electronics

MDPI

Article

A Cost Analysis of Internet of Things Sensor Data Storage on Blockchain via Smart Contracts

Yeşem Kurt Peker [1,*], Xavier Rodriguez [2], James Ericsson [1], Suk Jin Lee [1] and Alfredo J. Perez [1]

[1] TSYS School of Computer Science, Columbus State University, Columbus, GA 31907, USA; ericsson_james@columbusstate.edu (J.E.); lee_suk@columbusstate.edu (S.J.L.); perez_alfredo@columbusstate.edu (A.J.P.)

[2] Department of Electrical and Computer Engineering, Ana G. Méndez University, Gurabo, PR 00777, USA; xars.acm@gmail.com

* Correspondence: peker_yesem@columbusstate.edu

Received: 30 November 2019; Accepted: 26 January 2020; Published: 2 February 2020

Abstract: Blockchain is a developing technology that can be utilized for secure data storage and sharing. In this work, we examine the cost of Blockchain-based data storage for constrained Internet of Things (IoT) devices. We had two phases in the study. In the first phase, we stored data retrieved from a temperature/humidity sensor connected to an Ethereum testnet blockchain using smart contracts in two different ways: first, appending the new data to the existing data, storing all sensor data; and second, overwriting the new data onto the existing data, storing only a recent portion of the data. In the second phase, we stored simulated data from several sensors on the blockchain assuming sensor data is numeric. We proposed a method for encoding the data from the sensors in one variable and compared the costs of storing the data in an array versus storing the encoded data from all sensors in one variable. We also compared the costs of carrying out the encoding within the smart contract versus outside the smart contract. In the first phase, our results indicate that overwriting data points is more cost-efficient than appending them. In the second phase, using the proposed encoding method to store the data from several sensors costs significantly less than storing the data in an array, if the encoding is done outside the smart contract. If the encoding is carried out in the smart contract, the cost is still less than storing the data in an array, however, the difference is not significant. The study shows that even though expensive, for applications where the integrity and transparency of data are crucial, storing IoT sensor data on Ethereum could be a reliable solution.

Keywords: blockchain; Internet of Things; sensor data; smart contracts

1. Introduction

Blockchain technology is revolutionizing the way we store, share, and interact with data. It provides various essential services for data storage including immutability, transparency, decentralization, and fault tolerance without the need of a central authority. Initially developed by Satoshi Nakamoto as a trusted, distributed ledger system for the Bitcoin cryptocurrency [1], blockchain technology is currently being researched in areas such as healthcare, education, smart cities, financial services, logistics and supply chain, provenance, electronic voting, among other applications.

In the realm of Internet of Things (IoT) systems, the use of blockchain is currently researched under various aspects including decentralized architectures, authentication, autonomy, security, and marketplaces [2]. Many IoT systems are not computationally powerful in terms of storage and processing power, so much of the research in the integration of blockchain and IoT systems have focused on the computational cost to validate, process and integrate IoT devices in blockchain networks

under security constraints [2]. However, no rigorous study of the monetary cost to store IoT sensor data has been published to date.

Storing sensor data on a public blockchain where anyone can access it (almost in real-time), is a useful capability. For example, sensors that check air quality, monitor food temperature, or water quality in an area could publish their readings on a public blockchain which would allow reliable and tamper-resistant data storage. Any researcher could potentially download and use the data to address problems of interest in a community. The data could also provide reliable resources for auditing purposes [3]. In addition, using blockchain systems to store sensor data could potentially enable utilization of these systems for crowdsensing in which contributors could sell/auction their collected data to interested third parties, hence creating distributed marketplaces using smart contracts [4] for IoT sensor data and, at the same time, alleviating some privacy aspects in the utilization of IoT sensor data in crowdsensing systems since contributors would agree on a smart contract in the blockchain itself, and they would give consent through the smart contract.

In this work, we shed light on the monetary cost of storing sensor data on a public blockchain. We test various storage options for data from one sensor and data from several sensors. Our study comprises two phases. In the first phase, we describe an IoT testbed to store sensor data on the Ropsten Ethereum testnet blockchain via smart contracts and we examine the monetary cost of this operation under two different storage options: (1) appending the new data to the end of the existing data, hence storing all data; (2) overwriting on the existing data, hence storing a most recent portion of the data. In the second phase, we examine the cost when storing data from several IoT sensors on the blockchain. In this phase we simulate data for several sensors assuming sensor data is numeric; in particular, an integer and store the data on the blockchain in three ways: (1) storing the data in an array, (2) encoding the data from all sensors into one variable outside the smart contract and storing the variable on the smart contract, and (3) encoding the data from all sensors into one variable within the smart contract and storing the variable on the smart contract. Our results indicate that even though it could be expensive, for applications where the integrity and transparency of data are crucial, storing IoT sensor data on Ethereum could be a reliable solution.

Thus, the main contributions of this work are as follows:

- We describe an experimental testbed to store IoT sensor data on a public Ethereum-based blockchain network.
- We analyze the monetary cost (in gas, which is a monetary unit in Ethereum) when storing IoT sensor data using two storage mechanisms.
- We propose a method to store encoded data from several sensors for more efficient storage and examine the cost when storing data from several IoT sensors on the Ethereum blockchain using three methods.

This work is different than the work presented by Park et al. [4] and by Javaid [5] because we evaluate the monetary cost of storing IoT sensor data in a public blockchain rather than the monetary cost to create an IoT marketplace. In addition, both of the aforementioned studies use a model based on strings of characters (with no sensor data or IoT device used) to experiment with their market model. To the best of our knowledge, this work is the first analysis of the monetary cost of storing IoT sensor data on a public blockchain. The rest of the paper is organized as follows. Section 2 presents a review of the related work. In Section 3 we describe the testbed and the experiments to analyze the monetary cost of IoT sensor data in a public blockchain. Finally, Section 4 provides some concluding remarks.

2. Related Work

2.1. The Internet of Things

The Internet of Things (IoT) is a term that encompasses the development of Cyber-Physical Systems (CPS) that collect, share data and perform actions on some type of physical process while

connected to the Internet. Some IoT application areas include smart cities, intelligent transportation, entertainment, security, agriculture [6], and healthcare [7]. For example, the use of IoT technologies can improve the efficiency and effectiveness of rural systems [6], and IoT-enabled systems can deal with elderly monitoring and introduce a hierarchical model for elderly-centered monitoring [8]. Combined with advances in artificial intelligence, the IoT is having a significant impact on how consumers perform various activities in their daily lives especially in terms of making many of these activities a lot easier to perform. The availability and growth of these IoT devices (estimated to be about 75 billion by 2025 [9]) make computation transparent, in the sense that people are not aware of the availability of these devices and what they do in their surroundings. Typically, the architecture of IoT systems is made up of the following components:

- Internet of Things device: These components collect data (e.g., temperature, movement, sound, images) from physical actions or processes. In addition, IoT devices may perform initial data verification, aggregation and basic analysis (e.g., feature extraction) on the collected data. Some IoT devices may have actuators (e.g., rotors, relays, speakers, lights) that allow the IoT device to perform some type of physical response in the environment.
- Data transport: This part of the IoT system represents the communication network between the IoT device and cloud services. Typically, this is performed by cellular networks and the Internet. However, communication can be accomplished by home service Internet providers and WiFi.
- Cloud services: These components collect and store data sent from IoT devices. They also provide analytics services and feedback to IoT devices. Some cloud services may share data externally with other parties.

IoT systems can be classified into two broad categories: special-purpose IoT and consumer IoT. Special-purpose IoT systems are developed to satisfy the application requirements in specific realms (e.g., supervisory control and data acquisition (SCADA) systems, supply chain, smart agriculture [6]), and they require access to dedicated companies. In contrast, consumer IoT systems are easily acquired by the general public, and they generally include wearables, smart homes, and mobile IoT [10,11].

- Wearables: These are computers with embedded sensors and actuators/output devices developed as a garment, accessory, or device that is worn (or carried around) by consumers.
- Smart homes: These devices are deployed in homes with the goal of simplifying a consumers' life from the perspective of security, comfort and entertainment. This category may include Internet-connected toys.
- Mobile IoT: This category encompasses bicycles, smart cars, drones and others that people use either for transportation and/or leisure. This category may also include smartphones.

2.2. Blockchain, Smart Contracts and Their Applications

According to Bashir [12], blockchain is defined as "a peer-to-peer, distributed ledger that is cryptographically secure, append-only, immutable, and updateable only via consensus or agreement among peers" [12]. It is a ledger that consists of blocks chained together with cryptographic hashes and is distributed over all the nodes in the system. Each block in the chain contains transactions. Transactions posted from the nodes on the blockchain are verified based on a predetermined set of rules and only valid transactions are selected for inclusion in a block. The next block to be chained to the ledger is determined by a consensus algorithm which is run by all nodes in the system [13]. This distributed consensus mechanism is the primary underpinning of a blockchain. It allows a blockchain to present a single version of the truth which is agreed upon by all parties without the requirement of a central authority. The first blockchain, Bitcoin, was created by a person/group known by the pseudonym Satoshi Nakamoto in 2008 [14]. Bitcoin is one specific application of blockchain technology; namely, a cryptocurrency. Since the introduction of Bitcoin, the underlying, blockchain technology has been studied extensively to adapt it for other uses. It is still being actively researched by academia, government, industry and other interested parties.

There are two types of Blockchain: public and private, or more commonly called permissioned. Public blockchains are open to everyone; anyone can run applications and join the network. Bitcoin is an example of a public blockchain. Another well-known public blockchain is the Ethereum Mainnet. Ethereum is a global, open-source platform for decentralized applications. It was proposed in 2013 by Buterin [15]. Different from Bitcoin, Ethereum blockchain allows smart contracts.

A smart contract is a "computer program that encapsulates the business logic and code needed to execute a required function when certain conditions are met." [12]. An important feature of blockchain technology is that it provides a platform to execute smart contracts. The concept of smart contracts was first introduced by Nick Szabo in 1977 [16]. With the advent of blockchain technology, smart contracts have found applications especially in the financial services industry due to the reduced cost of transactions and simplification of complex contracts. Not all blockchains support smart contracts. For example, Bitcoin is a cryptocurrency-only platform.

Permissioned blockchains are blockchains in which access is controlled and only those who are vetted can participate in the network. Permissioned blockchains are popular among industry-level enterprises and businesses for which security, identity, and role definition are important. A prime example of permissioned blockchains is the Hyperledger project [17]. Hyperledger itself is not a blockchain; it is an umbrella project that offers the necessary framework, standards, guidelines and tools to build open source blockchains and related applications for use across various industries. It is a global collaboration, hosted by The Linux Foundation including leaders in finance, banking, Internet of Things, supply chains, manufacturing and technology. It was launched in 2016. Some projects and tools under the Hyperledger project include Fabric, Sawtooth, Iroha, Burrow, Indy, Cello, Composer, Explorer, and Quilt.

Blockchain development has gained adoption in various sectors since its introduction with Bitcoin, and it is being investigated for use in healthcare [18], education [19], smart contracts [20], smart cities [21], smart homes [22], finance [23], supply chains [24], provenance [25], electronic voting [26] and other sectors.

Kuo et al. describe benefits of blockchain for biomedical/health care applications compared with traditional distributed databases. With the benefits of blockchain that include decentralized management, immutable audit trail, data provenance, robustness and availability, and improved security and privacy, it can improve medical record management, enhance the insurance claim process, accelerate clinical/biomedical research, and advance biomedical/health care data ledgers [27]. Other use cases in healthcare and life sciences (HCLS) of blockchain architectures are described by Curbera et al. [28].

Christidis et al. described how smart contracts reside on the blockchain for the automation of multi-step processes and how a blockchain cooperates with IoT in terms of services/resources sharing and a cryptographical automation. They mention that the blockchain-IoT combination significantly change several industries in the form of new business models and novel [20].

Park et al. propose a review system that can confirm the reputation of the owner of data or the traded data in the P2P marketplace that is based on the Ethereum smart contracts [4]. They note that all functions to be processed by Ethereum transactions require a certain amount of gas and analyze the performance of their proposed model with required gas. Pieroni et al. [21] investigated the implementation of smart energy grid for citizens in the urban context for smart environments based on blockchain. In their work, they implemented a mobile application to enable citizens access to a blockchain network. Their distinctive solution uses the blockchain technology to join the grid, and it trades energy between energy providers and private citizens with blockchain granting ledger.

Litke et al. [24] analyzed the blockchain adoption for a large-scale deployment on the supply chain management industry. They investigated the factors that affect the adoption of blockchain in supply chain including scalability, performance, consensus mechanism, privacy considerations, location proof and cost. Ferraro et al. [29] described how distributed ledger technologies (DLTs) can be used to enforce social contracts and to orchestrate the behavior of agents trying to access a shared

resource. After analyzing the advantages and disadvantages of using DLTs architectures to implement certain control systems in an Internet of Things (IoT) setting, they proposed an application of DLTs as a mechanism for dynamic deposit pricing, wherein the deposit of digital currency is used to orchestrate access to a network of shared resources.

Mezquita et al. [30] propose a multi-agent system that combines smart contracts and blockchain to enable peer-to-peer electricity trading in a micro-grid scenario without the need for human intervention. The agents manipulate the blockchain through smart contracts creating a transparent and efficient market of electricity in which the peers trade electricity directly between themselves. They use the Ethereum blockchain and network of nodes, in order to have a decentralized tamper-proof registry which allows for the establishment of trustful agreements between agents, and the data recorded by them.

2.3. Blockchain Technology and IoT

An IoT system is composed of connected devices that communicate with each other for various purposes. The devices' ability to connect and communicate with each other without human interaction has enabled many conveniences for human in areas such as smart homes, smart vehicles, medical care, agriculture, manufacturing, and others [23]. In industrial settings, the integration of blockchain technology and the Industrial Internet of Things (IIoT) is bringing benefits to produce and manufacture goods with tight integrations between business partners [31].

Security is at the heart of most of these applications [32]. However, security issues in IoT systems have not been fully addressed in the rapid development of the concept/technology. Many security challenges remain to be solved in the IoT realm. For some of these challenges, blockchain technology is being investigated as a viable solution in academic, industry and government circles [31]. Khan et al. provide a categorization of security issues in IoT systems into low-level, intermediate-level and high-level issues based on the IoT deployment architecture [33]. They discuss how some of the intrinsic features of blockchain can be useful in addressing some of these challenges. Areas where blockchain could be useful according to authors include address space, identity of things and governance, data authentication and integrity, authentication, authorization, privacy, and secure communications.

Makhdoom et al. provide an excellent survey on the progression of blockchain technology and its impact on IoT [34]. After discussing the IoT threat environment and an overview of the blockchain technology, the authors discuss the challenges to blockchain's adoption in IoT. Yeray et al. also point out the limiting factors of blockchain technology impeding the rise in value of IoT systems [35]. The primary challenge they observe is the non-availability of an IoT centric consensus protocol. Most of the current consensus mechanisms are too heavy for IoT devices with limited computing power and storage capacity. Another challenge is the lack of transaction validation rules that can accommodate IoT systems which usually comprises heterogeneous devices, sending of sensor data, and/or data in distinct formats and different range of values. Scalability is also one of the challenges as the size of blockchain grows quickly especially in IoT systems [35]. Also, for many IoT systems such as wireless sensor networks (WSNs), industrial control systems (ICS), smart vehicles and smart grids, the sharing of real-time data is required. Better transaction confirmation times, without compromising the security, are needed to achieve real-time data sharing. Another problem in applying blockchain technology to the IoT realm is the integration of the IoT devices to the blockchain. Mechanisms to ensure the authenticity of the device and the integrity of data from the device are needed. Finally, the lack of a mechanism for secure and synchronized software upgrades of the devices when they remain in continuous operation is another challenge. Additional issues that may apply to certain use cases are user privacy, data security (encryption), and legal issues [36].

Jesus et al. analyzed the uses of blockchain to achieve security and privacy in IoT. They listed the limited capabilities, high transaction costs, and in certain cases privacy requirements for the data as main issues in integration of blockchain in IoT [37]. Ourad et al. [38] proposed a blockchain-based authentication and access control mechanism for IoT devices based on smart contracts. After the user

successfully authenticates to the smart contract, the user and the IoT device receive an authentication token and the Ethereum address of the authorized user. In their work, they compared their solution with Oauth2, Auth0, and Blockstack. They also showed that their solution using Ethereum smart contracts can provide tamper proof records and decentralization to improve current approaches.

Guin et al. [39] presented an authentication scheme using blockchain technology to authenticate IoT edge devices. Their scheme uses Static Random Access Memory–based (SRAM-based) physically unclonable functions to generate unique digital fingerprints for device IDs. Registered manufactures upload their IDs to a globally accessible blockchain. When a device is to be deployed in an IoT network, a locally permissioned blockchain is used to authenticate the device through the global blockchain. The dual blockchain approach counters the counterfeit and clone problems, enhances the reliability and usability of the supply chain, and ensures authenticity of edge devices.

Chen et al. [40] proposed a method based on blockchain technology to protect data integrity in the IoT. They develop a stochastic blockchain scheme to limit the number of cooperative nodes and distribute the load among IoT edge nodes. They also proposed a lightweight mining process designed for their scheme. Huang et al. [41] proposed a credit-based consensus mechanism for industrial IoT systems. In their work, they propose a credit-based proof-of-work (PoW) mechanism which can guarantee system security and transaction efficiency simultaneously. They also designed a data authority management method to regulate access to sensor data to protect the confidentiality of sensitive data [41].

Casado-Vara [42] et al. proposed the utilization of an adaptive control algorithm for IoT systems to optimize the block mining process. In their work, they made use of queuing theory and non-linear closed control loop to determine when an IoT device should send data to be stored in the blockchain so the mining process is more efficient. In addition to the queuing model and control system, they also proposed an IoT network architecture made of sidechains for IoT devices and fully connected blockchain nodes.

Ejaz et al. [32] investigated prime issues related to the successful adoption of blockhain-based IoT systems and their applications. In their research, they present two case studies related to the adoption of blockchain in IoT systems with applications in smart homes and food supply chain traceability to show the effectiveness of IoT blockchain technology in these applications.

3. Methodology

3.1. Experimental Testbed

In this study we created a testbed to store IoT sensor data in a public blockchain system. The IoT testbed used to collect IoT sensor data is composed of the following elements:

- A DHT11 temperature/humidity sensor: The DHT11 is a low-cost (less than $10 USD) capacitive sensor that captures humidity and temperature data, and provides a digital output. A limitation of the DHT11 sensor is that it generates data every two seconds.
- A Raspberry Pi 3 Model B device: The Raspberry Pi Model 3B is a small, low-cost and embeddable computer that can be used to create IoT prototypes. The 3B model is powered by a 1.32 GHz quad-core processor, it has 1GB of RAM, on-board WiFi/Bluetooth connectivity, and a 40-pin GPIO bus [43]. Figure 1 shows an image of the Raspberry Pi with the DHT11 sensor. The Raspberry Pi acts as a lightweight Ethereum node.
- MetaMask: MetaMask is a browser extension that acts as a bridge between Internet browsers, Ethereum-based blockchains, and decentralized applications (DApps) running on a full Ethereum node [44]. In our testbed, MetaMask was used as an Ethereum wallet and this extension is executed on the Raspberry Pi device. We used MetaMask because it provided an Ethereum faucet in which a wallet can obtain a limited number of free funds in Ethereum's digital currency (ethers) for the IoT sensor transactions/operations.

- Infura: Infura is an infrastructure that offers a suite of tools to connect apps to the Ethereum network [34]. It runs fully connected Ethereum nodes and provides access to them via an application programming interface (API). Once an account and project are created in Infura, an URL end point that can be used to push the smart contract transactions onto the blockchain via web services. Infura is needed because of the computing power of the Raspberry Pi does not allow to connect it directly to the Ethereum network.
- Ropsten: Ropsten is an Ethereum blockchain to test applications (a testnet) [29]. Among the testnets, Ropsten is the one that most closely follows the main Ethereum blockchain (Mainnet). Ropsten uses the proof of work (PoW) consensus mechanism, which is the same mechanism used by the Mainnet.

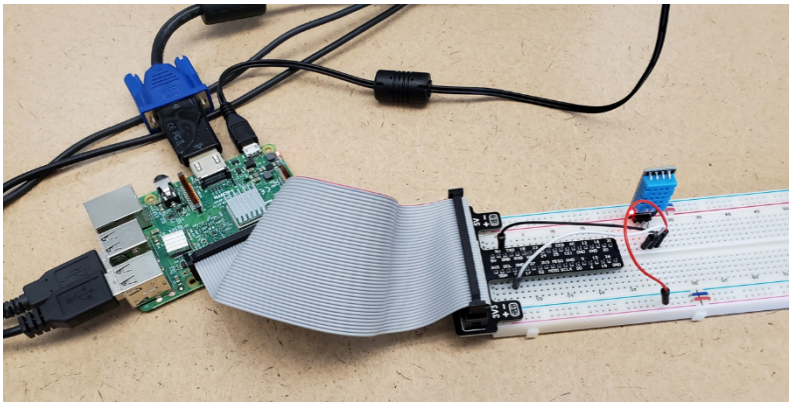

Figure 1. Prototype Raspberry Pi IoT (Internet of Things) device used for the testbed.

We used two major tools to collect data, write and deploy the smart contract transactions on Ropsten. The first tool is the Remix IDE [32], which is a web-based environment used to develop and test smart contracts using the Solidity programming language. Remix was used to upload the smart contract to the Ropsten testnet and it provided the application binary interfaces (ABI) to interact with the contract. The second tool is the Web3 Python library which we used in a Python script that was executed in the Raspberry Pi IoT device. The Web3 library contains the necessary methods to interact with the blockchain including signing and sending transactions, receiving receipts, among others. The Python script that we created was executed in the Raspberry Pi IoT device to collect sensor data, create, sign and send transactions to Ropsten via the Web3 library and Infura, as well as to log transactions. Thus, steps performed by this Python script (Figure 2) are as follows:

1. Sensor data collection from the sensor at the Raspberry Pi.
2. Creation of blockchain transaction data with inclusion of nonce, origin address, destination address, and other data needed to process the transaction in the blockchain.
3. Data signing with the account's private key to authenticate that the correct account is sending the data.
4. Sending transaction data to Ropsten in which fully connected nodes receive the data via Web3 and Infura.
5. Extraction of the transaction's receipt from the blockchain, in which the cost, specifically the monetary cost in gas usage of the operation. At this point the script repeats again from step 1.

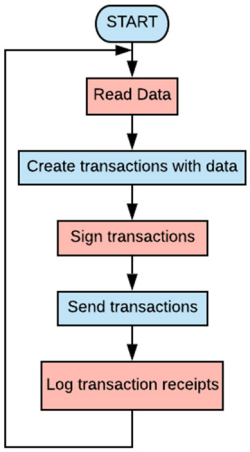

Figure 2. Steps performed by Python script at the Raspberry Pi IoT device.

3.2. Experimental Procedure

With the testbed described in the previous section, we then proceed to evaluate the monetary cost of IoT sensor data storage in Ropsten under two scenarios:

- Storing all data on a contract (array appending): In this scenario, we appended the new data generated by the IoT device at the end of an expanding array stored in the smart contract.
- Storing only a recent portion of the data on a contract (array substitution): In this scenario, we allocated a fixed array in the smart contract and stored only the most recent portion of the data that could fit in the array, thus overwriting the array entries. In this last scenario, we performed two types of substitution: substitution with array size = 200 entries and substitution with array size = 2000 entries.

In Ethereum the amount of computational effort required to execute the operations in a transaction or a smart contract is measured in gas. The basic operations allowed in an Ethereum blockchain and how much gas they require are published in the Ethereum beige paper [45]. When a transaction is created, a gas limit, which is the maximum amount of gas the originator is willing to pay for, along with how much the originator is willing to pay per gas is specified in the transaction. The currency to pay for gas is ether, more specifically the gas cost is specified in gwei which is 10^{-9} ether. The price for gas is not fixed. It is up to the sender of a transaction to specify any gas price they like. On the other side, it is up to the miner to verify any transactions they like. Naturally, to maximize their profit, the miners usually pick the transactions that specify the highest gas price.

If an operation runs out of gas, the miners stop mining and the operation is reverted back to its original state. The originator still pays the miners the fee for their computational costs and the operation gets added to the blockchain even though it is not executed. If there is any gas left over, it is refunded to the originator. There are mechanisms in place in Ethereum to prevent bloated gas limits in transactions [15,45]. On the most part the price one is willing to pay per gas determines the speed of the transaction. The website ethgasstation.info [46] provides information about current gas prices, transaction confirmation times, and miner policies on the Ethereum network. To gauge the cost of gas prices for different types of transaction speeds, we checked ethgasstation.info.

After pushing the transactions onto the blockchain, the Python script described in the previous section extracted the gas used for the operation from the transaction receipts when the sensor transactions were confirmed. Figure 3 shows a diagram with these steps. We also confirmed the transactions through etherscan for Ropsten [47], a web-based tool that allows viewing of all transactions on Ethereum blockchains including Ropsten.

We created three smart contracts to measure the gas cost of storing data on Ropsten. The first contract appended the data in an expanding array, thus storing all data points on the contract.

The second contract kept an array of size 200 (hence 200 recent data points were kept, making a substitution every time data sensor data was uploaded), and the third contract kept an array of size 2000 (hence 2000 recent data points were kept, making a substitution every time the IoT sensor data was uploaded). Ethereum allows unsigned integer data types of 8 bits to 256 bits with 8-bit increments named uint8, uint16, uint24, . . . , uint256. We initially tested storing individual 8-bit and 256-bit unsigned integers on the Ropsten blockchain and found that storing a 256-bit int (uint256) is not more expensive than storing an 8-bit int (uint8) in terms of gas usage. In fact, storing one 256-bit integer costs 26,796 gas which is slightly less than storing an 8-bit integer which costs 27,283 gas. For this reason, in the smart contracts, we used the data type uint256 even though the humidity sensor data would allow smaller number of bits to be used. When creating a transaction, and to make sure that we had enough gas to cover the extra operations for storage in an array, we specified the gas limit to be 140,000 which was consistent with the gas estimate that Remix provided. We specified the gas price to be 40 gwei for average transaction speed as was suggested by ethgasstain.info when we ran the transactions/experiments in July 2019.

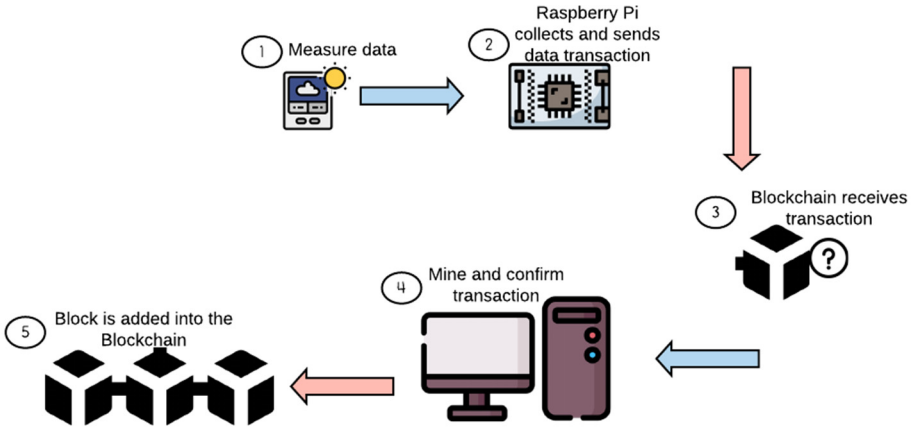

Figure 3. Steps to store sensor data in Ropsten as performed in our experiments.

3.3. Results

The sensor data and the gas consumption for the three smart contracts are shown in Figure 4 in top and bottom graphs, respectively. The graphs are shown in one figure to make the relation between sensor readings and costs for storing them more apparent. As shown in Figure 4 (top), humidity readings fluctuated between 50% and 80% during the execution of the experiments. We collected more than 2000 data points which were stored on Ropsten using the three different smart contracts. We observed from Figure 4 (bottom) that the gas cost was mostly affected by the method used to store the data on the blockchain.

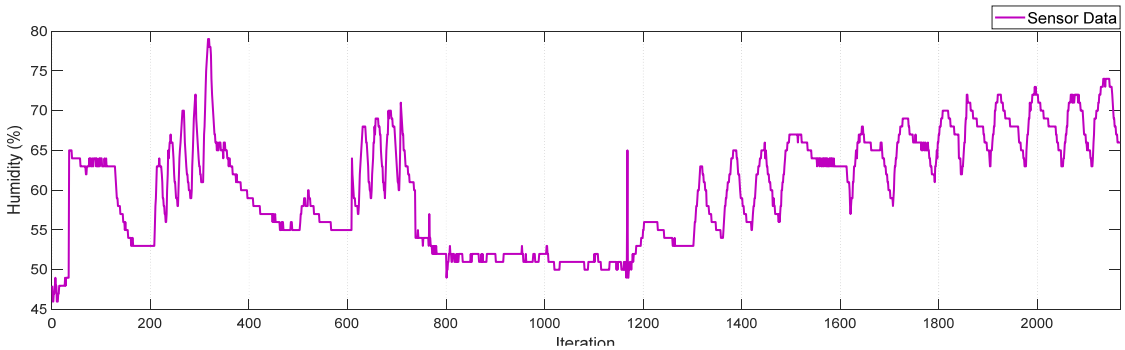

Figure 4. *Cont.*

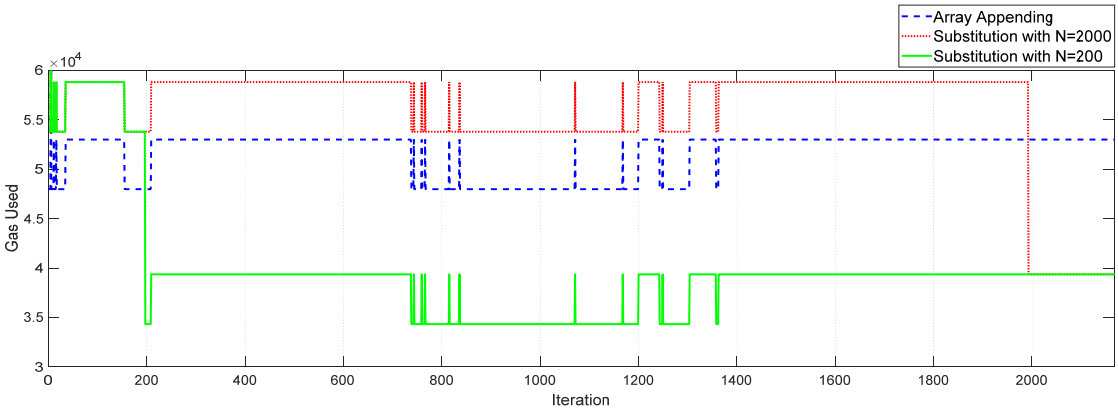

Figure 4. Data captured by the IoT device (**top**); gas consumption of storage options (Gas used: y-axis on left; data points) (**bottom**).

In the array substitution method, the average cost of the substitution method was 58,777 gas. However, once the actual substitution began, the cost was reduced by 33% (39,905 gas on average). In our experiments, the most expensive operations were array initializations and the least expensive were those of array substitution. From our experiments, array substitution was cheaper than array appending by 26% on average. We can deduce then that the smart contract operations have the most impact on the gas cost, and the size of the data holds little no to weight affecting the increase/decrease of gas.

We now calculate the total cost of storing T data points on the blockchain under the two different storage options as follows (in gas):

Method 1—array appending:

$$T_{\text{appending}} = \textit{Cost for Initialization} + \textit{Cost for appending } T \textit{ data points} \tag{1}$$

Method 2—array substitution (Keeping N data points):

$$\begin{aligned} T_{\text{substitution}} = {} & \textit{Cost for Initialization} + \textit{Cost for appending } N \textit{ data points} \\ & + \textit{Cost for overwriting } T - N \textit{ data points} \end{aligned} \tag{2}$$

Using the data from Table 1, we can calculate the cost of storing T data points on the blockchain as follows:

Method 1:

$$T_{\text{appending}} = 82{,}960 + 52{,}960 \times T, \tag{3}$$

Method 2:

$$T_{\text{substitution}} = 103{,}777 + 58{,}777 \times N + 39{,}305 \times (T - N), \tag{4}$$

For example, suppose that a business/entity needs to store 6000 data points on Ethereum in a month, their monthly gas cost would be:

Method 1: $T_{\text{appending}} = 82{,}960 + 52{,}960 \times 6000 = 317{,}842{,}960$ gas
Method 2: $T_{\text{substitution}}$ with $N = 200$:

$$T_{\text{substitution}} = 103{,}777 + 200 \times 58{,}777 + 5800 \times 39{,}305 = 239{,}828{,}177 \text{ gas}$$

$T_{\text{substitution}}$ with $N = 2000$:

$$T_{\text{substitution}} = 103{,}777 + 58{,}777 \times 2000 + 39{,}305 \times 4000 = 274{,}877{,}777 \text{ gas}$$

As of 22 November 2019, the recommended Ethereum gas price for standard mining in the Mainnet as published in ethgasstation.info was 10 gwei and 1 ether is 146.87 USD. Recalling that 1 gwei = 10^{-9} ether, we calculate the total cost in US Dollars (USD) for storing 6,000 data points as follows:

Method 1: $Cost_{\text{appending}}$ = 317,842,960 × 10 × 10^{-9} ether = 3.1784296 ether = 466.82 USD.
Method 2 with N = 200: $Cost_{\text{substitution}}$ = 239,828,177 × 10 × 10^{-9} =2.27969 ether = 334.82 USD.
with N = 2000: $Cost_{\text{substitution}}$ = 274,877,777 × 10 × 10^{-9} = 2.74877777 ether = 403.71 USD.

Table 1. Average gas consumption of storage options.

Method	Initialization	Adding First N Points	Adding after N Points
Array appending	82,960	52,960	52,960
Array substitution	103,777	58,777	39,305

3.4. Discussion

As previously stated, the gas required to push a new data point in the blockchain does not seem to depend on the size of the array in either appending or substitution methods. Out of the two methods, using substitution proved to be the more efficient method for data storage on the Ropsten Ethereum blockchain. The total gas cost for storing the most recent 200 data points was about 75.4% and storing the most recent 2000 data points was about 86.5% of that of appending.

On the Ethereum blockchain, when data is substituted it still exists in the ledger and can be retrieved from older blocks by looking at the transaction hash. The "get" functions in smart contracts do not change the state of the blockchain so they can be executed at no cost, allowing for cost-efficient data retrieval. As was mentioned in Section 3.2, our initial test showed that storing an unsigned 256-bit integer (in uint256 data type) on the blockchain costs 26,796 gas. If 6000 data points were stored individually on the blockchain, then the cost would be about 26,796 × 6000 = 160,776,000 gas. This would be about 51% of appending all data, 67% of storing the most recent 200 data points, and 59% of storing the most recent 2000 data points. In our study, we chose to store all or some portion of the data with the idea that the data could be used to make inferences about the environment from which the humidity sensor is collecting data. One out-of-place humidity value sensor may raise suspicion but when most of the recent values are abnormal, it may indicate problems with the environment, the sensor, or data submission process. Smart contracts can be used to take actions in these cases, although it will have its costs in gas units.

3.5. Storing Data from Several Sensors

In many applications data from several sensors are more useful. In this phase, we focus on storing data from several sensors on the blockchain. We assume that readings from all the sensors are written onto the blockchain at once via a smart contract. For example, we assume that there are 32 sensors and the data from each sensor is at most 8-bits long. Note that the total length of data from all sensors is 8 × 32 = 256 bits. The discussion below can be adapted to any number of sensors where the sum of maximum number of bits to represent the data for each sensor does not exceed 256. Since we assumed the sensor data is at most 8-bits long, we simulated the sensor data by generating random numbers between 0 and 255 in the Python script. We examined three methods of storing the data from the 32 sensors on the blockchain. The three methods and the gas usage for each transaction are given below. For each method we ran 20 simulations to obtain the gas usage. Except for a few cases, the gas usages were constant (equal to the values indicated in parentheses). This is as expected considering the gas cost structure in Ethereum. In the exceptions, the gas amounts were very close to the numbers indicated in parentheses.

- Method 1. Storing the data in an array of size 32 of uint8 values (gas: 88,600)

- Method 2. Encoding the 32 values into 256 bits in the Python script (outside the smart contract) and storing one uint256 on the blockchain (gas: 26,796)
- Method 3. Encoding the 32 values into 256 bits in the smart contract script and storing one uint256 on the blockchain (gas: 84,837)

The encoding in methods 2 and 3 is such that each 8 bits of the 256 bits represents the reading from one sensor. This is simply done by multiplying the sensor values by powers of 2^8 (i.e., 2^0, 2^8, 2^{16}, ..., 2^{248}) and adding them together. More specifically, if $d_0, d_1, d_2, \ldots, d_{31}$ are the data from the 32 sensors then the 256-bit integer that represents them is $d_0 x\ 2^0 + d_1 x\ 2^8 + d_2 x\ 2^{16} + \ldots + d_{31} x\ 2^{248}$. Note that each multiplication by 2^8 amounts to shifting the number 8 bits to the left.

With this encoding, method 2 comes down to storing an uint256 on the blockchain which needs 26,796 gas. This is the cheapest option as the encoding is done outside the smart contract. In our implementation the "get" method simply returns the 256-bit number so one needs to do the decoding once they get the data from the blockchain to get the actual sensor readings.

In method 3, the set method in the smart contract receives the 32 sensor readings in an array and encodes them into a 256-bit number value using the encoding described above and stores the result in an uint256 variable. As one expects, this is more expensive than method 2 because encoding operations cost gas. The gas required in method 3 is about 3.2 times the gas required in method 2. This directly translates to monetary equivalents. Even though more expensive, performing the encoding via the smart contract may be more desirable. It has the advantage of avoiding manipulations of data, hence providing stronger integrity.

We would like to note also that method 1, in which an array is used to store the data, costs the most, hence should be avoided unless there is an absolute need to store the sensor data separately in such a structure.

4. Conclusions

Blockchain is a developing technology that provides much desired security mechanisms such as integrity, authenticity, availability, and fault tolerance although at some cost. Blockchain is not suited for all scenarios but there are many cases where benefits may outweigh the cost. IoT is an area where developments are rapid but many challenges remain, especially with respect to security. Utilizing blockchain technology for challenges in IoT is an active area of research. More research that addresses the cost, in particular ways to minimize the cost of using blockchain for IoT, need to be conducted. Considering the amount of data generated by IoT devices, efficient representations of data that enable more manipulations via smart contracts at smaller costs should be studied. More functional smart contracts that can do some analysis on the data and provide alerts (for example light a led) when data is out of range, efficient ways to carry out these operations, and the cost analysis of such smart contracts could enable more practical applications of blockchain technology in IoT.

Author Contributions: Conceptualization, Y.K.P., methodology, Y.K.P.; software, J.E., X.R., Y.K.P., and S.J.L. validation, J.E., X.R., Y.K.P., and S.J.L.; resources, Y.K.P., A.J.P., S.J.L.; writing—original draft preparation, J.E., X.R., and Y.K.P.; writing—review and editing, S.J.L., A.J.P., and Y.K.P.; visualization, J.E. and X.R.; supervision, Y.K.P and S.J.L.; project administration, Y.K.P., S.J.L., and A.J.P.; funding acquisition, A.J.P., Y.K.P.

Funding: This research has been supported by the U.S. National Science Foundation and the U.S. Department of Defense under grant award No. 1560214, and the U.S. National Science Foundation under grant award No. 1950416, 'An REU Site on Security for Mobile Sensing and the Internet of Things'.

Acknowledgments: We thank the anonymous reviewers for their valuable comments, which helped improve the paper's content, quality, and organization.

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

References

1. Bello, G.; Perez, A.J. Adapting financial technology standards to blockchain platforms. In Proceedings of the 2019 ACM Southeast Conference, At Kenneasaw, GA, USA, 18–20 April 2019.

2. Reyna, A.; Martín, C.; Chen, J.; Soler, E.; Díaz, M. On blockchain and its integration with IoT. Challenges and opportunities. *Futur. Gener. Comput. Syst.* **2018**, *88*, 173–190. [CrossRef]
3. Kaaniche, N.; Laurent, M. A blockchain-based data usage auditing architecture with enhanced privacy and availability. In Proceedings of the 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 30 October–1 Novermber 2017.
4. Park, J.S.; Youn, T.Y.; Kim, H.B.; Rhee, K.H.; Shin, S.U. Smart contract-based review system for an IoT data marketplace. *Sensors* **2018**, *18*, 3577. [CrossRef] [PubMed]
5. Javaid, A.; Javaid, N. Ensuring Analyzing and Monetization of Data Using Data Science and Blockchain in IoT Devices. Master's Thesis, COMSATS University Islamabad, Islamabad-Pakistan, Comsats University, Islamabad, Pakistan, 2019.
6. González-Briones, A.; Castellanos-Garzón, J.A.; Martín, Y.M.; Prieto, J.; Corchado, J.M. A framework for knowledge discovery from wireless sensor networks in rural environments: A crop irrigation systems case study. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1–14. [CrossRef]
7. Perez, A.J.; Zeadally, S.; Cochran, J. A review and an empirical analysis of privacy policy and notices for consumer Internet of things. *Secur. Priv.* **2018**. [CrossRef]
8. Azimi, I.; Rahmani, A.M.; Liljeberg, P.; Tenhunen, H. Internet of things for remote elderly monitoring: A study from user-centered perspective. *J. Ambient Intell. Humaniz. Comput.* **2017**, *8*, 273–289. [CrossRef]
9. IoT: Number of Connected Devices Worldwide 2015–2025|Statista. Available online: Statista.com (accessed on 14 November 2019).
10. Perez, A.J.; Zeadally, S. Privacy Issues and Solutions for Consumer Wearables. *IT Prof.* **2018**, *20*, 46–56. [CrossRef]
11. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
12. Bashir, I. *Mastering Blockchain*, 2nd ed.; Packt Publishing: Birmingham, UK, 2018.
13. Nguyen, G.T.; Kim, K. A survey about consensus algorithms used in Blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128.
14. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Online*. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 14 November 2019).
15. Buterin, V. Ethereum White Paper. *GitHub Repos.* **2013**, *1*, 22–23.
16. Szabo, N. Smart Contracts: Building Blocks for Digital Free Markets. *Online*. 1996. Available online: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html (accessed on 2 February 2020).
17. Hyperledger. Available online: https://www.hyperledger.org/ (accessed on 29 November 2019).
18. Skiba, D.J. The Potential of Blockchain in Education and Health Care. *Nurs. Educ. Perspect.* **2017**, *38*, 220–221. [CrossRef]
19. Sharples, M.; Domingue, J. The Blockchain and Kudos: A Distributed System for Educational Record, Reputation and Reward. *Disinf. Open Online Media* **2016**, *9891*, 490–496.
20. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
21. Pieroni, A.; Scarpato, N.; di Nunzio, L.; Fallucchi, F.; Raso, M. Smarter City: Smart energy grid based on Blockchain technology. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2018**, *8*, 298–306. [CrossRef]
22. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017; pp. 618–623.
23. Ganne, E. *Can Blockchain Revolutionize International Trade?* World Trade Organization: Geneva, Switzerland, 2018.
24. Litke, A.; Anagnostopoulos, D.; Varvarigou, T. Blockchains for Supply Chain Management: Architectural Elements and Challenges Towards a Global Scale Deployment. *Logistics* **2019**, *3*, 5. [CrossRef]
25. Wang, J.; Wang, S.; Guo, J.; Du, Y.; Cheng, S.; Li, X. A Summary of Research on Blockchain in the Field of Intellectual Property. *Procedia Comput. Sci.* **2019**, *147*, 191–197. [CrossRef]
26. Ayed, A.B. A Conceptual Secure Blockchain Based Electronic Voting System. *Int. J. Netw. Secur. Its Appl.* **2017**, *9*, 1–9.

27. Kuo, T.T.; Kim, H.E.; Ohno-Machado, L. Blockchain distributed ledger technologies for biomedical and health care applications. *J. Am. Med Inform. Assoc.* **2017**, *24*, 1211–1220. [CrossRef]
28. Curbera, F.; Dias, D.M.; Simonyan, V.; Yoon, W.A.; Casella, A. Blockchain: An enabler for healthcare and life sciences transformation. *IBM J. Res. Dev.* **2019**, *63*, 8:1–8:9. [CrossRef]
29. Ferraro, P.; King, C.; Shorten, R. Distributed ledger technology for smart cities, the sharing economy, and social compliance. *IEEE Access* **2018**, *6*, 62728–62746. [CrossRef]
30. Mezquita, Y.; Gazafroudi, A.S.; Corchado, J.M.; Shafie-Khah, M.; Laaksonen, H.; Kamišalić, A. Multi-Agent Architecture for Peer-to-Peer Electricity Trading based on Blockchain Technology. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017.
31. Wang, Q.; Zhu, X.; Ni, Y.; Gu, L.; Zhu, H. Blockchain for the IoT and industrial IoT: A review. *Internet Things* **2019**. [CrossRef]
32. Ejaz, A.; Anpalagan, W. Blockchain Technology for Security and Privacy in Internet of Things. In Internet of Things for Smart Cities. In *Internet of Things for Smart Cities. Springer Briefs in Electrical and Computer Engineering*; Springer: Cham, Switzerland, 2019; pp. 47–55.
33. Khan, M.A.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Futur. Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
34. Makhdoom, I.; Abolhasan, M.; Abbas, H.; Ni, W. Blockchain's adoption in IoT: The challenges, and a way forward. *J. Netw. Comput. Appl.* **2019**. [CrossRef]
35. Mezquita, J.M.; Casado, Y.; Gonzalez-Briones, R.; Prieto, A.; Corchado, J. Blockchain Technology in IoT Systems: Review of the Challenges. *Ann. Emerg. Technol. Comput.* **2019**, *3*, 17–24. [CrossRef]
36. Mezquita, Y.; Valdeolmillos, D.; González-Briones, A.; Prieto, J.; Corchado, J.M. Legal aspects and emerging risks in the use of smart contracts based on blockchain. In *Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2019.
37. Jesus, E.F.; Chicarino, V.R.L.; de Albuquerque, C.V.N.; Rocha, A.A.D.A. A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack. *Secur. Commun. Netw.* **2018**, *2018*, 1–27. [CrossRef]
38. Ourad, A.Z.; Belgacem, B.; Salah, K. Using blockchain for IOT access control and authentication management. *Int. Conf. Internet Things* **2018**, *10972 LNCS*, 150–164.
39. Guin, U.; Cui, P.; Skjellum, A. Ensuring Proof-of-Authenticity of IoT Edge Devices Using Blockchain Technology. In Proceedings of the 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), Halifax, NS, Canada, 30 July–3 August 2018.
40. Chen, Y.J.; Wang, L.C.; Wang, S. Stochastic Blockchain for IoT Data Integrity. *IEEE Trans. Netw. Sci. Eng.* **2018**. [CrossRef]
41. Huang, J.; Kong, L.; Chen, G.; Wu, M.Y.; Liu, X.; Zeng, P. Towards secure industrial iot: Blockchain system with credit-based consensus mechanism. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3680–3689. [CrossRef]
42. Casado-Vara, R.; Chamoso, P.; de la Prieta, F.; Prieto, J.; Corchado, J.M. Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. *Inf. Fusion* **2019**, *49*, 227–239. [CrossRef]
43. Raspberry Pi 3B specifications. Available online: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ (accessed on 17 December 2019).
44. Metamask. Available online: https://metamask.io/ (accessed on 29 November 2019).
45. Dameron, M. Beigepaper: An Ethereum Technical Specification v.0.8.5. *Online*. 2019. Available online: https://github.com/chronaeon/beigepaper/blob/master/beigepaper.pdf (accessed on 1 February 2020).
46. Ethgasstation. Available online: https://ethgasstation.info/ (accessed on 29 November 2019).
47. Ropsten Etherscan. Available online: https://ropsten.etherscan.io/ (accessed on 29 November 2019).