AEAD. *Cryptography* **2023**, *7*, 15. [\[CrossRef\]](#)
36. Kandi, A.; Banerjee, A.; Mukhopadhyay, D. Side-Channel and Fault Resistant ASCON Implementation: A Detailed Hardware Evaluation. In Proceedings of the 2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Knoxville, TN, USA, 1–3 July 2024; pp. 1–6. [\[CrossRef\]](#)
37. Adomnicai, A.; Fournier, J.J.A.; Masson, L. Masking the Lightweight Authenticated Ciphers ACORN and ASCON in Software. *Cryptol. ePrint Arch.* **2018**, *2018*, 1–20. [\[CrossRef\]](#)
38. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. Cryptanalysis of Ascon. In *Topics in Cryptology—CT-RSA 2015*; Springer: Cham, Switzerland, 2015; Volume 9048, pp. 371–387. [\[CrossRef\]](#)
39. Xu, R.; Chen, Y.; Blasch, E.; Chen, G. BlendCAC: A Blockchain-Enabled Decentralized Capability-based Access Control for IoTs. *arXiv* **2018**, arXiv:1804.09267. [\[CrossRef\]](#)

40. Bao, Z.; Shi, W.; He, D.; Choo, K.-R.R. IoTChain: A Three-Tier Blockchain-based IoT Security Architecture. *arXiv* **2018**, arXiv:1806.02008. [[CrossRef](#)]
41. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart Contract-Based Access Control for the Internet of Things. *arXiv* **2018**, arXiv:1802.04410. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.