future internet

Article

Securing Environmental IoT Data Using Masked Authentication Messaging Protocol in a DAG-Based Blockchain: IOTA Tangle

Pranav Gangwani [1], Alexander Perez-Pons [1,*], Tushar Bhardwaj [2], Himanshu Upadhyay [2], Santosh Joshi [2] and Leonel Lagos [2]

[1] Department of Electrical & Computer Engineering, Florida International University, Miami, FL 33199, USA; pgang002@fiu.edu
[2] Applied Research Center, Florida International University, Miami, FL 33199, USA; tbhardwa@fiu.edu (T.B.); upadhyay@fiu.edu (H.U.); sajoshi@fiu.edu (S.J.); lagosl@fiu.edu (L.L.)
* Correspondence: aperezpo@fiu.edu

Abstract: The demand for the digital monitoring of environmental ecosystems is high and growing rapidly as a means of protecting the public and managing the environment. However, before data, algorithms, and models can be mobilized at scale, there are considerable concerns associated with privacy and security that can negatively affect the adoption of technology within this domain. In this paper, we propose the advancement of electronic environmental monitoring through the capability provided by the blockchain. The blockchain's use of a distributed ledger as its underlying infrastructure is an attractive approach to counter these privacy and security issues, although its performance and ability to manage sensor data must be assessed. We focus on a new distributed ledger technology for the IoT, called IOTA, that is based on a directed acyclic graph. IOTA overcomes the current limitations of the blockchain and offers a data communication protocol called masked authenticated messaging for secure data sharing among Internet of Things (IoT) devices. We show how the application layer employing the data communication protocol, MAM, can support the secure transmission, storage, and retrieval of encrypted environmental sensor data by using an immutable distributed ledger such as that shown in IOTA. Finally, we evaluate, compare, and analyze the performance of the MAM protocol against a non-protocol approach.

Keywords: IoT; security; privacy; environment; IOTA; Tangle; MAM; directed acyclic graph; blockchain; distributed ledger

Citation: Gangwani, P.; Perez-Pons, A.; Bhardwaj, T.; Upadhyay, H.; Joshi, S.; Lagos, L. Securing Environmental IoT Data Using Masked Authentication Messaging Protocol in a DAG-Based Blockchain: IOTA Tangle. *Future Internet* **2021**, *13*, 312. https://doi.org/10.3390/fi13120312

Academic Editor: Christoph Stach

Received: 22 October 2021
Accepted: 4 December 2021
Published: 6 December 2021

1. Introduction

1.1. Mobile and Electronic Environment

Current technological and economic advancements are exerting a tremendous influence on the environment, to the extent of raising severe concerns about climate change and pollution. Human activities have an undeniable and ever-increasing impact on the climate system, along with recent developments that are unprecedented and currently acknowledged by the Intergovernmental Panel on Climate Change [1]. Environmental monitoring in this context refers to an Internet of Things (IoT) system where sensors are used to collect useful data about the ecosystem, leading to further discoveries and a better and more comprehensive understanding, to execute specific actions in mitigating and addressing the degradation of the environment [2]. Environmental monitoring in indoor environments is another related field that is now gaining popularity. This has proved essential not only for the building's or housing's residents [3] but also in terms of lowering greenhouse gas emissions [4]. Temperature, humidity, rainfall, atmospheric pressure, light intensity, and air quality, which are impacted by pollutants such as carbon dioxide (CO_2), carbon monoxide (CO), sulfur oxide (SO_x), volatile organic compounds, and many more are among the most commonly measured parameters. CO is a gas that is colorless and odorless but can cause serious harm to the human population and to the environment. When a substance is burned, smoke and fumes are released containing CO. SO_x is a group of sulfur chemicals that cause serious harm to the environment. Therefore, monitoring of the environment and pollutants is required to achieve a safe and healthy environmental ecosystem [5].

To achieve the goal of a more distributed environmental ecosystem, algorithmic analysis of a large quantity of data [6] will train models that regularly monitor environmental parameters and notify or respond to anomalies in real-time. For the improvement of environmental monitoring, it is crucial to develop efficacious algorithms and models at a continued pace within the environmental community; fostering confidence in these methods requires publicly validated and verifiable processes. Within this system, devices can receive and submit data in a federated manner, with over-the-air updates as the shared algorithms, and models are enhanced with time.

The data used to train models must be tamper-proof and dependable, and the technique must be secure. This might help acquire the confidence of environmentalists and environmental officials, as well as make data collection for investigations easier. Environmentalists will have to look beyond their present methods to achieve this verified future. The rise of distributed ledger technology has the potential to close this gap. Therefore, we need to evaluate and analyze the performance of distributed ledger systems such as IOTA to securely exchange environmental data [7] and establish trust among environmental professionals.

1.2. Distributed Ledger Technologies

Distributed databases or distributed ledgers, such as blockchain [8], are managed via a consensus process by nodes in a peer-to-peer network. Despite the fact that all peers participate in maintaining database integrity, this consensus approach eliminates the necessity for a central administrator. Individuals may reclaim control over their data due to the lack of a central controller.

Blockchain was introduced in 2008 [9] as a distributed ledger technology that is decentralized and immutable. These attributes ensure that the data stored on the blockchain is secure, authentic, and distributed among all the peers in the network. There is no third party involved while making transactions on this technology and no central authority that can control it. These features open the door to various application domains and research areas such as IoT, healthcare, environment monitoring, AI, deep learning, security, and IoT data integrity, wherein the data needs to be distributed and tamper-resistant [10] to avoid a single point of failure when stored on a centralized database.

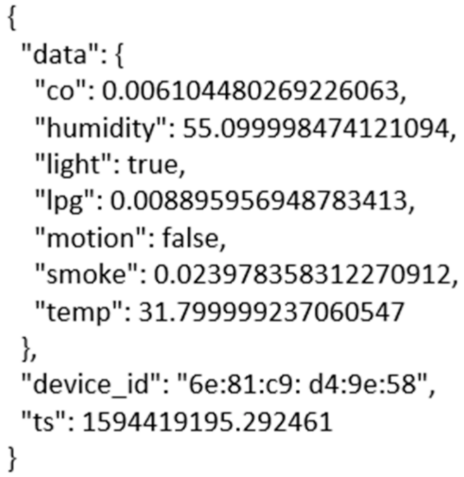
However, blockchain technology is facing several technical challenges despite having a great potential for the construction of future Internet systems [11]. The main challenge or concern is the scalability of blockchain. The time taken to mine a block is about 10 min and the block size is limited to 1 MB only. Moreover, the bitcoin blockchain is not able to deal with high-frequency trading since it is limited to 7 transactions per second. Additionally, the propagation of the blocks will be slow [12] if the block size is large, as it will require more storage space. Since few users would wish to maintain such a large blockchain, this will lead to centralization. Hence, it has been a tough challenge to address the tradeoff between block size and latency. Moreover, there is the possibility of selfish mining strategies, whereby miners can access greater rewards than they are entitled to.

In this paper, we have leveraged environmental sensory telemetry data, which consists of various sensory data parameters such as temperature, humidity, CO, liquid petroleum (LPG), smoke, light, and motion. The data was generated by a series of three customized sensor arrays. Sensor arrays, consisting of an MQ135 hazardous gas detection sensor, DHT22 temperature and humidity sensor, Onyehn IR pyroelectric infrared PIR motion sensor detector, and Anmbest light intensity detection photosensitive sensor were connected to Raspberry Pi sensory devices. Moreover, these devices were placed in distinct physical locations and variable environmental conditions, as shown in Table 1.

Table 1. Data description.

Column	Description	Units
Ts	Timestamp of event	Epoch
Device	Unique device name	String
CO	Carbon Monoxide	ppm (%)
Humidity	Humidity	Percentage
Light	Light detected?	Boolean
LPG	Liquid Petroleum Gas	ppm (%)
Motion	Motion detected?	Boolean
Smoke	Smoke	ppm (%)
Temp	Temperature	Fahrenheit

Each of these IoT devices is continuously collecting the sensory values from four sensors at a standard interval of 5 s. The data was collected during the span of "from 07/12/2020 00:00:00 UTC–07/19/2020 23:59:59 UTC" with a total number of "405,184 rows". In this framework, the "ISO standard Message Queuing Telemetry Transport (MQTT)" protocol [13] is leveraged to bind the sensory readings, a unique ID, and a timestamp and broadcast in terms of a single message; a sample payload is shown in Figure 1.



```
{
  "data": {
    "co": 0.006104480269226063,
    "humidity": 55.099998474121094,
    "light": true,
    "lpg": 0.008895956948783413,
    "motion": false,
    "smoke": 0.023978358312270912,
    "temp": 31.799999237060547
  },
  "device_id": "6e:81:c9: d4:9e:58",
  "ts": 1594419195.292461
}
```

Figure 1. Sample JSON payload.

Different consensus protocols and network topologies have been investigated; these are distributed to ensure the integrity of a distributed ledger while providing high transactions per second and zero fees for transactions. Algorand [14], IOTA [15], Hashgraph [16], and Ouroboros [17] are a few prominent protocols that promise to accomplish the aforementioned characteristics. This technology is suitable not just for the future of electronic finance but also for every data-driven industry.

In this paper, we propose an environmental monitoring [18] application of IOTA, which will allow environmental professionals, such as environmentalists and environmental officers, to share and store encrypted IoT sensor data in a secure way for monitoring purposes. IOTA is a permissionless distributed ledger protocol with no transaction fees. Its goal is to address the scalability concerns that have plagued previous distributed ledger technologies. Moreover, we have leveraged the "Masked Authenticated Messaging extension module of the IOTA protocol" in the proposed approach for the secure transmission, storage, and retrieval of encrypted environmental sensor data. The proposed approach is compared with another method without any data encryption protocol and the performance is measured in terms of time taken in the creation, attachment, and retrieval of payloads.

In summary, the contribution of this paper is as follows:

1. Design of and work using an environmental monitoring application that uses a DAG-based blockchain called IOTA to ensure the security and integrity of environmental IoT data.
2. Propose an architecture that uses IOTA nodes to implement an environmental monitoring application.
3. Implementation of a working model with its architectural design and an extensive evaluation of the model's performance with experiments.
4. Performance evaluation and comparison of the MAM protocol against a non-protocol approach with clear results.

The remainder of the paper is structured as follows. Section 2 illustrates the background technologies for the proposed framework. Section 3 contains the related literature review for choosing this technology and the proposed work. Section 4 highlights the detailed framework of the proposed model. Section 5 showcases the experimental setup, results, and discussion. Finally, Section 6 focuses on a conclusion and future directions in terms of optimizing the MAM protocol for securing sensory data.

2. Background

This section describes the various technologies utilized for the proposed work in this research.

2.1. IOTA

The IOTA is a distributed ledger technology to manage secure data transmission between different IoT devices. The main difference between the IOTA and other distributed ledger technologies is that it utilizes the directed acrylic graph (DAG) structure called the "Tangle" in place of the conventional blockchain. IOTA is highly scalable [19] since there are no blocks in its DAG structure, which leads to a faster confirmation of transactions, unlike in the case of blockchain. Making a transaction on IOTA consumes less energy [20] as compared to other distributed ledgers and, hence, the adoption of IOTA in low power devices such as the IoT becomes rudimentary.

The scalable architecture of the IOTA Tangle enables faster transaction confirmation, as shown in Figure 2. In Figure 2, each square represents a transaction and the arrows also known as edges connect these transactions to form a Tangle. There are three types of transactions, called tips, ongoing transactions, and approved transactions, as shown in Figure 2. Tips are the unconfirmed transactions that are new and have just been added to the ledger. Ongoing transactions are the transactions that have been added to the ledger and are waiting to be referenced by new transactions [21] to achieve confirmation. Approved transactions are the transactions that have been confirmed or have been referenced by all the tips, either directly or indirectly.

The working model and the security of the IOTA protocols were designed with quantum computers in mind, as well as environments with constraints on bandwidth. The Winternitz one-time signature system, which protects against quantum computer access, is used in the IOTA protocol. This one-time signature approach enables effective broadcast authentication in sensor networks since the communication and computing needs are low. As there is no transaction fee for publishing a transaction to IOTA, it can be seamlessly used to send transactions, store data, and ensure data integrity with time. A data transmission protocol [22] called masked authenticated messaging (MAM) enables a user to publish streams of encrypted data in the form of transactions. Participants can broadcast a message at any time by forming a channel [23]. Subscribers can subscribe to the channel of the publisher to receive the data by using the address of the transactions. However, a small amount of proof-of-work is necessary for the data to circulate through the network and prevent spamming. MAM allows the user to send encrypted data streams that are a chain of messages or sensor data to IOTA with zero cost per transaction through the Tangle.

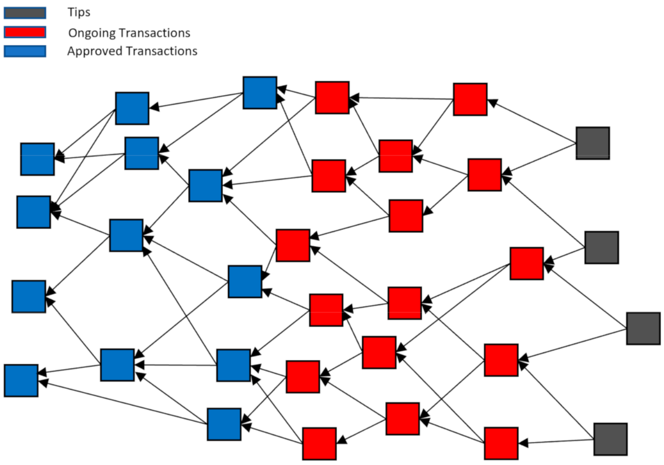
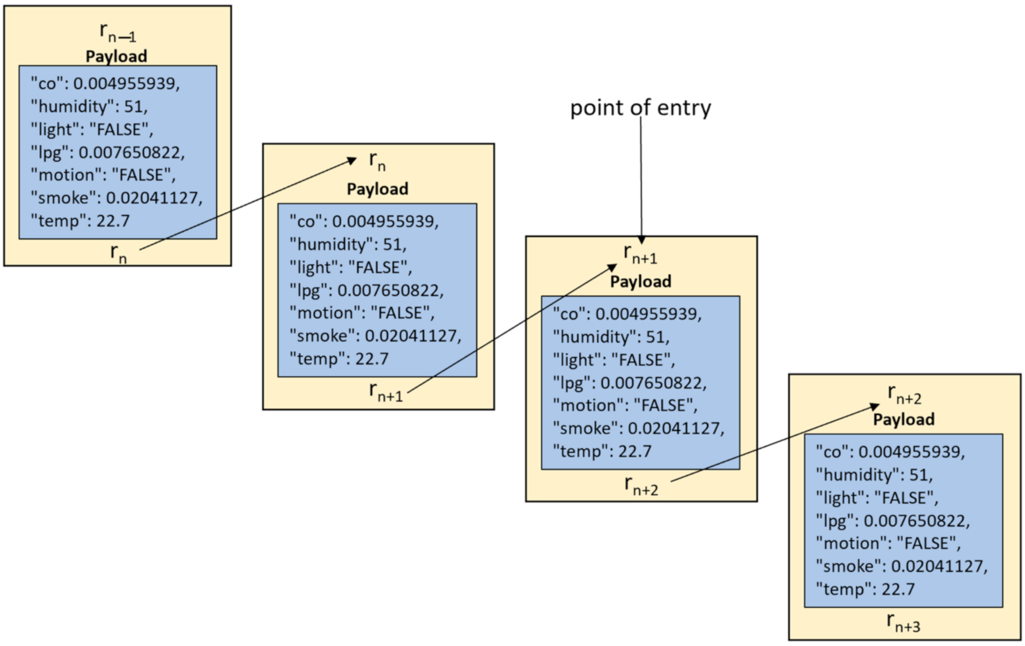


Figure 2. The DAG structure of the Tangle.

Forward secrecy and quantum-resistant cryptography are the two most important features of MAM implementation. Attacks by a quantum machine [24] that is adequately powerful can be resisted due to the secure post-quantum cryptographic algorithms. Many cryptographic algorithms traversing today over the Internet that are presently used to encrypt data [25] are not sufficiently secure. MAM is a useful protocol to transmit confidential data, due to the feature of forward secrecy. Every transaction is linked to the next transaction with a pointer known as next root, which is a Merkle root of the next transaction. As a result, the transaction at the point of entry and the subsequent transactions linked to it can be retrieved efficiently. However, it becomes infeasible for a user to fetch transactions before their point of entry due to forward transaction linking, as shown in Figure 3.



```
"co": 0.004955939,
"humidity": 51,
"light": "FALSE",
"lpg": 0.007650822,
"motion": "FALSE",
"smoke": 0.02041127,
"temp": 22.7
```

```
"co": 0.004955939,
"humidity": 51,
"light": "FALSE",
"lpg": 0.007650822,
"motion": "FALSE",
"smoke": 0.02041127,
"temp": 22.7
```

```
"co": 0.004955939,
"humidity": 51,
"light": "FALSE",
"lpg": 0.007650822,
"motion": "FALSE",
"smoke": 0.02041127,
"temp": 22.7
```

```
"co": 0.004955939,
"humidity": 51,
"light": "FALSE",
"lpg": 0.007650822,
"motion": "FALSE",
"smoke": 0.02041127,
"temp": 22.7
```

Figure 3. Transaction linking in MAM, displaying forward secrecy.

2.2. Modes and Channels of MAM

A channel is first established, then the publisher is able to encrypt data with the channel key and publish them into the Tangle. Clients can fetch the transaction from the Tangle and decode the message on it only if they know the MAM channel key. Messages are connected in chronological order and are published on the same channel. If the users gain access to a channel, they cannot view past transactions on that channel before their entry; this provides the notion of forward secrecy [26]. There are three modes of privacy provided by MAM, known as public, private, and restricted, that control visibility and access to the channels. The address of the transaction is the channel ID of MAM in each mode and allows the system to return a MAM transaction [27] by performing a straightforward request to the Tangle. In contrast, to decode the payload, the key provided in the transaction of MAM does not have to be the same as the channel ID. When the current payload is decoded, the user receives the message as well as the channel key for the subsequent message. For both private and public modes, this property becomes useful, as we will see below.

The channel key is the channel ID that makes up the transaction address for the public mode. Thus, all the contents of the message chain can be read by any user on the network. Due to the additional degree of protection, unauthorized users cannot read a message chain in private mode. The channel key is hashed [28], which becomes the channel ID as well as the transaction address. As a result, the channel key must be safely broadcasted to all subscribed users by the publisher in order that the message can be located on the Tangle network.

The next step involves the subscribed users obtaining the channel key's hash by querying the Tangle, using that key to decode the data payload. If an adversary intercepts a transaction of MAM sent in private mode, they will not be able to read or decode the content of the message payload by utilizing the channel ID, since it was produced by hashing the channel key.

Figures 4–6 represent the different channel modes of MAM and how transactions are linked. The root shown in the three figures is also known as the channel key; the address of the transaction is also known as the channel ID. In all three modes, as shown in the figures, each transaction contains a root and next root. The next root of the current transaction becomes the root for the next transaction, as shown. For the public mode, as shown in Figure 4, the transaction address or the channel ID is the same as that of the next root. For the private mode, as shown in Figure 5, the transaction address is the hash of the next root. However, in the restricted mode, as shown in Figure 6, there is an additional key, known as an authorization key, that is used for performing access control on the data. The transaction address for this mode is the hash of the next root, concatenated with the authorization key.

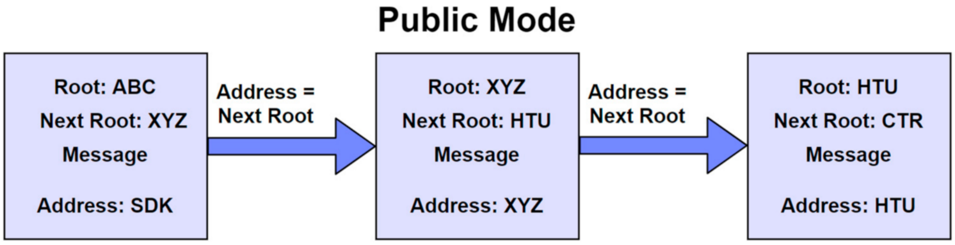


Figure 4. The flow of data in public mode.

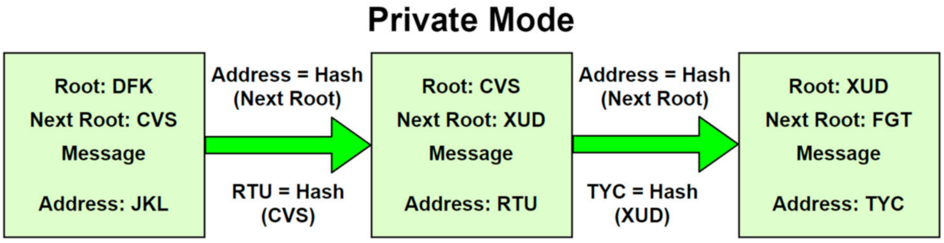


Figure 5. The flow of data in private mode.

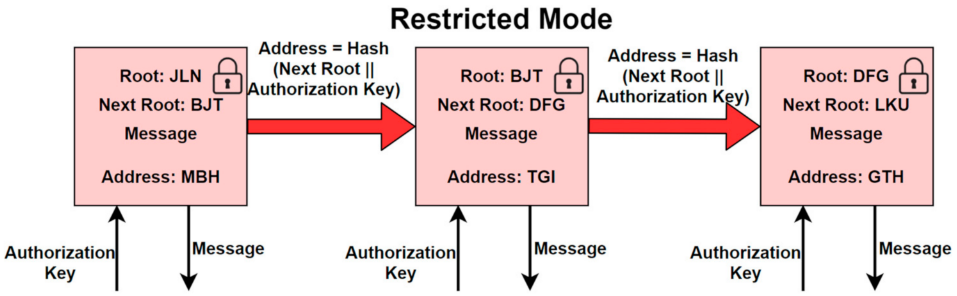


Figure 6. The flow of data in restricted mode.

Hashing of the authorization key is performed, and the hash is concatenated with the channel key to produce the transaction address of the restricted mode in MAM. The authorization key and the channel key are both necessary for decoding the data payload. The publisher specifies the authorization key and can change it at any time in the channel's stream. This enables the publisher to revoke access [29] in their channel from future messages at any point in time. If access is not granted to the subscriber for the current authorization key, they will be unable to decode and locate subsequent transactions in the chain of messages. As a result, subscribers' access can be revoked at any time using this approach. Figure 7 depicts a simplified representation of the many components that go into building an MAM channel.

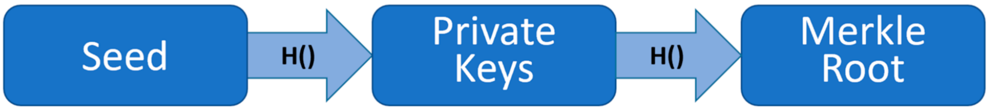


Figure 7. Generation of channel key, using one-way hash functions.

3. Related Work

This section provides a detailed review of the literature in which blockchain technology and IOTA have been adopted in the domain of environmental monitoring and other IoT applications.

Bhandary et al. [30] present the use of a DAG-based blockchain structure called IOTA for the secure sharing of sensor data by integrating two technologies. The paper describes the work and the features of IOTA that can enable seamlessly integrating IoT devices with IOTA to safely transmit IoT data into the Tangle. An architecture was presented in the paper that included the use of Raspberry Pi devices to aggregate and send sensor data to the IOTA network. However, the architecture proposed in the paper was highly generic and lacked a working methodology. Moreover, there was no experimental evaluation of the architecture, especially in terms of performance.

Yu et al. [31] analyzed the stereotypical privacy and security issues in IoT and developed a framework that utilized Ethereum blockchain with an IoT system. A four-layered architecture was proposed where blockchain was used at the database layer to adapt to the IoT system. A good theoretical description of how the proposed framework tackles IoT security and privacy issues were provided. However, a proper working model to practically address these IoT security and privacy issues was missing. Furthermore, there was no performance or latency evaluation for the proposed framework.

Lamtzidis et al. [32] proposed a sensor node system that was distributed and utilized the IOTA distributed ledger to exchange data with IoT devices. In this paper, a distributed wireless sensor node system was proposed that ensured integrity of the data across the entire pipeline. The proposed system consisted of three entities: super nodes (SNs) that aggregated the data, full nodes (FNs), which are the IOTA nodes that perform the proof-of-work (PoW), and a back-end server. However, their proposed method did not show how the three entities are connected and how the data is flowing. Additionally, there was no implemented architecture and an experimental evaluation of their proposed system.

Yan et al. [33] proposed an environmental monitoring system that uses blockchain to provide integrity to the environmental data and prevent falsification. Additionally, a three-dimensional architecture using intelligent trusted devices was presented for environmental monitoring, to ensure the integrity and originality of the data collected by the IoT devices. The raw data from the sensors were transmitted to a whole node that could send data to the blockchain and could synchronize all the data within the blockchain nodes. However, an extensive performance evaluation of the proposed model was missing, which would provide some details on the latency of blockchain operations. Moreover, the traditional blockchain setup cannot meet the scalability demands of the IoT system; hence, a scalable blockchain or distributed ledger technology is required.

Shabandri et al. [34] presented an approach using the IOTA distributed ledger technology and IoT devices to demonstrate two IoT applications on the Tangle, such as a "smart utility meter system" and a "smart car transaction system". These proposed applications were connected to the internet using low power wide area networks (LPWAN). A DAG-based blockchain IOTA was used by the researchers to overcome the scalability and transactional cost of the conventional blockchain. Although the research paper gave detailed steps to implement the proposed applications, a well-defined architecture was missing and only a proof-of-concept (PoC) was presented.

Benedict et al. [35] proposed an implementation in the cloud that uses IoT-enabled blockchain to address some existing issues in smart cities. The research focuses on the use of "chaincodes", which are also known as smart contracts, for monitoring air quality systems in smart cities. An architecture called an "IoT-enabled blockchain for an air quality monitoring system (IB-AQMS)" was proposed and an experiment to assess the model was performed. However, the "chaincode" execution time for their approach was too high and would not satisfy the current IoT demands for a scalable system.

Guanochanga et al. [36] developed a wireless sensor network that monitored several air quality parameters within smart cities. An experiment was conducted on their proposed system and excellent results were obtained in the preliminary analysis. The preliminary results showed that the proposed approach could be used as a cost-effective tool for monitoring air quality. However, the approach lacked a framework or an entity that could ensure the integrity and security of the air quality data.

Mahmoud et al. [37] presented a review on the security of the IoT, various requirements for security, and proposed different countermeasures to secure IoT devices. A detailed description of the security issues that must be addressed at each layer of the IoT architecture was explained.

Bures et al. [38] provide a comprehensive review of the various features of IoT and the security challenges specifically related to IoT. The paper covered a vast number of security features and challenges that must be addressed to secure IoT devices and emphasized that security and privacy are the major security challenges that must be addressed to achieve a secure IoT system.

Our proposed architecture, which uses the IOTA nodes, overcomes the above-mentioned limitations. The proposed model satisfies the major security requirements for IoT, which include data confidentiality, integrity, and security at the application layer of the IoT stack or where the end-user requires the data. This provides a secure working environment for monitoring environmental IoT data generated from various IoT devices. Furthermore, in this paper, we conduct an extensive experimental evaluation of our proposed model to access its performance.

4. Proposed Architecture

The proposed work in this research paper aims to provide an environmental monitoring application by using the IOTA distributed ledger and the masked authenticated messaging (MAM) protocol. This application aims to ensure the security and privacy of the sensor data, as well as to control and prevent various environmental issues and hazards such as air pollution and greenhouse emissions. Moreover, this paper measures, analyzes, and compares the performance of the capability of MAM protocol using a non-protocol method.

4.1. Publishing and Fetching Environmental Sensor Data

We set out to evaluate MAM's potential for publishing environmental sensor data since it is a lightweight data communication protocol over an immutable distributed ledger. Using an MAM protocol, a system that could publish and fetch the environmental sensor data was developed. We installed the MAM Client JavaScript Wrapper library [39], as well as preparing the data payloads to be published to the private Tangle using MAM, and structured the data, utilizing the JSON format in the Windows client.

The Windows client was configured to publish the MAM data payloads through a restricted channel where a channel key and authorization key are used by the data publisher, i.e., an environmentalist, to encrypt the MAM data payloads. At the transaction level, an environmentalist can define the access controls. If an environmentalist wants to give one or more environmental officers access to their channels, they can send their channel keys to them. In return, the environmental officer could retrieve and authenticate the corresponding data payloads from the Tangle. If an environmentalist would like to revoke access to their stream of data at any time, this just requires updating their MAM channel's authorization key and safely transmitting it to a desired environmental officer.

With this architecture in place, as shown in Figure 8, the client device automatically published environmental sensor data to the private Tangle using the MAM Client JavaScript Wrapper library. Using MAM's restricted channel mode, data payloads were attached. We were able to examine how an environmentalist could change controls for accessing a specific stream of messages by upgrading their authorization key. To acquire the data payloads once the transactions were published to the Tangle, we used an authorization key and channel key.

We evaluated and compared the performance of our MAM implementation with a non-protocol-based approach, to further assess MAM's capability and applicability for this functionality. We published payloads, sized 145, 330, 515, and 740 KB, in the restricted channel configuration by utilizing the MAM client JavaScript Wrapper library for Node.js on the Intel(R) Core^{TM} i7-8565U processor of the Windows client device. We chose these sizes of payloads due to the limitations of the MAM protocol, which can handle a maximum size of 740 KB. Keeping the sensor data payloads, the processor, and the client device the same, we published the data payloads with a non-protocol approach [40] to the private Tangle, using Python and Jupyter Notebook as the runtime environment. Furthermore, we analyzed and compared the results of the two approaches.

4.2. Hashing of Merkle Tree

Hashed trees are generated using the Merkle hashing technique, where the direction of the trees goes upward. This tree is called the Merkle hash tree (MHT), wherein the leaves of the tree represent the hash of the values of the data or the ordered elements of a set. Let this authentic ordered set of elements for MHT be $x_{0,0}$, $x_{0,1}$, $x_{0,2}$, . . . , $x_{0,\mathrm{n}}$; therefore, the leaf node of the element $x_{0,i}$ will be the hash of that element. Let this leaf node be represented by $x_{1,i}$, where $x_{1,i}$ = $H(x_{0,i})$ and $H()$ is a function that is cryptographically hashed one way.

A node in the MHT contains multiple incoming edges; the value of a node is the combined or concatenated hash [41] of its preceding nodes, also known as child nodes, where the sequence of the nodes is maintained. An internal node or a non-leaf node $x_{2,0}$ with child nodes $x_{1,0}$ and $x_{1,1}$ hence contains the value $x_{2,0} = H(x_{1,0}||x_{1,1})$. The MHT and a verification object that contains a set of nodes can be used to demonstrate the existence of an element. The root of the MHT [42] can be recomputed by the verifier by using the verification object and a set of nodes that are contained within it. The verifier compares the recomputed root using the verification object, with the publicly known root that the tree generates. For instance, consider the element $x_{0,0}$ in the MHT shown in Figure 9; the verification object consists of the values of the nodes $x_{0,0}, x_{1,1}$, and $x_{2,1}$. $x_{1,0} = H(x_{0,0})$, $x_{2,0} = H(x_{1,0}||x_{1,1})$ and conclusively, $root = H(y_{2,0}||x_{2,1})$ is constructed by the verifier. Once this verification object is constructed, the verifier can compare the computed root with the publicly known root and verify the value.

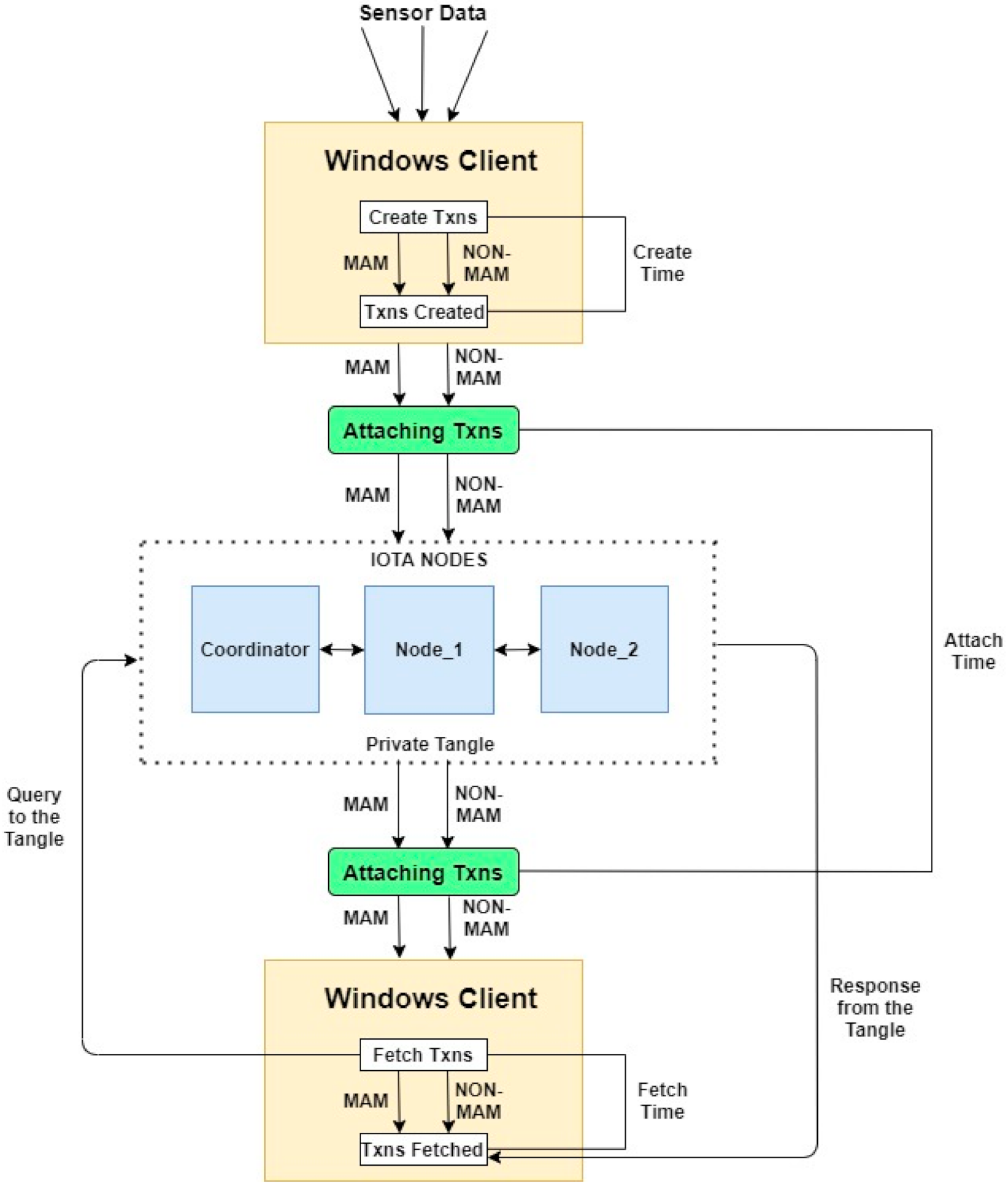


Figure 8. System architecture.

4.3. Hashing, Merkle Tree Signature Scheme, and One-Time Signatures

A digital signature technique is also known as the one-time signature (OTS) scheme can only be used to do a signature on one message with one key pair. Faster signing and algorithms for verification can be achieved with different techniques using hash-based OTS when compared to schemes such as RSA [43], which is a public-key digital signature technique. However, there are significant restrictions to OTS approaches, such as the length of signatures, the size of keys, and the maximum number of signatures possible.

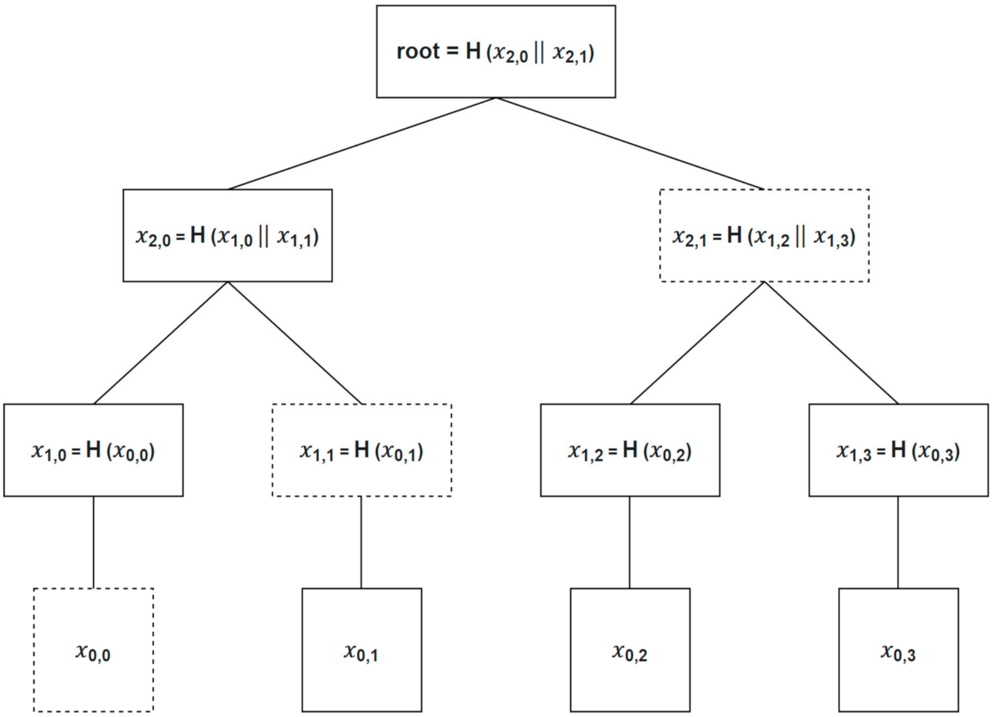

root = H ($x_{2,0}$ || $x_{2,1}$)
$x_{2,0}$ = H ($x_{1,0}$ || $x_{1,1}$)
$x_{2,1}$ = H ($x_{1,2}$ || $x_{1,3}$)
$x_{1,0}$ = H ($x_{0,0}$)
$x_{1,1}$ = H ($x_{0,1}$)
$x_{1,2}$ = H ($x_{0,2}$)
$x_{1,3}$ = H ($x_{0,3}$)
$x_{0,0}$
$x_{0,1}$
$x_{0,2}$
$x_{0,3}$

Figure 9. A binary Merkle hash tree built for the authentic values $x_{0,0}$, $x_{0,1,}$ $x_{0,2}$, $x_{0,3}$. The values of nodes required to verify $x_{0,0}$ are bounded with broken lines.

Cryptographic secure hash functions ensure the security of OTS. The properties of a cryptographic secure hash function will be defined in this section. There are three categories, namely, "preimage-resistant", "second preimage-resistant" and "collision-resistant", and a hash function H: {0,1}* tends to {0,1}^{s} that is cryptographically secure if it falls in the above three categories.

- Preimage-resistant.

For a hash function H, if it is hard to find any m for a given h with $h = H(m)$, then it is preimage-resistant.

- Second preimage-resistant.

For a hash function H, if it is hard to find any m_2 for a given m_1 with $H(m_1) = H(m_2)$, then it is second preimage-resistant.

- Collision-resistant.

For a hash function H, if it is hard to find a pair of m_1 and m_2 with $H(m_1) = H(m_2)$, then it is collision-resistant.

This multiple OTS can be verified by using a single public verification key, which is possible due to the MHT-based Merkle signature scheme (MSS). Each OTS scheme is represented by one leaf of the MHT. This implies that the same number of messages can be produced by each tree as the leaves of the MHT. The OTS technique's public verification keys [44] will be used to verify all of these communications. The OTS scheme's public verification keys are validated by computing the MHT's root from a specified verification object, as illustrated in Figure 7.

Only a limited number of messages can be signed with one public key, "pub_key", by using MSS. Let NUM = 2^{n} be the total possible number of messages since they must be a power of two. To generate the public key, "pub_key", the first step is to generate the private keys X_i and public keys Y_i of 2^{n} one-time signatures. For each public key Y_i, a hash value $H(Y_i)$ is calculated, where $1 \leq i \leq 2^n$. An MHT is constructed with these hash values,

h_i, $2^{n+1} - 1$ nodes, and 2^n leaves. In the MSS, the public key "pub_key" is the root of the Merkle tree.

Consider a message, M, which is to be signed with MSS [45]; first, a signature S results due to the signing, using a one-time signature technique on the message M. To execute the signature, S, one of the public and private key pairs $(X_i,\ Y_i)$ is used [46]. Let the path from a given leaf to the root be denoted by P. The total number of nodes that path P contains is $n + 1$, with paths $P_1,\ \ldots,\ P_{n+1}$, where $P_1 = y_{1,\ i}$ is the leaf and $P_n = y_{n+1,\ 0} = pub\_key$ is the root of the MHT. We require every child of the nodes $P_2,\ \ldots,\ P_{n+1}$ to compute P. As it is known that P_i is a child of P_{i+1}, therefore, to calculate the next node P_{i+1} of the path P, both the children of P_{i+1} must be known. To solve this computation, we require the sibling node of P_i. Let s_i be the sibling, such that $P_{i+1} = H(P_i||s_i)$ in the case where s_i is odd; if it is even, then $P_{i+1} = H(s_i||P_i)$. Therefore, n nodes, $s_0,\ \ldots,\ s_{n-1}$ are required to compute each node present in P. The signature of the MSS $sig = (S||s_2||s_3||s_{n-1})$ comprises the one-time signature S of the message M, plus the nodes.

The recipient knows the signature, $sig = (S||s_2||s_3||s_{n-1})$, the message M, and the public key pub_key. Firstly, the one-time signature S of message M is verified by the recipient. $P_1 = H(Y_i)$ is computed by the recipient, who hashes the public key of the one-time signature. For $k = 1,\ \ldots,\ n-1$, the nodes of P_k of path P are calculated with $P_k = H(y_{k-1}||s_{k-1})$ if the sibling index is odd, $P_k = H(s_{k-1}||y_{k-1})$ if it is even. The signature is valid if $P_n = pub\_key$ of the MSS.

5. Experimental Results and Analysis

An experiment was successfully performed that proved the feasibility of the proposed system to publish and retrieve authenticated, encrypted environmental IoT sensor data by using a distributed ledger. The MAM protocol ensured the source's validity and the data's integrity, which were formatted in the JSON format. We also demonstrated how an environmentalist might change the authentication keys to restrict permission to the data they may publish in the future. As a result, we demonstrated the potential of granular access controls, defined by the environmentalist.

A private Tangle was created, which consisted of three full nodes, called the coordinator, and two neighbor nodes, namely, "Neighbor Node_1" and "Neighbor Node_2" to test our proposed work, as shown in Figure 8. All three nodes were set up on three Linux Ubuntu servers with Hornet installed on them. Hornet is a powerful, community-driven IOTA node software written in the Go language and is a lightweight alternative to the IOTA reference implementation (IRI). Hornet was developed for the secure transfer of tokens or data, and for experimenting and implementing IOTA protocols between nodes or network participants. Machines can act as a node and connect to the IOTA network with the help of the Hornet software. These nodes or machines have functions such as authenticating the transactions, storing these authenticated transactions on the Tangle, and fetching these transactions back from the Tangle whenever required.

The dataset used was an open-source dataset that contained the environmental sensor data in a JSON format. The data included environmental parameters, such as temperature, timestamp, unique device id, carbon monoxide level, humidity percentage, light detected, liquified petroleum gas content, motion detected, and smoke levels. The payloads were created, and three actions (namely, create, attach, and fetch) were performed and analyzed in 300 trials. The Windows client machine, also known as the IOTA client, published and retrieved sensor data in the form of transactions to the Tangle. The client machine connects to the private Tangle using IOTA API and can make various API calls to perform various tasks. The experiment was performed with two approaches—(1) using the MAM protocol and (2) a non-protocol-based approach.

An experimental assessment was performed to evaluate the scalability of the two approaches. To achieve this, we focused on the three major tasks (i.e., create, attach, and fetch) that occur when publishing and fetching transactions. For the "create" task, we calculated the time it takes to create the transactions before publishing them to the IOTA nodes. The IOTA API was used to create the transaction object from the data payload and the execution time for this task was measured, which is called “create time”.

The next step was to execute the “attach” task and calculate its execution time. Once the transaction object is created, it is published to the IOTA network by conducting the PoW and storing the transactions that the IOTA nodes perform. We calculated the execution time for this and labeled it as “attach time”.

The final step was to execute the “fetch task” and calculate its execution time. After the transactions are published and stored in the IOTA network, we fetched these transactions by performing a query to the private Tangle, which in response provides the transactional data. We calculated the execution time for this fetch task and labeled it as “fetch time”.

The three tasks can be mathematically expressed for the two approaches in the following way:

$$\begin{aligned} MAM_c &= E(data) + ch\_gen \\ MAM_a &= PoW(MAM_c) + stor(MAM_c) \\ MAM_f &= D(que(MAM_a) + response) \end{aligned}$$

$$\begin{aligned} NON\text{-}MAM_c &= Enc(data) \\ NON\text{-}MAM_a &= PoW(NON\text{-}MAM_c) + stor(NON\text{-}MAM_c) \\ NON\text{-}MAM_c &= que(NON\text{-}MAM_c) + response \end{aligned}$$

where E represents encryption; ch_gen represents channel generation; $stor$ represents storing; Enc represents encoding; and que represents a query.

5.1. MAM

The proposed work was performed using Node.js, an open-source cross-platform runtime environment for web application development [47]. Node.js apps are written in JavaScript and operate on a variety of platforms. MAM is an IOTA protocol that ensures only the authenticated parties are sending messages that are encrypted, ensuring both confidentiality and security. We used the restricted channel mode of MAM, which encrypts the data using the channel key and authorization key; only those parties having the correct keys can access the data from the IOTA Tangle.

With this system in place, the IOTA client created, attached, and fetched the sensor data payloads by executing the JavaScript code in Node.js, using the MAM Client JavaScript Wrapper library. After the transaction was published to the Tangle, the sensor data was fetched by using the channel key along with the authorization key. The results of this approach are displayed in Table 2.

Table 2. Results of the MAM experiment.

Processor	Action	Payload Size (KB)	Trials	Avg. Time (s)	St. Dev. (s)	Variance (s^2)	Min (s)	Max (s)
Intel(R) Core^{TM} i7-8565U	Create	145	300	0.49	0.13	0.017	0.415	1.34
Intel(R) Core^{TM} i7-8565U	Create	330	300	0.7	0.064	0.004	0.661	1.31
Intel(R) Core^{TM} i7-8565U	Create	515	300	0.95	0.026	0.001	0.91	1.07
Intel(R) Core^{TM} i7-8565U	Create	740	300	1.39	0.42	0.17	1.21	3.53
Intel(R) Core^{TM} i7-8565U	Attach	145	300	20.24	1.99	3.996	18.186	35.873
Intel(R) Core^{TM} i7-8565U	Attach	330	300	50.26	4.06	16.53	44.97	75.64
Intel(R) Core^{TM} i7-8565U	Attach	515	300	77.22	7.79	60.77	69.47	132.75
Intel(R) Core^{TM} i7-8565U	Attach	740	300	115.01	12.85	165.17	99.17	180.45
Intel(R) Core^{TM} i7-8565U	Fetch	145	300	0.426	0.032	0.001	0.361	0.697
Intel(R) Core^{TM} i7-8565U	Fetch	330	300	1.82	0.105	0.011	1.7	2.7
Intel(R) Core^{TM} i7-8565U	Fetch	515	300	1.92	0.138	0.019	1.79	2.94
Intel(R) Core^{TM} i7-8565U	Fetch	740	300	2.25	0.148	0.022	2.05	3.11

5.2. Non-Protocol Method

Using this approach, we published and retrieved the data payloads from the Tangle by only publishing the sensor data in the form of zero-value transactions [48], which are transactions that only contain data and no cryptocurrency, and without using any IOTA protocol. The proposed work was performed using Jupyter Notebook [49] which is an open-source web tool for creating and sharing documents with live code, equations, visualizations, and machine learning. Two Python scripts were written, where one script was configured to publish the sensor data, i.e., creating and attaching the transactions to the Tangle, while the other one was used to fetch the data from the Tangle. These two scripts utilized the official Python library for IOTA, called Pyota. Jupyter Notebook, running on the IOTA client, executed the two scripts to perform the abovementioned tasks.

With this system in place, the IOTA client published and fetched the sensor data from the Tangle with the help of the address whence the transactions were sent. The results of this approach are shown in Table 3.

We concentrated our investigation on the tasks that caused significant time delays for publishing and retrieving the messages. The three acts were studied: create, attach, and fetch. First, based on our findings, we discovered that the execution time for the message creation task was precise and was dependent on both the processor and the size of the payload. Second, we found that the average time for the attack process had a strong relationship with the size of the payload. Because the attaching stage involves the proof-of-work [50], which was conducted remotely by the private IOTA Tangle, a large variance and high correlation to payload size were expected. Thirdly, the average time to fetch a message from the private Tangle showed a high correlation to the payload size.

The average time was calculated for all three actions i.e., create, attach, and fetch, respectively, and are displayed in Figures 10–12. It can be seen that the MAM protocol performs far better than using any non-protocol method for publishing and retrieving sensor data from the private Tangle.

Table 3. Results of the non-protocol method.

Processor	Action	Payload Size (KB)	Trials	Avg. Time (s)	St. Dev. (s)	Variance (s^2)	Min (s)	Max (s)
Intel(R) Core^{TM} i7-8565U	Create	145	300	15.82	0.35	0.12	15.35	17.25
Intel(R) Core^{TM} i7-8565U	Create	330	300	29.03	0.62	0.38	27.88	31.49
Intel(R) Core^{TM} i7-8565U	Create	515	300	34.91	0.74	0.56	33.75	37.96
Intel(R) Core^{TM} i7-8565U	Create	740	300	91.75	1.73	2.99	88.76	97.07
Intel(R) Core^{TM} i7-8565U	Attach	145	300	23.3	0.11	0.012	22.97	23.59
Intel(R) Core^{TM} i7-8565U	Attach	330	300	52.91	0.11	0.013	52.45	53.2
Intel(R) Core^{TM} i7-8565U	Attach	515	300	82.53	0.18	0.033	81.87	82.95
Intel(R) Core^{TM} i7-8565U	Attach	740	300	118.3	0.68	0.471	116.4	119.6
Intel(R) Core^{TM} i7-8565U	Fetch	145	300	56.51	1.6	2.58	53.77	64.08
Intel(R) Core^{TM} i7-8565U	Fetch	330	300	135.1	38.4	6.2	159.2	127.2
Intel(R) Core^{TM} i7-8565U	Fetch	515	300	223.4	10	101	206.2	254.4
Intel(R) Core^{TM} i7-8565U	Fetch	740	300	327.9	5.99	35.9	318.1	346

5.3. Discussion

Since IoT devices are utilized in a variety of applications, there is a need to ensure data privacy and security, based on the application domain and the type of data being communicated between parties, such that an adversary cannot eavesdrop or tamper with the data.

Due to the IOTA distributed ledger, we achieved a tamper-proof audit trail of environmental sensor data, published from various IoT devices. The MAM extension module of IOTA provides environmentalists with the ability to publish, store and fetch the encrypted, authenticated, on-demand environmental sensor data by using the Tangle. The MAM protocol empowers the environmentalists by providing agency over the collected environmental sensor data, allowing them to share this data with the environmental officers for monitoring purposes. MAM's limited mode gives environmentalists fine-grained access controls over how data is shared across specialists in the digital environmental ecosystem, while the Tangle adds an extra layer of integrity to ensure that data is not tampered with. We discuss the privacy, security, and feasibility of our proposed system in the remainder of this section.

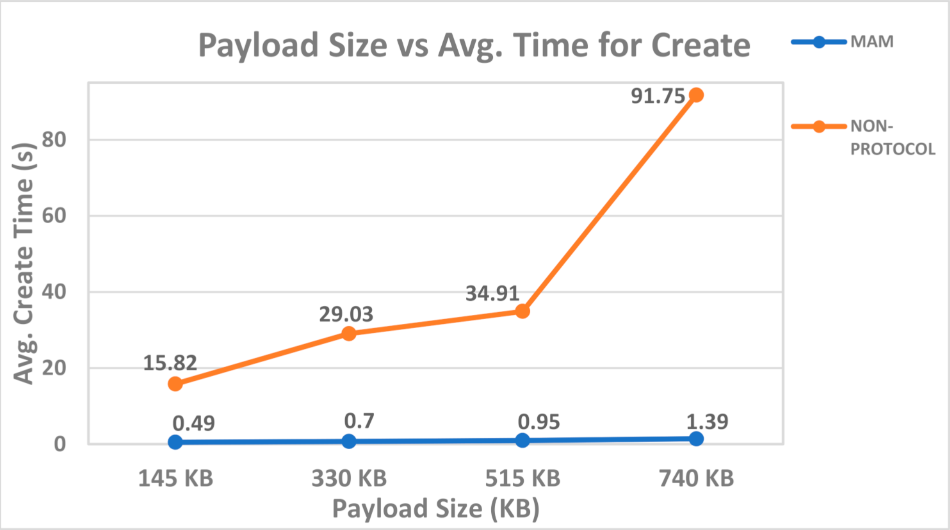


Figure 10. Line graph displaying the average "create time", with respect to payload size, for the MAM protocol and non-protocol methods.

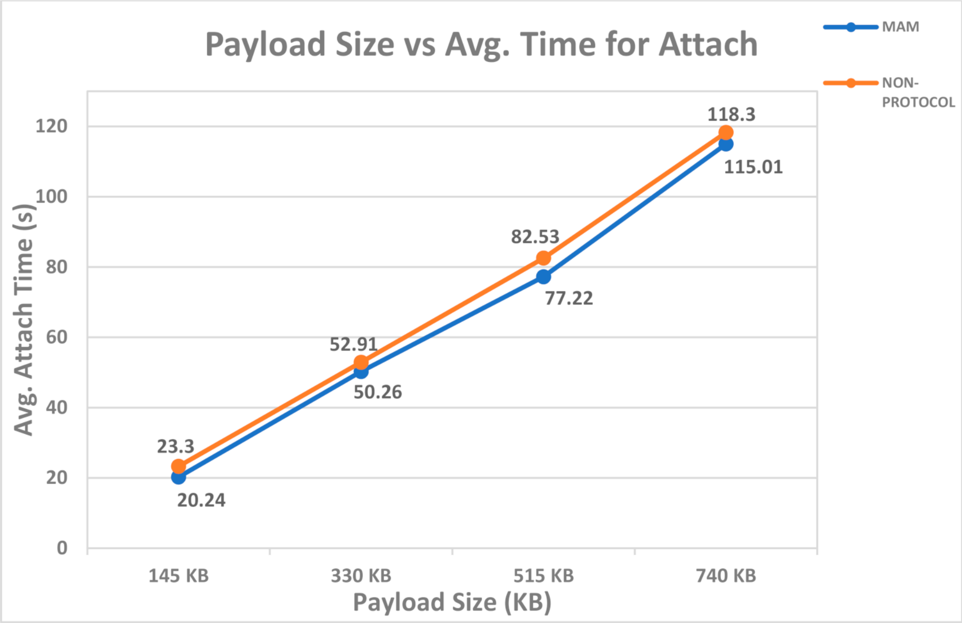


Figure 11. Line graph displaying the average "attach time", with respect to payload size, for the MAM protocol and non-protocol methods.

Since every node in a distributed ledger's network needs a copy of the current state of the ledger, distributed ledger technology seems to go against our present understanding of digital privacy. Despite the fact that value transactions on distributed ledgers may be pseudonymous, monitoring network traffic by analyzing the frequency of transactions and locations of origin could lead to the conclusion that one person has communicated with another regularly. Moreover, finding out the number of tokens an entity possesses is also possible, with varying levels of uncertainty. On a distributed ledger, enhancing privacy while maintaining auditability is still an ongoing area of research. Nevertheless, since MAM eliminates the concept of two entities communicating with each other, this issue does not pose a difficulty for our proposed system. Instead, the issuer generates transaction addresses at random in a data stream, regardless of who has access to the information required to decrypt the data. The following along of a public or private chain of messages can be achieved by the subscribers from their point of entrance forward since the next channel key is incorporated with the current message. Our technique, on the other hand, makes use of MAM's restricted mode, which, as mentioned in Section 2.2, allows an individual to revoke access from previous subscribers by making the addresses of future transactions unknown to them. This can be achieved only if the user changes the authorization key and, hence, access by undesired subscribers is revoked. Data can be kept private inside the transactions, due to the access controls that the MAM channel modes offer and are, thus, contradictory to the feature of transparency in distributed ledgers. If an environmentalist prefers not to have that amount of control over their data, they can generate and store authorization keys themselves, or they can delegate that power to environmental professionals with higher authority. The United States Environmental Protection Agency (US EPA) or another federal agency must store these authorization keys. Environmental officers will be allowed to access data if an environmentalist is unable to recollect them, give a log of their authorization keys, or, if the officer is needed to take immediate action, to manage an environmental threat.

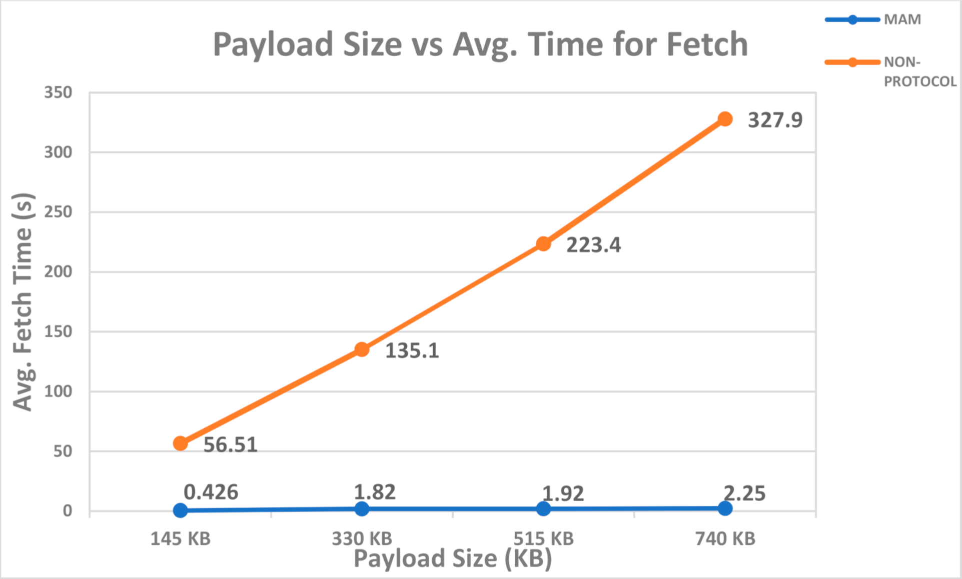


Figure 12. Line graph displaying the average "fetch time", with respect to payload size, for the MAM protocol and non-protocol methods.

Even though we demonstrated how MAM could be used to enable secure environmental sensor data exchange, we built our framework to be flexible enough to accommodate any open environmental data exchange standards. Furthermore, data can be transmitted using MAM from any endpoint with an internet connection, such as an environmentalist's computer, a server at a government institution like the US EPA, a mobile device, or a Bluetooth low-energy sensor. Our proposed system can be effortlessly linked with any professional in the digital environmental ecosystem due to the accessibility of encrypted data through open APIs. This, we believe, will facilitate acceptance, and open the door to new uses that go beyond environmental data collected without the oversight of environmental experts.

6. Conclusions and Future Directions

This study investigated the creation of an on-demand digital environmental ecosystem that relies on algorithms to analyze a huge amount of data, as well as the requirement that this data should be immutable, authenticated, and distributed. Using the MAM protocol, we demonstrated how encrypted environmental sensor data can be broadcasted, stored, and fetched from the IOTA Tangle to prove the data's integrity, security, and privacy. We also showed how granular access controls can be defined and updated by environmentalists. The MAM protocol proved to be a useful tool for encrypting and authenticating sensor data, although it may be improved in terms of performance and design. Many application fields, such as healthcare, the supply chain, and data storage of many kinds of sensor data, could benefit from this way of storing and delivering encrypted sensor data.

Based on the results of our extensive experimental evaluation of the proposed model, we can conclude that the MAM protocol performs better and provides better security than the non-protocol approach. The MAM protocol provided some additional features such as data encryption and granular access control, which provided better security and privacy, compared to the non-protocol method. Therefore, the MAM protocol can be seamlessly linked to various IoT devices to meet the scalability demands of these devices.

For future work, we suggest that the MAM protocol must integrate a secure and efficient key-transmitting method that would exchange the authorization keys between different entities. Additionally, as the MAM protocol develops and matures, we will demonstrate how this protocol can be used to ensure data integrity, by developing a proof-of-concept across academic universities.

Finally, we need to address how a huge dataset can be maintained across different stakeholders since the sensor data is widely distributed and the sensors are producing more data exponentially with time. IoT and embedded devices have the potential to generate massive amounts of data that will be incompatible with complete nodes, which cannot store the entire history of data. The complete nodes will keep track of the current state and prune the remaining data to make room for new transactions. An organization must have a complete record of all relevant transactions; nevertheless, the trimmed transactions will still have provable cryptographic links.

Author Contributions: P.G. created the proposed architecture and experimental scripts, and is the main writer. T.B. supervised the proposed model and is the secondary writer. S.J. contributed toward the writing of background technologies for this research. A.P.-P. and H.U. supervised the conducted research and reviewed the research article. Finally, L.L. provided the funding and the resources to perform this research. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the U.S Department of Energy National Energy Technology Laboratory (DOE NETL), grant number is DE-FE0031745.

Data Availability Statement: The dataset used for this research was publicly available and can be found here [https://www.kaggle.com/garystafford/environmental-sensor-data-132k (accessed on 20 July 2020)].

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

References

1. Mois, G.; Folea, S.; Sanislav, T. Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2056–2064. [CrossRef]
2. Harris, M. Mules on a mountain. *IEEE Spectr.* **2016**, *53*, 50–56. [CrossRef]
3. Zhang, L.; Tian, F. Performance Study of Multilayer Perceptrons in a Low-Cost Electronic Nose. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 1670–1679. [CrossRef]
4. du Plessis, R.; Kumar, A.; Hancke, G.; Silva, B. A wireless syste m for indoor air quality monitoring. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 5409–5414.
5. Mukhopadhyay, S. Research activities on sensing, instrumentation, and measurement: New Zealand perspective. *IEEE Instrum. Meas. Mag.* **2016**, *19*, 32–38. [CrossRef]
6. Lee, M.; Offutt, A.J.; Alexander, R.T. Algorithmic analysis of the impacts of changes to object-oriented software. In Proceedings of the 34th International Conference on Technology of Object-Oriented Languages and Systems—TOOLS 34, Santa Barbara, CA, USA, 4 August 2000; pp. 61–70.
7. Lazarescu, M.T. Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2013**, *3*, 45–54. [CrossRef]
8. Dinh, T.T.A.; Liu, R.; Zhang, M.; Chen, G.; Ooi, B.C.; Wang, J. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1366–1385. [CrossRef]
9. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Bus. Rev.* **2008**, 21260. [CrossRef]
10. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352. [CrossRef]

11. Batubara, F.R.; Ubacht, J.; Janssen, M. Challenges of blockchain technology adoption for e-government. In Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, Delf, The Netherlands, 30 May–1 June 2018; pp. 1–9.
12. Zachariadis, M.; Hileman, G.; Scott, S.V. Governance and control in distributed ledgers: Understanding the challenges facing blockchain technology in financial services. *Inf. Organ.* **2019**, *29*, 105–117. [CrossRef]
13. Yassein, M.B.; Shatnawi, M.Q.; Aljwarneh, S.; Al-Hatmi, R. Internet of Things: Survey and open issues of MQTT protocol. In Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS), Monastir, Tunisia, 8–10 May 2017; pp. 1–6.
14. Gilad, Y.; Hemo, R.; Micali, S.; Vlachos, G.; Zeldovich, N. Algorand. In Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, 28 October 2017; pp. 51–68.
15. Guo, F.; Xiao, X.; Hecker, A.; Dustdar, S. Characterizing IOTA Tangle with Empirical Data. In Proceedings of the GLOBECOM 2020–2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
16. Akhtar, Z. From Blockchain to Hashgraph: Distributed Ledger Technologies in the Wild. In Proceedings of the 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON), Aligarh, India, 8–10 November 2019; pp. 1–6.
17. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In *Advances in Cryptology—CRYPTO 2017*; Springer International Publishing: Cham, Switzerland, 2017; pp. 357–388.
18. Othman, M.F.; Shazali, K. Wireless Sensor Network Applications: A Study in Environment Monitoring System. *Procedia Eng.* **2012**, *41*, 1204–1210. [CrossRef]
19. Silvano, W.F.; Marcelino, R. Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. *Futur. Gener. Comput. Syst.* **2020**, *112*, 307–319. [CrossRef]
20. Silvano, W.F.; De Michele, D.; Trauth, D.; Marcelino, R. IoT sensors integrated with the distributed protocol IOTA/Tangle: Bosch XDK110 use case. In Proceedings of the 2020 X Brazilian Symposium on Computing Systems Engineering (SBESC), Florianopolis, Brazil, 24–27 November 2020; pp. 1–8.
21. Zivi, N.; Kadusic, E.; Kadusic, K. Directed Acyclic Graph as Tangle: An IoT Alternative to Blockchains. In Proceedings of the 2019 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, 26–27 November 2019; pp. 1–3.
22. Bhandary, M.; Parmar, M.; Ambawade, D. Securing Logs of a System—An IoTA Tangle Use Case. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 697–702. [CrossRef]
23. Shafeeq, S.; Alam, M.; Khan, A. Privacy aware decentralized access control system. *Futur. Gener. Comput. Syst.* **2019**, *101*, 420–433. [CrossRef]
24. Ambainis, A.; Rosmanis, A.; Unruh, D. Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding. In Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, Philadelphia, PA, USA, 18–21 October 2014; pp. 474–483.
25. Lee, H.K.; Malkin, T.; Nahum, E. Cryptographic strength of ssl/tls servers. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement—IMC '07, San Diego, CA, USA, 24–26 October 2007; p. 83.
26. Korotkyi, I.; Sachov, S. Hardware Accelerators for IOTA Cryptocurrency. In Proceedings of the 2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 16–18 April 2019; pp. 832–837.
27. Lamtzidis, O.; Pettas, D.; Gialelis, J. A Novel Combination of Distributed Ledger Technologies on Internet of Things: Use Case on Precision Agriculture. *Appl. Syst. Innov.* **2019**, *2*, 30. [CrossRef]
28. Ordieres-Meré, J.; Villalba-Díez, J.; Zheng, X. Challenges and Opportunities for Publishing IIoT Data in Manufacturing as a Service Business. *Procedia Manuf.* **2019**, *39*, 185–193. [CrossRef]
29. Nakanishi, R.; Zhang, Y.; Sasabe, M.; Kasahara, S. IOTA-Based Access Control Framework for the Internet of Things. In Proceedings of the 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Paris, France, 28–30 September 2020; pp. 87–95.
30. Bhandary, M.; Parmar, M.; Ambawade, D. A Blockchain Solution based on Directed Acyclic Graph for IoT Data Security using IoTA Tangle. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 827–832.
31. Yu, Y.; Li, Y.; Tian, J.; Liu, J. Blockchain-Based Solutions to Security and Privacy Issues in the Internet of Things. *IEEE Wirel. Commun.* **2018**, *25*, 12–18. [CrossRef]
32. Lamtzidis, O.; Gialelis, J. An IOTA Based Distributed Sensor Node System. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [CrossRef]
33. Yan, J.; Zhang, F.; Ma, J.; An, X.; Li, Y.; Huang, Y. Environmental Monitoring System Based on Blockchain. In Proceedings of the 4th International Conference on Crowd Science and Engineering, Jinan, China, 18–21 October 2019; pp. 40–43.
34. Shabandri, B.; Maheshwari, P. Enhancing IoT Security and Privacy Using Distributed Ledgers with IOTA and the Tangle. In Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 7–8 March 2019; pp. 1069–1075. [CrossRef]
35. Benedict, S.; Rumaise, P.; Kaur, J. IoT Blockchain Solution for Air Quality Monitoring in SmartCities. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019; pp. 1–6.

36. Guanochanga, B.; Cachipuendo, R.; Fuertes, W.; Benitez, D.S.; Toulkeridis, T.; Torres, J.; Villacis, C.; Tapia, F.; Meneses, F. Towards a real-time air pollution monitoring systems implemented using wireless sensor networks: Preliminary results. In Proceedings of the 2018 IEEE Colombian Conference on Communications and Computing (COLCOM), Medellin, Colombia, 16–18 May 2018. [CrossRef]
37. 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.
38. Bures, M.; Klima, M.; Rechtberger, V.; Ahmed, B.S.; Hindy, H.; Bellekens, X. Review of Specific Features and Challenges in the Current Internet of Things Systems Impacting Their Security and Reliability. In *Trends and Applications in Information Systems and Technologies*; Springer International Publishing: Cham, Switzerland, 2021; pp. 546–556.
39. Zheng, X.; Sun, S.; Mukkamala, R.R.; Vatrapu, R.; Ordieres-Meré, J. Accelerating health data sharing: A solution based on the internet of things and distributed ledger technologies. *J. Med. Internet Res.* **2019**, *21*, e13583. [CrossRef] [PubMed]
40. Zhang, Y.; Nakanishi, R.; Sasabe, M.; Kasahara, S. Combining IOTA and Attribute-Based Encryption for Access Control in the Internet of Things. *Sensors* **2021**, *21*, 5053. [CrossRef] [PubMed]
41. Brogan, J.; Baskaran, I.; Ramachandran, N. Authenticating Health Activity Data Using Distributed Ledger Technologies. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 257–266. [CrossRef] [PubMed]
42. Zhang, Y.; Wu, S.; Jin, B.; Du, J. A blockchain-based process provenance for cloud forensics. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 2470–2473.
43. Zhou, X.; Tang, X. Research and implementation of RSA algorithm for encryption and decryption. In Proceedings of the 2011 6th International Forum on Strategic Technology, Harbin, Heilongjiang, 22–24 August 2011; pp. 1118–1121.
44. Mohassel, P. One-Time Signatures and Chameleon Hash Functions. In *Selected Areas in Cryptography*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 302–319.
45. Buchmann, J.; Dahmen, E.; Klintsevich, E.; Okeya, K.; Vuillaume, C. Merkle Signatures with Virtually Unlimited Signature Capacity. In *Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 31–45.
46. Colavita, M.; Tanzer, G. A Cryptanalysis of IOTA's Curl Hash Function. 2018, pp. 1–13. Available online: https://www.boazbarak.org/cs127/Projects/iota.pdf (accessed on 3 December 2021).
47. Tilkov, S.; Vinoski, S. Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Comput.* **2010**, *14*, 80–83. [CrossRef]
48. Florea, B.C. Blockchain and Internet of Things data provider for smart applications. In Proceedings of the 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2018; pp. 1–4.
49. Randles, B.M.; Pasquetto, I.V.; Golshan, M.S.; Borgman, C.L. Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study. In Proceedings of the 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), Toronto, ON, Canada, 19–23 June 2017; pp. 1–2.
50. Sarfraz, U.; Zeadally, S.; Alam, M. Outsourcing IOTA proof-of-work to volunteer public devices. *Secur. Priv.* **2020**, *3*, e98. [CrossRef]