Data Protection in Internet of Medical Things Using Blockchain and Secret Sharing Method

Shreyshi Shree
Carleton University
Chen Zhou
Carleton University
Masoud Barati (masoud.barati@carleton.ca)
Carleton University

Research Article

Keywords: Internet of Medical Things, Online healthcare systems, Data security, Secret sharing algorithm, Blockchain

Posted Date: April 12th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2791374/v1

Additional Declarations: No competing interests reported.

Data Protection in Internet of Medical Things Using Blockchain and Secret Sharing Method

Shreyshi Shree[1], Chen Zhou[2] and Masoud Barati[2]

[1]School of Computing, Newcastle University, Newcastle Upon Tyne, NE4 5TG, UK.
[2]School of Information Technology, Carleton University, Ottawa, K1S 5B6, Canada.

Contributing authors: s.shree2@newcastle.ac.uk; chenzhou4@cmail.carleton.ca; masoud.barati@carleton.ca;

Abstract

Internet of Medical Things (IoMT) combines Internet of Things (IoT) with medical devices to facilitate healthcare, providing affordable solutions, and faster treatments to patients. An increase in IoMT use has led to several security/privacy challenges, since many IoMT devices have not been designed with security and privacy features in mind, which makes them vulnerable to attacks. Furthermore, as security risks and threats affect all the layers developed for IoMT-based architectures, an IoMT network must follow stricter privacy and security specifications compared to other IoT devices. In order to address this, we present a new IoT-based architecture to improve the data privacy, security, and integrity leveraging distributed InterPlanetary File System (IPFS) storage and blockchain. The data captured from medical devices is split into multiple encrypted pieces using Secret Sharing Algorithm (SSA) and these pieces are then stored in distributed IPFS storage hosted on edge and cloud servers, and their copies are verifiable by a blockchain network. The applied SSA method ensures that even if a piece of data is compromised, the original data is neither leaked nor lost. Our proposed architecture is implemented and tested on an IoMT-based monitoring system to investigate its feasibility, scalability, and performance.

Keywords: Internet of Medical Things, Online healthcare systems, Data security, Secret sharing algorithm, Blockchain

1 Introduction

Internet of Medical Things (IoMT) technologies have been rapidly developed within healthcare industries. IoMT has several benefits in healthcare sectors, including automated real-time monitoring, enhanced patient experience, and improved treatment Kumar and Tripathi (2021). In an IoMT ecosystem, patients' data is transferred to edge devices and cloud hosts for processing and analysis. Care givers and healthcare service providers normally use IoMT devices (e.g., smart wearable devices and biosensors) to monitor the patients remotely on a real-time basis and make timely and data-driven decisions. However, the high sensitivity and massive amount of data generated by IoMT devices bring multiple challenges to the IoMT implementation. The cybersecurity vulnerability of IoMT and its awareness among professionals have widely been studied in Thielfoldt (2022). The cloud-based solutions (e.g., Bharati et al (2020)) may handle such a volume of data with scalability, remote availability, and cost effectiveness. However, the cloud environment has its own security challenges, and there have been several breaches over the last few years with cloud-hosted data leaked Arcserve (2022). The data generated by IoMT devices consists of personal and sensitive information that should not be disclosed to the public Dilawar et al (2019). Therefore, storing such data on external cloud services raises security challenges for data privacy and integrity. Moreover, the cloud is not transparent and cloud service providers (CSPs) may retain access to the stored data by maintaining backup copies, even after users request to delete their data. Even if the data is encrypted, the encryption may be broken with higher computational techniques, implemented by attackers, in future. Though access control mechanisms can be implemented to protect the data from unauthenticated access Dilawar et al (2019), a CSP may have an administrative access to the data without users' authorization. Hence, the IoMT data stored in the cloud is vulnerable to malicious CSPs and attackers, and better security and transparency are necessary.

Blockchain and InterPlanetary File System (IPFS) are potential to be integrated with IoMT systems to enhance the security of users' data Barati et al (2021). Blockchains are essentially digital ledgers of transactions, duplicating each other across the network of computer systems on an immutable and tamper-proof structure Boudguiga et al (2017); Wang et al (2019); Barati et al (2019, 2020). The IPFS is a content-addressed, peer-to-peer file system composed of decentralized nodes over the Internet Benet (2014), which maintains node connectivity and file locations through a distributed hash table (DHT) technique. Blockchains can provide beneficial features (e.g., immutability) but with high overhead due to redundancy. Meanwhile, the IPFS is cost efficient for keeping large data, but lacks a trusted security pattern. Therefore, blockchains and the IPFS are often combined to support massive data storage with certain security requirements. For example, Steichen et al (2018) proposed an architecture where the IPFS is extended with an access control mechanism based on a public blockchain. The solution in Kumar and Tripathi (2021) integrated an IPFS cluster with a private blockchain to manage patients, doctors, and

IoMT devices and to provide an appropriate access control method. Another approach is proposed in Nguyen et al (2019), where a trusted and secured data manager maintains access control information in a public blockchain and restricts the access to the medical data stored on IPFS so as to improve the security of data sharing and storage. However, these solutions did not address the problem of malicious IPFS node providers (e.g., a cloud provider hosting a large number of IPFS nodes) accessing the stored data, or only employed encryption which is still vulnerable to attacks.

A secret sharing algorithm (SSA) was proposed by Adi Shamir Shamir (1979) to securely share cryptography keys among multiple participants as pieces. The original secret can be recovered only if a sufficient amount (over a threshold) of pieces are collected. With this characteristic, a secret sharing scheme can protect data even when a cloud provider or an IPFS node provider is malicious or is attacked. Several approaches integrated SSA and blockchain into IoT-based architectures to boost data security Si et al (2019); Cha et al (2021). However, these architectures utilize cloud storage, which lacks transparency and trust of IoT users and has the security risks as already mentioned. A solution was proposed in Kim et al (2022), combining secure secret sharing, the IPFS, and cloud storage to handle IoT generated data of different sensitivity and dynamicity. Yet in the scenario of IoMT, the solution is effectively storing sensitive data in the IPFS with symmetric encryption, which is possible to be broken in the future.

In order to address the drawbacks of past solutions, we propose a novel IoMT-based architecture with IPFS, blockchain technology, and SSA to improve data integrity, trust, and privacy for the massive sensitive data in IoMT. The proposed architecture sends the secret pieces of data generated by the SSA into an IPFS platform. Following that, the data added to the IPFS network is addressed with a unique cryptography hash, allowing it to be content-addressable and verifiable by a blockchain network. The key contributions of this paper are summarised below:

- Identifying practical challenges of sharing sensitive and confidential data within existing IoMT architectures;
- Analyzing security issues arising from external public cloud services;
- Designing a secure cloud architecture for protecting the data generated by IoMT devices;
- Investigating the proposed architecture for security, privacy, scalability, integrity, efficiency, and decentralisation;
- Presenting an IoMT-based scenario for facilitating the implementation of our architecture/solution;
- Evaluating the performance of the architecture by the network latency and the gas consumed by blockchain transactions.

The rest of the paper is structured as follows. Section 2 reviews the related work and identifies limitations of existing solutions. Section 3 proposes an architecture to further enhance data privacy and protection through SSA.

Section 4 describes the implementation of the proposed architecture, presents its data flow along with a case study in an IoMT scenario. Section 5 provides some experimental results in terms of time, overhead, and gas consumption. Section 6 analyses and discusses the security and other benefits of the proposed architecture. Finally, Section 7 concludes the paper and provides the direction for future work.

2 Related Work

With the massive and still increasing amount of IoMT devices, IoMT data security and privacy have gained attention for recent years. Correspondingly, the technologies like blockchain and IPFS along with encryption techniques have been adopted in different solutions to address the issues. An electronic health record sharing solution was proposed in Nguyen et al (2019), which utilizes a public blockchain to provide transparent and trusted access control and uses the IPFS for immutable decentralized data storage. A blockchain-based IoT data sharing solution was proposed in Si et al (2019), which stores light-weighted data summary in blockchains to gain security and keeps major data in out-sourced storage such as cloud storage. The solution used the secret sharing mechanism to protect private keys of nodes. Another solution using blockchains and the IPFS was proposed in Kumar and Tripathi (2021), which integrates a consortium blockchain with the IPFS cluster nodes to support the access control of the data stored in the IPFS. Compared to Nguyen et al (2019), this solution does not include a centralized data manager to coordinate the blockchain and IPFS components. An architecture of edge-based IoMT networks was designed in Nguyen et al (2021), which offloads data to the IPFS for decentralized storage and uses smart contracts to handle user authentication at edge servers. A blockchain-based framework was proposed in Egala et al (2021) that utilizes a public blockchain and smart contracts to handle business logic and access control and employs the IPFS to store data. A cloud architecture was proposed in Cha et al (2021), using blockchain to track data integrity and applies a secret sharing mechanism to protect user data during transmission and in cloud storage. A blockchain-based architecture was designed in Elsayeh et al (2021), which handles data collection with a private blockchain for security and immutability and integrates the IPFS to store data. A layered architecture was proposed in Mehbodniya et al (2021), employing an IPFS cluster and a hybrid blockchain in the edge layer to enhance security during registration of patient and device, and to provide secure data storage. An architecture using a private blockchain and smart contracts was proposed in Mohan et al (2021), which applies an IPFS blockchain technique Dey et al (2017) to extend blockchain data storage. An architecture was implemented in Bigini et al (2021) that combines a private blockchain and the IPFS to demonstrate secure collection, storage and sharing of IoMT data. A private blockchain based architecture was proposed in Bhattacharjya et al (2022) to facilitate and secure the P2P data communication among IoMT devices

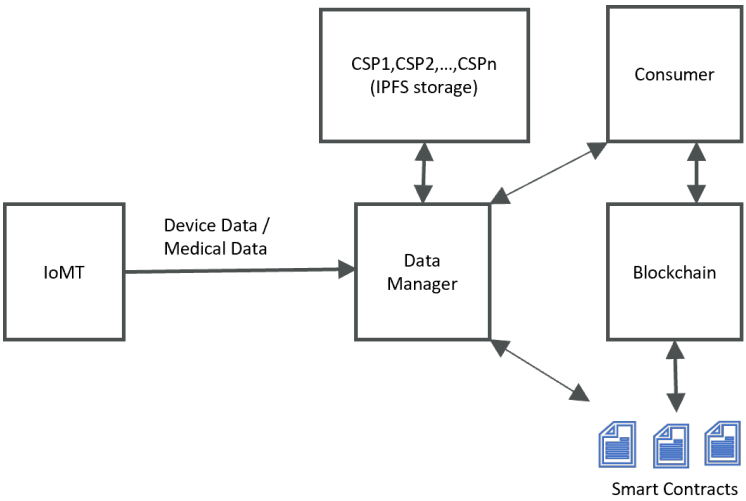


Fig. 1: Proposed IoMT architecture

and participants. A distributed cloud system for data sharing was proposed in Kim et al (2022), which selectively applies blockchain, IPFS, and secure secret sharing technologies to protect data privacy for different types of data.

These continually proposed IoMT solutions has shown that blockchain technology and the IPFS are effective and trending for enhancing data security and privacy in IoMT. The usage of blockchain in IoMT has been reviewed by Hölbl et al (2018); Bigini et al (2020); Gupta et al (2022); Alhaj et al (2022). Interestingly, it can be seen through these surveys that the IPFS was often used with blockchain in IoMT solutions, as an outsourced decentralized data storage. Most of these solutions merely used a symmetric encryption to protect data stored in the IPFS or cloud storage and mostly applied secret sharing for the encryption keys of the data. However, the symmetric encryption may be broken with higher computation power techniques and developed algorithms. The architecture in Kim et al (2022) used secure secret sharing but only for the encryption keys in case of massive static data (e.g., IoMT device generated one). The solution in Cha et al (2021) employed secret sharing scheme for data storage but required multiple cloud storage providers to apply the scheme. Meanwhile, the opaqueness and centralization makes the providers easier to collude compared to decentralized data storage like the IPFS. This paper presents a new architecture implementation that leverages secure secret sharing technique, blockchain technology, and the IPFS, to further enhance the security and privacy of IoMT data.

3 Proposed Architecture

Figure 1 shows our proposed architecture, which consists of a data manger, smart contracts, and a decentralised data storage. The architecture depends on

blockchain to ensure data integrity and employs SSA along with IPFS storage to provide data protection. The description of the components involved in the architecture is provided as follows.

Data manager provides data uploading and sharing functions. It applies SSA to uploaded data and persists the generated data pieces in the decentralised storage. It maintains the addresses of the data pieces and reconstructs the original data from the pieces when an IoMT consumer requests data. It integrates a sub-component (i.e., *secret manager*) to implement SSA and data integrity measures. The data manager is hosted on inter-network layer within the data owner's trusted domain.

Smart contracts maintains registration and integrity information. It defines three smart contracts for device registration, patient registration, and data integrity.

- *Device registration*: This smart contract maintains registration of user devices. It stores device addresses added by users and allows a user to edit their devices from registration. The device addresses are required to be encrypted to protect from unauthorized access.
- *Patient registration*: This smart contract maintains registration of IoMT device users (patients) using their blockchain Wallet IDs. Since this data will be present on the public blockchain, the identifiers are encrypted before reaching the smart contract. The contract allows a patient to register itself and to remove itself from registration.
- *Patient device registration*: This smart contract maintains the relations between devices and patients and associates continually uploaded IoMT data to patients and devices. Whenever new data pieces are stored by the data manager, hashes of their identifier and content will be recorded through this contract.

Security manager is responsible for splitting the original data on the inter-network servers into multiple data pieces using SSA and recovering the original data from the data pieces. A threshold-based secure sharing scheme is employed (i.e., a minimum number of data pieces are required to recover the original data). Therefore, the original data can still be recovered even if one or more data pieces are tampered or lost. Meanwhile, the pieces are distributed to decentralised storage and their relation (i.e., being pieces of the same original data) is secret. Therefore, an attacker or malicious CSP is unlikely to collect sufficient pieces to reconstruct the original data. Different SSA and threshold may be used for different levels of protection. For less critical data (e.g., IoMT data without patient identity), splitting data into two data pieces may be sufficient. For highly confidential data (e.g., data that may identify a user), a threshold higher than two may be more appropriate. Except for the protection, which is based on SSA, the security manager uses the hashes stored in blockchain to verify data integrity. The data flow of this SSA protected data uploading is shown in Fig. 2, in which the SSA and data hash verification are handled by the security manager.

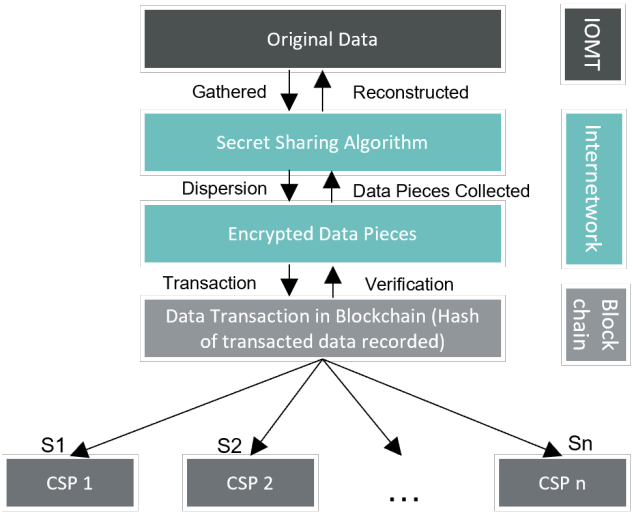


Fig. 2: Secret sharing data flow

Decentralised Storage is provisioned via multiple cloud services to store the secure sharing data pieces. The IPFS is a decentralised file system, but it may be less decentralised if its nodes are mostly hosted by a cloud provider. In the proposed architecture, we recommend using multiple IPFS clusters on different CSPs to highly secure sensitive data. This will significantly reduce the possibility that an attacker or CSP collects enough data pieces to reconstruct the data.

4 Architecture Implementation

This section discusses the implementation of the proposed architecture and describes a prototype implementation. The main components are the data manager, the smart contracts, and the security manager. The decentralised storage is external and is not included in the prototype.[1] The data flow in the architecture is presented to describe the interaction among the components. In this section, an elder people health monitoring scenario is used to demonstrate how the architecture can practically be implemented.

4.1 Implementation of the Data Manager

The data manager is a core component in the architecture, which coordinates the data uploading and data sharing processes. A mobile gateway (e.g., an application in a mobile phone collecting data from local sensors) uploads data

[1]The decentralised storage is an external component. It is expected to be multiple IPFS clusters which have diverse node providers. The IPFS service provided by NFT.Storage is used for our prototype to upload data.

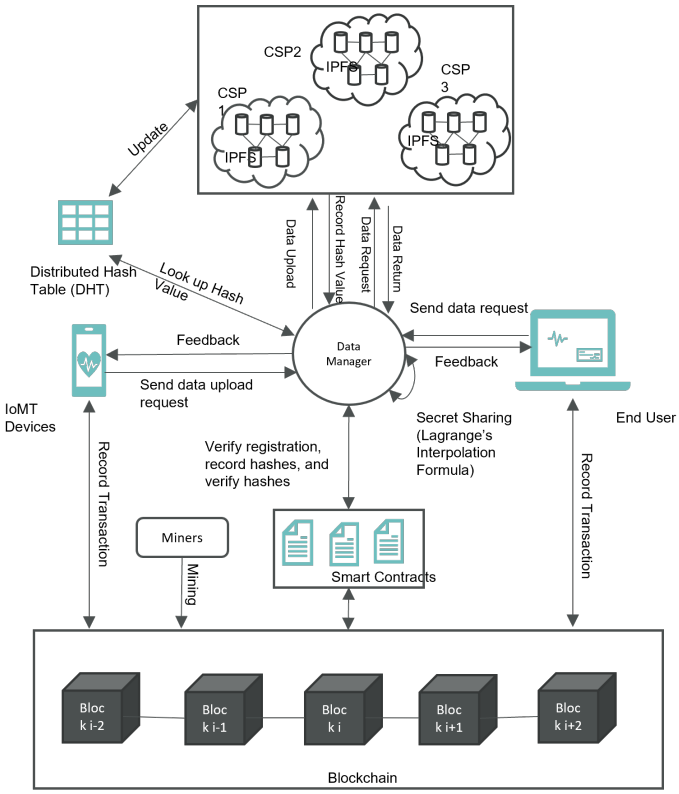


Fig. 3: Data flow within our proposed architecture

to the data manager on behalf of a user (patient). The data manager needs to check the user identity (e.g., the user's Wallet ID in the Ethereum blockchain) through the patient registration smart contract. Also, the device identities of the sensors and the mobile gateway should be checked through the device registration contract and the patient device registration contract. With identities verified, the data manager can receive original data from the mobile gateway. The data manager employs its internal security manager to apply SSA to the uploaded data. Then, the generated sharing data pieces are sent to different IPFS clusters. The content identifiers (CID) of the data pieces in the IPFS clusters should be encrypted and stored in trusted area by the data manager to keep the location of the data pieces secretly. Notably, CIDs are results of content addressing in the IPFS and can ensure data integrity. However, the users hashes of the original data and data pieces can be recorded to the patient device registration contract to transparently ensure the data integrity. These hashes may be used by the mobile gateway to confirm the integrity.

4.2 Implementation of Smart Contracts

The data manager depends on the smart contracts to maintain registration and records of data hashes. The smart contracts contain IDs that can identify patients and devices. Therefore, IDs should be encrypted before sending to the contracts. The original IDs are not important in these contracts, which only maintain the registration of IDs and relations between IDs. Hence, any deterministic encryption can be used for the IDs. In our prototype, the hash values of IDs are used as encrypted IDs, which are 32 bytes derived through SHA-256. Since the hash algorithm is public, a patient can access these contracts through the hashed IDs to check registration and related information. The device registration contract maintains a mapping of the device IDs to devices; the patient (user) registration contract maintains a mapping of the patient IDs to patients. Both contracts provide basic addition, update, check, and delete functions. The patient device registration contract maintains a mapping of the patient IDs to the lists of device IDs. This mapping contains the ownership of devices by patients. The patient device registration contract also maintains a mapping of the patient-device ID pairs to the lists of hashes of uploaded data. This mapping enables the mobile gateway to check the integrity of uploaded data and serves as an evidence of data upload to the data manager. In our prototype, the hashes of SSA generated data pieces are also stored in the patient device registration contract to make them transparent and immutable. However, a practical implementation may remove this mapping to reduce the overhead, when it uses the CIDs of the IPFS to ensure data integrity.

4.3 Implementation of Secret Sharing Algorithm on Data

To protect the data from unauthorized access, a secret sharing scheme is employed in the architecture. The data is split into n data pieces and a minimum of k pieces are needed to recover the data. Shamir's secret sharing algorithm is used in the prototype implementation, but a practical implementation may choose other algorithms as long as they support the (n, k) threshold scheme. Shamir's SSA has a limit to the data size of the secret. Therefore, the data needs to be split into small chunks before Shamir's SSA can be applied. To avoid confusion, data *pieces* refer to the pieces generated by a SSA from the original data; data *chunks* refer to the chunks of a large data (e.g., the original data or a data piece).

Algorithm 1 gives the steps to split the original data into chunks, applies SSA to each chunk, and concatenates chunks of data pieces into final data pieces. The original data is represented by S. It can be split into small chunks $c_1, \cdots, c_m$, where m is the number of chunks. Each chunk $c_j \in S$ is split by a SSA into n chunked pieces, denoted as $c_{1,j}, \cdots, c_{n,j}$, and the minimum number of chunked pieces needed to recover the chunk is k. The chunked pieces concatenate according to their indices and form the final data pieces $S_1, \cdots, S_n$ (i.e., $S_i = c_{i,1} \cdots c_{i,m}$). The final data pieces will be sent to different data storage according to their indices.

Algorithm 1 Split Data with Secret Sharing Algorithm

Input: S, n, k
Output: $S_1, \cdots, S_n$

```
function SCHEME(S, n, k)
    for c_j ∈ CHUNK(S) do
        c_{1,j}, ··· , c_{n,j} ← SSA(c_j, n, k)
        for i ← 1, n do
            S_i ← S_i ‖ c_{i,j}
        end for
    end for
    return S_1, ··· , S_n
end function
```

Algorithm 2 represents Shamir's SSA, where the (chunked) pieces are generated using a random polynomial on a finite field of integers modulo a prime number p. Let s be a chunk of data represented as an integer, n be the number of (chunked) pieces to be generated, k be the threshold for recovering s, p be a large prime number where $p > s$ and $p > n$, $a_0, \cdots, a_{k-1}$ be the coefficients, and $s_1, \cdots, s_n$ be the generated chunked pieces for chunk s. The value of s is given to a_0, and the other $k-1$ coefficients $a_1, \cdots, a_{k-1}$ are randomly generated with distinct integers in $[0, p)$. A polynomial with $k-1$ degree can be represented by these coefficients, and the polynomial is evaluated for each $x \in \{1, \cdots, n\}$. An evaluation result q for $x = i$ is the i-th chunked piece s_i for chunk s. Notably, modulo is necessary during calculation, since the finite field of integers modulo p is used. Normally, a generated chunked piece should be a pair $(x, q(x))$ representing an interpolation point. However, the x values are not returned in Algorithm 2, as they are fixed values $1, \cdots, n$ equal to the index numbers.

Algorithm 3 provides the steps to reconstruct the original data from data pieces using Lagrange interpolation. Ideally, all n data pieces $S_1, \cdots, S_n$ are available, but k pieces are enough to reconstruct the original data S. Let $X = \{i_1, \cdots, i_k\}$ be the index numbers of the data pieces to be used, $S_X = \{S_{i_1}, \cdots, S_{i_k}\}$ be these pieces, k be the threshold for recovering the original data S. Firstly, a Chunk function is used to split each piece S_i into chunks $c_{i,1}, \cdots, c_{i,m}$. Secondly, for j from 1 to m, the j-th chunks of all pieces in S_X are collected as C_j. Then, the j-th chunk of the original data S, denoted by c_j, can be calculated using Lagrange interpolation with X, C_j, k and $x = 0$. Finally, the chunks c_j are concatenated to form the original data S.

Algorithm 4 represents Lagrange interpolation on a finite field of integers modulo p, where p is a prime number. Let $X = x_1, \cdots, x_k$ and $Y = y_1, \cdots, y_k$ be the x and y values of interpolation points, and $k-1$ be the degree of the Lagrange polynomial. The y value at x can be calculated using the Lagrange interpolation. First, the $l_i(x)$ values for $i \in 1, \cdots, k$ are calculated. Then, y is calculated according to $\Sigma_{i=1}^{k} y_i l_i(x)$. Note that modulo is used during calculation because this Lagrange interpolation is on finite field of integers

Algorithm 2 Shamir's Secret Sharing Algorithm

Input: s, n, k
Output: s_i

```
1: function SSA(s, n, k)
2:     a_0 ← s
3:     for i ← 1, k − 1 do
4:         a_i ← RandomNumber(0, p)
5:     end for
6:     for i ← 1, n do
7:         q ← a_0
8:         x ← i
9:         y ← 1
10:        for j ← 1, k − 1 do
11:            y ← y × x  mod p
12:            q ← q + a_j × y  mod p
13:        end for
14:        s_i ← q
15:    end for
16:    return s_1, ··· , s_n
17: end function
```

modulo p. The division operation on this field is calculated with a function named "Divmod".

Algorithm 5 shows the division operation on the finite field of integers modulo p. Let a, b, and p be the dividend, the divisor, and the prime number for modulo operation. Dividing a by b is equivalent to multiplying a with b^{-1} (the multiplicative inverse of b). On a finite field of integers modulo p, b^{-1} can be found with the extended greatest common divisor algorithm (extended GCD). The lines 2 to 11 of Algorithm 5 show the usage of extended GCD to find b^{-1} and line 12 calculates the division of a by b modulo p.

Optimizations exist in practice for the calculation of Shamir's SSA, though we didn't implement them in our prototype. For example, if a Mersenne prime like $2^{127} - 1$ is used as p, the modulo operation can be replaced by binary `xor` and shift operations, which are cheaper and faster for computers. The following equations explain this usage, where $>>$ is the right shift binary operation:

$$x \equiv (x \mod 2^{127}) + \left\lfloor \frac{x}{2^{127}} \right\rfloor \mod 2^{127} - 1$$

Hence,

$$x \equiv x \oplus (2^{127} - 1) + (x >> 127) \mod 2^{127} - 1$$

This equation can be applied recursively until the value of the equation is smaller than $2^{127} - 1$ to calculate the modulo of $2^{127} - 1$.

Algorithm 3 Reconstructing data pieces using Lagrange's Interpolation

Input: X, S_X, k

Output: S

function RECONSTRUCT(X, S_X, k)
  for $i \in X$ **do**
    $c_{i,1}, \cdots, c_{i,m} \leftarrow \text{Chunk}(S_i)$
  end for
  for $j \leftarrow 1, m$ **do**
    $C_j \leftarrow \emptyset$
    for $i \in X$ **do**
      C_j.add($c_{i,j}$)
    end for
    $c_j \leftarrow$ LAGRANGEINTERPOLATION($X, C_j, k, 0$)
    $S \leftarrow S \| c_j$
  end for
  return S
end function

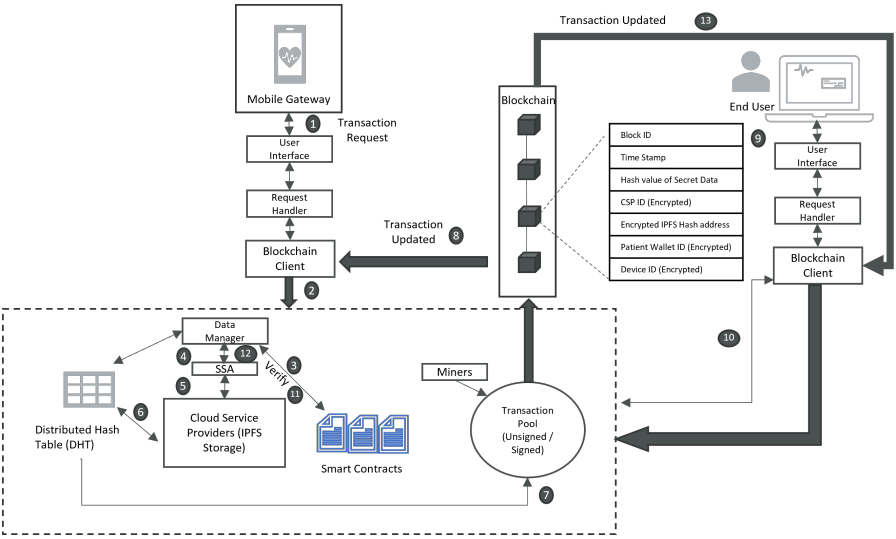


Fig. 4: Data uploading and sharing process

4.4 Data Flow in Data Uploading and Sharing

Figure 4 shows the data uploading and sharing process. In a data uploading process, a mobile gateway initiates a request for uploading data to the data manager through a blockchain client. The blockchain client forwards the request to the data manager and listens to the events in smart contracts. The data manager verifies the request (e.g., access rights, registration status, etc.)

Algorithm 4 Lagrange's Interpolation over a finite field of integers mod p

Input: X, Y, k, x
Output: y

function LagrangeInterpolation(X, Y, k, x)
  $y \leftarrow 0$
  for $i \leftarrow 1, k$ **do**
    $l \leftarrow 1$
    $d \leftarrow 1$
    for $j \leftarrow 1, k$ **do**
      if $i \neq j$ **then**
        $l \leftarrow l \times (x - x_j) \mod p$
        $d \leftarrow d \times (x_i - x_j) \mod p$
      end if
    end for
    $l \leftarrow$ Divmod(l, d, p)
    $y \leftarrow y + l \times y_i \mod p$
  end for
  return s
end function

Algorithm 5 Division in modulo finite field using extended GCD

Input: a, b, p
Output: $a/b \mod p$

function Divmod(a, b, p)
  $t \leftarrow 0$
  $t_{new} \leftarrow 1$
  $r \leftarrow p$
  $r_{new} \leftarrow b$
  while $r_{new} \neq 0$ **do**
    $q \leftarrow \lfloor r / r_{new} \rfloor$
    $t, t_{new} \leftarrow t_{new}, t - q \times t_{new}$
    $r, r_{new} \leftarrow r_{new}, r - q \times r_{new}$
  end while
  $t \leftarrow t \mod p$ ▷ t is b^{-1}, in other words, $b \times t \equiv 1 \mod p$
  return $a \times t \mod p$
end function

via the smart contracts. If the request is approved, the data manager starts receiving the data uploaded from the gateway. The data manager redirects the data to its security manager which applies a SSA to the data and generates the secure sharing data pieces. The data pieces are then uploaded to corresponding IPFS clusters. The IPFS clusters will update the DHT and return the CIDs of the data pieces to the data manager. After all data pieces are persisted, the data manager updates the information to the patient device smart

contract. An update event is emitted by the contract to notify the mobile gateway through the blockchain client. In a data sharing process, an end user willing to access data (e.g., doctor) listens to the smart contracts through a blockchain client. Once the end user receives the data upload event emitted by the smart contracts, it sends a request to the data manager to acquire the new data. The data manager verifies the request using the smart contracts. If the request is fine, the data manager asks its security manager to recover the original data from the data pieces on IPFS clusters. The data manager then provides the original data to the end user.

4.5 Case Study: Health Monitoring in Smart Home for Elder People

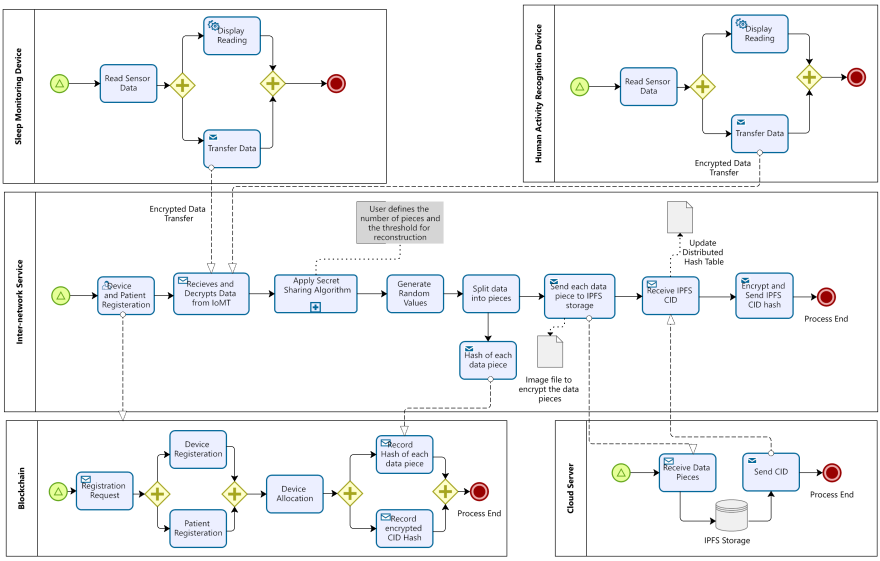


Fig. 5: Business process for uploading data from IoMT devices

One application of IoMT is to facilitate regular health check-up of risk groups (e.g., Haghi et al (2022), Laplante et al (2018)). Figure 5 and 6 demonstrate the business process flow of our proposed architecture in a scenario of elder people health monitoring in a smart home. First, the patient and the devices (i.e., the sleep monitor and the human activity recognition device) are registered through the smart contracts. In this process, the relation between the patient and the devices are also registered. Then, the data is collected from the monitoring devices and split into secure sharing data pieces through SSA. The data pieces are uploaded to IPFS clusters and the CIDs of the data pieces are returned by the IPFS. The hashes of the CIDs and data pieces are

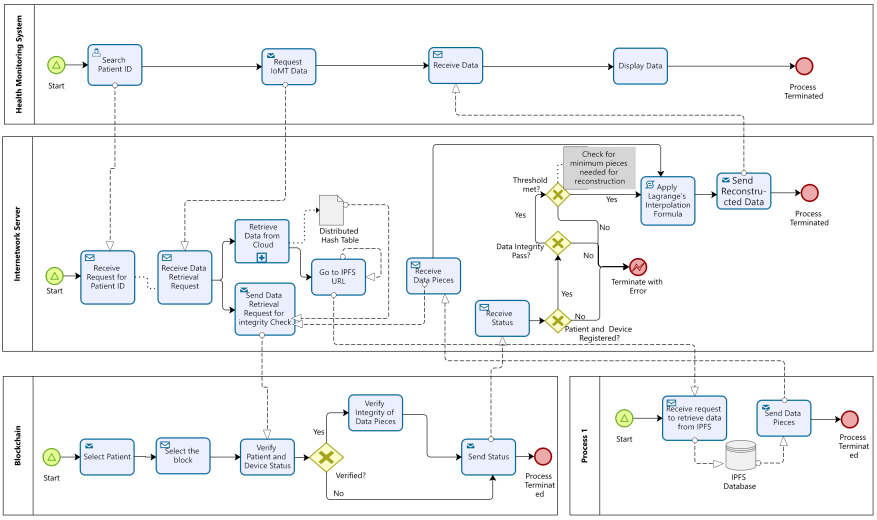


Fig. 6: Business process for retrieving of data by health monitoring system

reported to the blockchain for audit and integrity purposes. When a monitoring system requests for the data of the patient, the patient ID and device IDs are checked through the smart contracts. Then, the data pieces are retrieved from the IPFS clusters. The hashes of the CIDs and data pieces are checked through the smart contracts to double check their integrity. Finally, the original data is reconstructed from the data pieces through SSA and sent to the monitoring system for analysis and display.

5 Experimental Results

To implement the proposed IoMT architecture for the case study represented in subsection 4.5, we use public Ethereum network, Ropsten, as well as NFT.Storage for storing data in IPFS. We evaluate the performance of our solution based on the gas and time consumed for validating transactions. We also implemented a user interface for the purpose of experiment using JavaScript. Moreover, our smart contracts were implemented to register a patient (user), register a device, allocate device to the patient, check status of the device and patient, remove the device's data, remove the patient's information, upload and retrieve data.

5.1 Uploading and Retrieving Data

For data upload, each data is processed using secret sharing algorithm with ($n = 4, k = 3$) and are split into four pieces, y_1, y_2, y_3, and y_4, ($n = 4$), and each piece is then stored on IPFS storage which provides a CID associated with each piece. These CIDs are then stored on DHT maintained by data manager and are encrypted using private key by data manager before uploading to the

Table 1: Performance evaluation for uploading and downloading

Data size	Time to split (ms)	Time to upload (ms)	Time to download (ms)
1	0.19	995	20.497
2	0.38	1990	40.994
3	0.57	2985	61.491
4	0.76	3980	81.988
5	0.95	4975	102.485
6	1.14	5970	122.982
7	1.33	6965	143.479
8	1.52	7960	163.976
9	1.71	8955	184.473
10	1.9	9950	204.97

blockchain transaction. Finally, we upload encrypted wallet address, encrypted device ID, encrypted CID and hash of data pieces to the blockchain transaction for verification and integrity check.

In our experiment, we provide secret key/private keys to upload each data piece to the IPFS storage and only the authorised users can have access to those key set by the data manager. Since we are using free version of IPFS for this study, we face a limitation with uploading large files. However, to test the secret sharing algorithm and evaluate the performance, we upload small files with a few values and then retrieved the data. We deployed the algorithm to test the feasibility and integrated it into a blockchain network.

For retrieving data, we reconstruct the original data only with three data pieces (based on the health monitoring scenario). Table 1 shows the time taken to upload the files to IPFS storage and to download them, which increases as the size of data. Hence, using this framework should consider the time sensitivity versus data criticality.

Regarding data retrieval from the IPFS storage, a user (e.g., a physician) requests to access data using health monitoring system after user authentication. The user requests to fetch the data for a particular patient and a device registered by that patient. Once the request is received by the inter-network server, the data manager retrieves the data from blockchain and decrypts the CID to access the data on IPFS. The data retrieved from the IPFS is verified against the hash of the data on the blockchain for integrity. The user provides the secret key to decrypt the data retrieved from the IPFS storage which is then processed using the Lagrange's Interpolation formula to reconstruct original data shared by the IoMT device.

The time taken to split the data using secret sharing algorithm can be seen as an overhead to store the data on IPFS. With this experiment, we noted that the data split process does not have a major impact on the network latency, since the time taken for splitting the data is a fraction of microseconds as shown in Table 1 when compared to the overhead due to uploading data on the IPFS network, the blockchain network, and reconstruction.

Table 2: Deployment costs

Transactions	Gas used
Device Registration	45374
Patient Registration	52123
Device Allocation	84021
Data Upload (all 4 pieces)	2388727
Updating Blockchain	231285
Delete data	20684

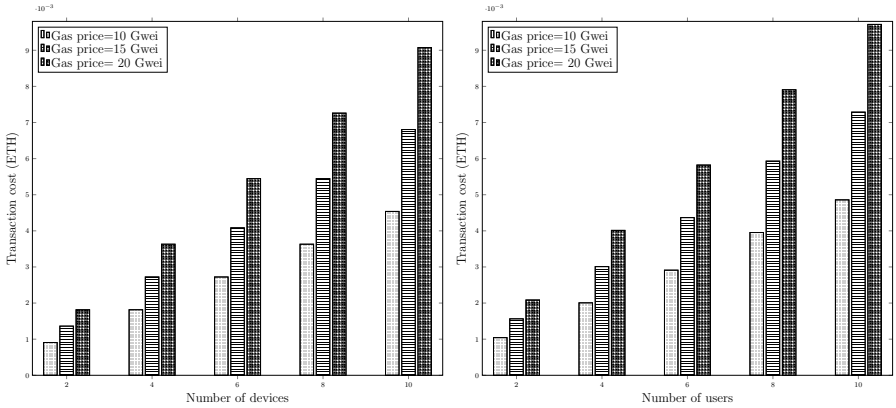
Table 3: Number of Registrations vs Costs

No. of devices/users	Device registration	Device allocation	Patient registration
1	45374	84021	52123
2	90748	168042	104242
3	136122	252063	155300
4	181496	336084	200624
5	226870	420105	245858
6	272244	504126	291092
7	317618	588147	343215
8	362992	672168	395338
9	408366	756189	440572
10	453740	840210	485806

5.2 Device and Patient Registration

As an initial step, we first register the devices and patients (users) on blockchain and select the execution time (seconds) for the transaction by varying the transaction fees. The user can opt for higher transaction fees for faster execution. The assumption is that the number of devices or users varies from 1 to 10. We implemented our smart contracts using Solidity language and deployed them in Ethereum test network (Ropsten). Table 2 represents the deployment costs (wei) required for running our smart contracts. To calculate the costs (used gas) for executing device registration, allocations, and user/-patient registration contracts, their functions have been tested using Ropsten. Table 3 shows the trend of gas consumption, transaction fees based on number of devices being registered and the gas used to allocate devices per user. As seen, rising the number of patients or devices will directly effect on an increase in the gas used for their registration.

The transactions costs for both device registration and patient registration contracts have also been calculated in Ether. The results of this experiment have been obtained with regards to different gas price units (i.e., 10, 15, and 20 Gwei). Figure 7 illustrates the costs in ETH, where the number of devices/users varies from 2 to 10. Since more data is collected through the patient registration transaction compared with the device registration one, more fees must be paid to the miners for registering users through blockchain. Furthermore, when the amount of gas price unit increases, the cost rises sharply. The reason of choosing higher gas price unit is to encourage miners for validating transactions faster.



(a) Device registration costs (b) User/patient registration costs

Fig. 7: Transaction costs

5.3 Investigating Mining Time

This experiment evaluates the mining time taken for running the patient and device registration over the blockchain. We used Ropsten test network to obtain the results, as it gives a measurement of the time taken from activation to mining of a transaction. The average mining time for executing the device and patient registration with different gas prices (i.e., 10, 15, and 20 Gwei) was calculated. The functions of our smart contracts were executed ten times to get the average time. Figures 8 and 9 demonstrate the results. As seen from the figures, when the gas price increases, the mining time decreases noticeably, since the miners will be encouraged to process the transactions sooner.

Given a fixed gas price, we observe a fluctuation in the trend of mining process. The reason is that when the transactions are released in the network with the same gas price, the miners normally follow an arbitrary manner for selecting and validating a transaction.

6 Discussion

This section discusses the security analysis of the proposed architecture in the two threat scenarios and further analyzes the advantages and potential challenges of the architecture.

6.1 Security Analysis

Since IoMT data consists of health records of an individual, it is crucial to maintain its security against potential threats so that it is not leaked or misused. In this section, we discuss how the proposed architecture can safeguard the data and help preserve the privacy and integrity of the data. For an example, assume that an attacker has gained access to IPFS storage without the

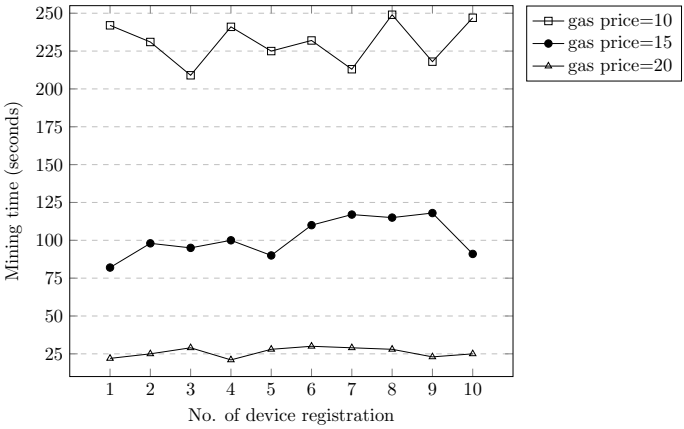


Fig. 8: The time taken for mining the device registration transactions

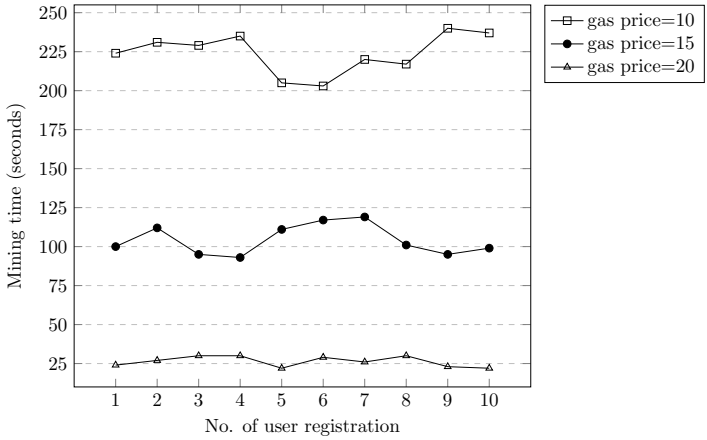


Fig. 9: The time taken for mining the user registration transactions

data manager's approval. Using the proposed architecture, all data uploaded to the IPFS are encrypted using AES 256 bit encryption using a secret key. To retrieve the data, the attacker needs to know the secret key, which is set by the data manager. Moreover, even if the attacker is able to gain access to the secret key, they need to get hold of at least minimum number of data pieces to reconstruct the data which means that they need to retrieve data from more than one IPFS storage hosted with different CSP. However, it is extremely challenging to gain unauthorised access to multiple CSP servers and to retrieve the data.

6.2 Privacy

By using blockchain, smart contracts, and IPFS storage, the proposed architecture using secret sharing algorithm guarantees data privacy. Using smart contracts for user verification protects any malicious user access, which can be further integrated with end-to-end access control for authentication and authorization using smart contracts to block user IDs as potential threats. With the usage of secret sharing algorithm, the data is split into unrecognisable data pieces maintaining the privacy of original data even if a few data pieces are compromised. The only possibility to compromise the original data is if the attacker could gain access to the minimum number of data pieces required to reconstruct the original data. Such a situation is extremely challenging to occur, where an attacker gains unauthorised access to multiple CSPs or IPFS storage. In a scenario where the data is split into two pieces and both the pieces are required to reconstruct the data (n=2, k=2), and the attacker is successful in gaining access to both pieces, the original data can be retrieved, but the possibility of such a scenario is extremely low. In case, there is a smaller number of CSPs or a single CSP for storing data, the data pieces on IPFS storage are encrypted, but in order to preserve privacy, it is recommended to increase the number of pieces and the threshold to reconstruct the original data.

6.3 Scalability

IoT devices generate massive amount of real-time data including sensitive medical data particularly published by IoMT objects. In real-time scenarios when such volume of data is generated on a regular basis, the scalability of the system is crucial. The proposed architecture meets this requirement by provisioning scalable IPFS storage based on cloud services and blockchain network. In this case, even if the system is designed to use only a single cloud service, all data pieces generating from the secret sharing algorithm can be maintained on a single IPFS storage which is dynamically scalable.

6.4 Integrity

In the proposed architecture, the data integrity is managed by the blockchain network. Blockchain with smart contacts and IPFS storage are immutable technologies and guarantee integrity, since every transaction on the blockchain network are on consensus basis. Hence, an unauthorised or illegal transaction will be invalidated and will not be allowed in the network by the consensus process. For integrity, it is required that the IoMT data is not modified by any unauthorised user. In a blockchain, each block has the hash value of previous block with timestamp along with transaction data such as hash value of data pieces, encrypted IPFS CID, encrypted wallet ID and encrypted device ID. These data are provided by the inter-network server which implements the secret sharing algorithm. When these data are recorded on blockchain network, they are verified by the miners and only if the verification or validation is successfully completed, these data can be added to the blockchain. Moreover,

a CSP also maintains the integrity of data using the blockchain network and requires penalties for falsifying the data. The ledger maintained by blockchain is tamper-proof, and hence any falsification can be easily identified. With secret sharing algorithm, once the data pieces are stored on IPFS storage and when threshold is 3 or more and the number of pieces is 4 or more, even if one of the pieces are modified, it will not impact on the original data. Therefore, using this architecture, the data integrity can be ensured.

6.5 Efficiency

The proposed architecture provides efficient privacy and security. However, it has a high system overhead in terms of network latency and associated costs. The experiment done as part of this study was on a small scale and by using free version of IPFS storage which has limit on uploading the data. Hence, it is recommended to conduct an analysis on the paid version of IPFS to estimate the real-time performance. However, the functions defined as part of the architecture are efficient to store and share the data securely on the cloud using blockchain and secret sharing algorithm.

6.6 Decentralisation

We proposed a decentralised cloud storage using IPFS storage for storing IoMT data since the centralised cloud storage suffers several challenges such as denial of service attacks or data leakage, which lead to single points of failure and make it difficult to recover. This paper presented an approach of storing data on the decentralised IPFS storage on multiple distributed cloud nodes. Because blockchain can verify the integrity of data from the user's side, the service is securely available to the IoMT device owners and the end users. The proposed architecture provides better privacy and integrity in a decentralised storage with secret sharing of data.

7 Conclusion

This paper proposed a new IoMT-based architecture that leverages secret sharing algorithm for protecting patients' data from unauthorized cloud service providers and external adversaries. The architecture made use of a blockchain-based technique and a distributed secure storage (i.e., IPFS) to enhance the privacy and integrity of IoMT users' information. We implemented smart contracts in order to register and verify both patients and smart medical devices. We also defined a threshold value for join-able data pieces required to reconstruct the original data. An adversary or a cloud provider cannot retrieve the original data if the number of pieces leaked is less than the threshold. To investigate the performance of our proposed architecture, we implemented an Ethereum blockchain on Ropsten testnet, where IoMT data can be uploaded after being split into four pieces by secret sharing algorithm. The experimental results showed that when the data is uploaded, a single file is split into

multiple pieces, which increases the time consumed and gas usage to upload a larger dataset. Moreover, the evaluation of mining time indicated that miners can validate blocks in an arbitrary manner when the transactions are released in a blockchain network with the same gas price value.

Future work will focus on implementing the proposed architecture on existing IoT testbeds to practically assess the scalability and feasibility of our solution. Furthermore, integrating an authentication/authorization mechanism such as access control into the architecture implementation can improve the distribution of secret keys in a more secure manner. Checking the compliance of the designed architecture with available privacy regulations (e.g., General Data Protection Regulation (GDPR)) is another direction for future investigation.

8 Declarations

8.1 Ethical Approval

This declaration is not applicable.

8.2 Competing Interests

We have no interests of a financial or personal nature.

8.3 Authors' Contributions

Shreyshi proposed the architecture idea, wrote the paper, conducted several experimental results. Chen worked on the relate work and SSA algorithms. Masoud validated the architecture, conducted some evaluations, wrote some parts of the paper, and supervised the authors.

8.4 Funding

No funding is received for publishing the paper.

8.5 Availability of Data and Materials

This declaration is not applicable.

References

Alhaj TA, Abdulla SM, Iderss MAE, et al (2022) A survey: To govern, protect, and detect security principles on internet of medical things (IoMT). IEEE Access 10:124,777–124,791. https://doi.org/10.1109/access.2022.3225038, URL https://doi.org/10.1109/access.2022.3225038

Arcserve (2022) 7 most infamous cloud security breaches. https://www.arcserve.com/blog/7-most-infamous-cloud-security-breaches, [Accessed: Nov. 10, 2022]

Barati M, Rana O, Theodorakopoulos G, et al (2019) Privacy-aware cloud ecosystems and GDPR compliance. In: 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, pp 117–124, https://doi.org/10.1109/FiCloud.2019.00024, URL https://doi.org/10.1109/FiCloud.2019.00024

Barati M, Rana O, Petri I, et al (2020) GDPR compliance verification in internet of things. IEEE Access 8:119,697–119,709. https://doi.org/10.1109/ACCESS.2020.3005509, URL https://doi.org/10.1109/ACCESS.2020.3005509

Barati M, Buchanan WJ, Lo O, et al (2021) A privacy-preserving distributed platform for COVID-19 vaccine passports. In: 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion. IEEE/ACM, pp 1–6, https://doi.org/10.1145/3492323.3495626, URL https://doi.org/10.1145/3492323.3495626

Benet J (2014) Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:14073561

Bharati S, Podder P, Mondal MRH, et al (2020) Applications and challenges of cloud integrated IoMT. In: Cognitive Internet of Medical Things for Smart Healthcare. Springer International Publishing, p 67–85, https://doi.org/10.1007/978-3-030-55833-8_4, URL https://doi.org/10.1007/978-3-030-55833-8_4

Bhattacharjya A, Kozdrój K, Bazydło G, et al (2022) Trusted and secure blockchain-based architecture for internet-of-medical-things. Electronics 11(16):2560. https://doi.org/10.3390/electronics11162560, URL https://doi.org/10.3390/electronics11162560

Bigini G, Freschi V, Lattanzi E (2020) A review on blockchain for the internet of medical things: Definitions, challenges, applications, and vision. Future Internet 12(12):208. https://doi.org/10.3390/fi12120208, URL https://doi.org/10.3390/fi12120208

Bigini G, Freschi V, Bogliolo A, et al (2021) Decentralising the internet of medical things with distributed ledger technologies and off-chain storages: A proof of concept. In: Smart Objects and Technologies for Social Good. Springer International Publishing, p 80–90, https://doi.org/10.1007/978-3-030-91421-9_7, URL https://doi.org/10.1007/978-3-030-91421-9_7

Boudguiga A, Bouzerna N, Granboulan L, et al (2017) Towards better availability and accountability for IoT updates by means of a blockchain. In: 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, https://doi.org/10.1109/eurospw.2017.50, URL https://doi.org/10.1109/eurospw.2017.50

Cha J, Singh SK, Kim TW, et al (2021) Blockchain-empowered cloud architecture based on secret sharing for smart city. Journal of Information Security and Applications 57:102,686. https://doi.org/10.1016/j.jisa.2020.102686, URL https://doi.org/10.1016/j.jisa.2020.102686

Dey T, Jaiswal S, Sunderkrishnan S, et al (2017) HealthSense: A medical use case of internet of things and blockchain. In: 2017 International Conference on Intelligent Sustainable Systems (ICISS). IEEE, https://doi.org/10.1109/iss1.2017.8389459, URL https://doi.org/10.1109/iss1.2017.8389459

Dilawar N, Rizwan M, Ahmad F, et al (2019) Blockchain: Securing internet of medical things (IoMT). International Journal of Advanced Computer Science and Applications 10(1). https://doi.org/10.14569/ijacsa.2019.0100110, URL https://doi.org/10.14569/ijacsa.2019.0100110

Egala BS, Pradhan AK, Badarla V, et al (2021) Fortified-chain: A blockchain-based framework for security and privacy-assured internet of medical things with effective access control. IEEE Internet of Things Journal 8(14):11,717–11,731. https://doi.org/10.1109/jiot.2021.3058946, URL https://doi.org/10.1109/jiot.2021.3058946

Elsayeh M, Ezzat KA, El-Nashar H, et al (2021) Cybersecurity architecture for the internet of medical things and connected devices using blockchain. Biomedical Engineering: Applications, Basis and Communications 33(02):2150,013. https://doi.org/10.4015/s1016237221500137, URL https://doi.org/10.4015/s1016237221500137

Gupta S, Sharma HK, Kapoor M (2022) Internet of medical things (IoMedT) vs internet of things (IoT). In: Blockchain for Secure Healthcare Using Internet of Medical Things (IoMT). Springer International Publishing, p 27–37, https://doi.org/10.1007/978-3-031-18896-1_3, URL https://doi.org/10.1007/978-3-031-18896-1_3

Haghi M, Spicher N, Wang J, et al (2022) Integrated sensing devices for disease prevention and health alerts in smart homes. In: Studies in Health Technology and Informatics. IOS Press, https://doi.org/10.3233/shti220007, URL https://doi.org/10.3233/shti220007

Hölbl M, Kompara M, Kamišalić A, et al (2018) A systematic review of the use of blockchain in healthcare. Symmetry 10(10):470. https://doi.org/10.3390/sym10100470, URL https://doi.org/10.3390/sym10100470

Kim TW, Azzaoui A, Koh B, et al (2022) A secret sharing-based distributed cloud system for privacy protection. Hum Centric Comput Inf Sci 12

Kumar R, Tripathi R (2021) Towards design and implementation of security and privacy framework for internet of medical things (IoMT) by leveraging blockchain and IPFS technology. The Journal of Supercomputing 77(8):7916–7955. https://doi.org/10.1007/s11227-020-03570-x, URL https://doi.org/10.1007/s11227-020-03570-x

Laplante PA, Kassab M, Laplante NL, et al (2018) Building caring healthcare systems in the internet of things. IEEE Systems Journal 12(3):3030–3037. https://doi.org/10.1109/jsyst.2017.2662602, URL https://doi.org/10.1109/jsyst.2017.2662602

Mehbodniya A, Neware R, Vyas S, et al (2021) Blockchain and IPFS integrated framework in bilevel fog-cloud network for security and privacy of IoMT devices. Computational and Mathematical Methods in Medicine 2021:1–9. https://doi.org/10.1155/2021/7727685, URL https://doi.org/10.1155/2021/7727685

Mohan D, Alwin L, Neeraja P, et al (2021) A private ethereum blockchain implementation for secure data handling in internet of medical things. Journal of Reliable Intelligent Environments 8(4):379–396. https://doi.org/10.1007/s40860-021-00153-2, URL https://doi.org/10.1007/s40860-021-00153-2

Nguyen DC, Pathirana PN, Ding M, et al (2019) Blockchain for secure EHRs sharing of mobile cloud based e-health systems. IEEE Access 7:66,792–66,806. https://doi.org/10.1109/access.2019.2917555, URL https://doi.org/10.1109/access.2019.2917555

Nguyen DC, Pathirana PN, Ding M, et al (2021) BEdgeHealth: A decentralized architecture for edge-based IoMT networks using blockchain. IEEE Internet of Things Journal 8(14):11,743–11,757. https://doi.org/10.1109/jiot.2021.3058953, URL https://doi.org/10.1109/jiot.2021.3058953

Shamir A (1979) How to share a secret. Communications of the ACM 22(11):612–613. https://doi.org/10.1145/359168.359176, URL https://doi.org/10.1145/359168.359176

Si H, Sun C, Li Y, et al (2019) IoT information sharing security mechanism based on blockchain technology. Future Generation Computer Systems 101:1028–1040. https://doi.org/10.1016/j.future.2019.07.036, URL https://doi.org/10.1016/j.future.2019.07.036

Steichen M, Fiz B, Norvill R, et al (2018) Blockchain-based, decentralized access control for IPFS. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, https://doi.

org/10.1109/cybermatics_2018.2018.00253, URL https://doi.org/10.1109/cybermatics_2018.2018.00253

Thielfoldt K (2022) Internet of medical things cybersecurity vulnerabilities and medical professionals' cybersecurity awareness: A quantitative study. PhD thesis, Colorado Technical University

Wang X, Zha X, Ni W, et al (2019) Survey on blockchain for internet of things. Computer Communications 136:10–29. https://doi.org/10.1016/j.comcom.2019.01.006, URL https://doi.org/10.1016/j.comcom.2019.01.006