electronics

MDPI

Article

Proposed Model for Secured Data Storage in Decentralized Cloud by Blockchain Ethereum

Nabeel Khan [1], Hanan Aljoaey [1], Mujahid Tabassum [2], Ali Farzamnia [3,*], Tripti Sharma [4] and Yew Hoe Tung [3]

[1] Department of Information Technology, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia
[2] Noroff Accelerate, Noroff School of Technology and Digital Media, 4608 Kristiansand, Norway
[3] Faculty of Engineering, Universiti Malaysia Sabah, Kota Kinabalu, Sabah 88400, Malaysia
[4] Department of IT, University of Technology and Applied Sciences, Muscat 133, Oman
* Correspondence: alifarzamnia@ums.edu.my

Abstract: Since cloud computing is an essential component of any modern company (usually accounting for a considerable share of information technology (IT) infrastructure investment), consumers rely on cloud services. Data privacy and security are worries when data remains in third-party storage. Existing document version control systems are centralized and at risk from data loss, as seen by higher time utilization and incorrect document update procedures that allow modifications to a document without the awareness of other network operators. Underutilized peer resources might be leveraged to construct storage. According to this argument, an elevated level of data security may be obtained by encrypting the data and dispersing it among numerous nodes. In this study, we attempted to review the security of cloud systems when using the blockchain Ethereum, and cloud computing was briefly discussed with its advantages and disadvantages. The idea of a decentralized cloud was briefly demonstrated with blockchain technology. Furthermore, previous papers were reviewed and presented in tabular form. This dictated that there are still research gaps in the field of blockchain-based cloud computing systems. This study proposed a model for secured data storage over a decentralized cloud by blockchain Ethereum.

Keywords: cloud computing; decentralized cloud; data storage; blockchain Ethereum; encryption algorithm

Citation: Khan, N.; Aljoaey, H.; Tabassum, M.; Farzamnia, A.; Sharma, T.; Tung, Y.H. Proposed Model for Secured Data Storage in Decentralized Cloud by Blockchain Ethereum. *Electronics* **2022**, *11*, 3686. https://doi.org/10.3390/electronics11223686

Academic Editors: Sebelan Danishvar, Morad Danishvar, Seyed Naser Razavi and Flavio Canavero

Received: 4 July 2022
Accepted: 27 October 2022
Published: 10 November 2022

1. Introduction

Recent advancements in data processing technologies interest ordinary consumers seeking improved data storage. Cloud computing is used as a system for cloud customers. Depending on their location, cloud users have access to distribute or trade data at any time and from any site. As we progress into the "technology age," we can see an enormous increment, pace, and diversity of material on the internet. Data can come from various sources, including mobile devices, archives, sensors, and social networks [1].

Due to the substantial increase in data, cloud storage is necessary to store data. Most of the data now available over the internet is highly centralized and is held by a few technological corporations with the ability and funding to make extensive cloud services able to manage this large dataset. The issue with this method is data security [2]. As a result, a trusted computer system utilizes hardware and software assurance methods to process several classified or sensitive data levels. This system satisfies the specified dependability and security criteria. Unfortunately, despite the numerous benefits of the cloud in ensuring the confidentiality, integrity, and availability of stored data, the number of security breaches is continuously rising. Even the advent of cloud technology comes with several different risks for organizations, particularly in secure computing.

Ensuring the security of an organization's private cloud while also overseeing the operations of the cloud services provider may be a daunting undertaking. A wide range

of issues like compliance, controls, administrative challenges, security management, and security awareness influence and directly impact cloud security architecture. A similar problem is the management strategy for data kept in the cloud. Security and privacy must be considered when a client's involvement with a cloud provider ends. In certain circumstances, information must be retained to follow safety obligations. In contrast, the provider should not keep a customer's data in primary or backup storage if the client believes it has already been deleted. Furthermore, if data is held in a specific country, it may be governed by that country's privacy laws rather than the rules applicable in the client's geographic region [3].

On the other hand, it provides a significant possibility for a new cloud industry, matching the supply and demand for IT resources of a large user base through decentralized cloud storage services. Blockchain is a decentralized public ledger that stores all digital currencies like Bitcoin or digital money. The transactions are kept in blocks; cryptographic hash values are linked to form the blockchain, unlike the older method, which required companies to trust third parties to function. Over 25% to 50% of supply chain specialists support blockchain for lowering transaction costs and boosting supply chain transparency. Paper-based systems were previously employed, which had the disadvantage of file system techniques. However, the challenges of having paper-based procedures were addressed with digitization [4]. This study focused on these challenges by proposing a decentralized client management framework based on the Ethereum blockchain [5]. Ethereum is more than simply a cryptocurrency system; it is a network of independent computers that combine to make one supercomputer [6]. It is adaptable because it allows transactions to be established via a permission or permissionless network. Furthermore, it is a blockchain-based platform for implementing smart contracts. Thus, it enables more than simply Bitcoin transactions. The Ethereum Virtual Machine is the name given to this platform (EVM). An Ethereum wallet, depending on the kind of account, is a smart contract wallet that may also develop, deploy, or activate smart contracts using the Solidity programming language [7]. Across hundreds of computers, nodes repeat the same transactions and maintain the same state, shown in Figure 1. The EVM comprises many private computers, such as a shared ownerless computer.

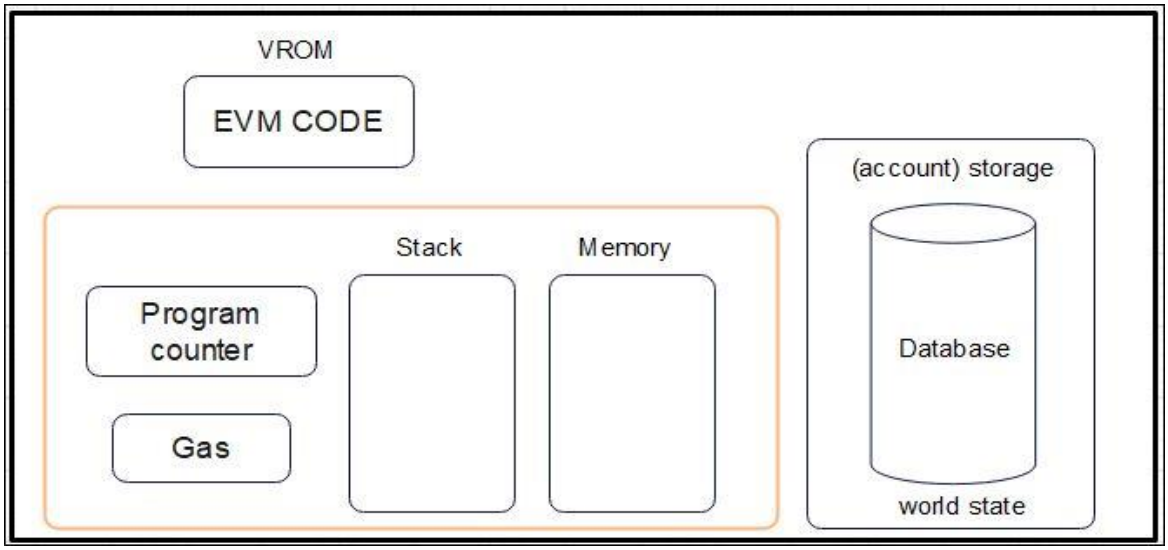

Figure 1. EVM's description.

The ownerless configuration aims to maximize uptime and security while limiting subterfuge incentives.

Research Contributions

Cloud computing has emerged as an essential technique for delivering infrastructure and data service objectives at a low cost, with minimum effort, and with a high standard of scalability. It has therefore been widely applied in numerous parts of the IT industry. However, information security issues continue to impede the development of cloud computing and must be addressed.

At the same time, based on the success of Bitcoin, blockchain has emerged as one of the most promising technologies. Bitcoin's core technology is the blockchain [8]. Moreover, blockchain has developed a critical technique for ensuring security, particularly integrity, authenticity, and secrecy. These benefits prompted us to attempt to provide a security model using blockchain. The study's goal was to determine the purpose of blockchain technology in cloud computing, as well as potential dangers and problems in the implementation of blockchain technology in the field of cloud computing. The goals are listed below:

- Identify the breadth of blockchain technology and its application in cloud computing;
- Assess the security implications of blockchain technology's use in cloud computing;
- Examine the most recent solutions in terms of security by ensuring the confidentiality, integrity, and authenticity of public information.

This paper is organized as follows: Section 2 provides a brief introduction to cloud computing and blockchain technology and outlines how Ethereum has evolved in managing data within the cloud; Section 3 provides a literature review and related work analysis based on previously conducted research; Section 4 presents a proposed system with results and discussion; Section 5 provides a conclusion to the work and gives direction for future work.

2. Cloud Computing and Blockchain Technology Background

A significant development trend in information technology is the rental of storage and computation services from third parties by many individuals and companies. With cloud technology, what was previously controlled independently now involves the participation of servers, frequently in an unknown place but instantly accessible wherever internet access is present. Today, the existence of a cloud service provider (CSP) overseeing the service is commonly assumed while using these internet services. A variety of variables can explain the current situation. There are considerable scale savings in the purchase and administration of IT resources, and large-scale CSPs may deliver services at lower rates than smaller companies. Nonetheless, many users have extra computing, storage, and network capacities in their systems and would be willing to rent these resources to other users in exchange for payment. According to market theory, the development of an infrastructure that allows the combination of supply and demand for IT services would result in the considerable possibility of producing economic value through the utilization of otherwise underutilized resources [2].

In contrast to centralized cloud storage, which requires transferring and storing duplicate entry files over the internet to a core data center located thousands of miles away, a decentralized cloud or edge-computing architecture attempts to solve the inefficiency issues of uploading, downloading, and storing to the limited storage capabilities of cloud servers. A decentralized design may improve cloud service security. To protect information privacy and covert theft from third parties, law enforcement, and foreign governments, files can be locally kept behind a security system in specified geographic zones, with access controlled. The attack surface is limited because data is not copied to third-party systems or other sites. Compliance with other requirements is also hastened since files and storage are within an organization's control [1]. This shift in the environment is evidenced by the growing focus of the research and development community on creating decentralized cloud storage (DCS) services, distinguished by the availability of many nodes that may be utilized to store resources in a decentralized way. Individual resources are split into shards and allocated (with replication to ensure availability) to different nodes in such systems. In some related research [1–4], blockchain is shown to be a secure and distributed ledger

that can assist in overcoming many of the difficulties associated with centralization. However, their main goal was to provide insights into the use of security services for current applications, highlight the innovative techniques currently used to offer numerous benefits, describe their challenges, and discuss how blockchain technology can resolve them.

2.1. Cloud Computing

Cloud computing provides an enticing and appealing computer service through resource sharing and virtualization techniques. In cloud computing concepts, security as a service is a new problem. End-users might benefit from a variety of application services provided by a company. The finest examples of application service providers are e-mail and web servers [9]. Cloud computing is a modern technology that offers advantages such as high scalability, dependability, flexibility, and dynamic properties. Cloud computing is cost-effective, and all IT areas are migrating to it. However, the considerable development and demand for cloud computing have resulted in significant concerns about its security and privacy, defined by the rules, regulations, and technologies required to secure information, applications, and the cloud computing infrastructure [3]. Data protection, transparency, and availability are the three pillars of traditional data security. However, ensuring the provenance of the data (where the data came from) is an issue in cloud settings. The rise of digitization has resulted in content inaccuracy and document collaboration challenges, with version management concerns consuming 83 percent of output [10]. Existing document version control systems are centralized and at risk from data loss, as seen by higher time utilization and incorrect document update procedures that allow modifications to a document without the awareness of other network operators [11]. Furthermore, using centralized systems, modifications to the document and upgrade history can be altered, putting the integrity of alterations and their upgrade history at risk. As a result, there is a need for an utterly secure and decentralized infrastructure for digital document edition management. In [11], the authors reviewed the use of blockchain in the cloud computing system. First, the idea of blockchain was briefly explored, along with its benefits and drawbacks. Then, using blockchain technology, the concept of cloud computing was briefly presented. According to the evaluation, the research on blockchain-based cloud platforms is still in the initial stages. One of the most pressing concerns affecting the researchers was access control. Communication between multi-party calculations disturbed networks and caused unexpected monetary loss to reward the data. The establishment of fake accounts also reduced the system's scalability.

2.2. Blockchain Ethereum Technology

Blockchain is a decentralized environmental system in which all nodes are individual in job execution but intricately connected in managing an accurate ledger through competition and cooperation. Blockchain technology is a rapidly growing security cryptography system that provides decentralized techniques that have superseded many current security implementations. The usage of blockchain technology has lately increased because of the degree of security it provides. Although blockchain uses a distributed database, data corruption is more complicated. According to the rules, the data is encrypted and processed through blockchain software machines [3].

Ethereum is a toolkit that allows us to build an economic software model, manage accounts, and ensure every essential exchange item, coin, or token is identical to every other token in any system. The core of this concept is to provide a blockchain platform that utilizes the development of a decentralized platform. In contrast, Bitcoin has no concept of accounts, resulting in transactions switching from one address to another, making it more susceptible to external threats, while Ethereum allows account transactions. Also, account creation in Ethereum plays a significant role in developing a platform, and the combination of these existing accounts is called the world state of the Ethereum blockchain. Ethereum is a state-machine platform that can change the state of the transaction in lieu of enabling users to have secured data transactions [12].

The Ethereum world state mentions existing accounts with their addresses. There are two types: externally owned accounts (EOA) and contract accounts. Users have full control of their accounts using EOA, and contract accounts are controlled by their unique smart contract code. Ethereum brings the blockchain idea to the next level by allowing users to establish financial contracts, known as smart contracts, within the system, in a matter of minutes [3].

The smart contract governs all interactions and transactions between members. It manages user registration, requests for the approval of new IPFS versions, document approval and rejection histories, and the registration of development teams or validators in the chain [13].

Ethereum transactions are responsible for every interaction between users and the Ethereum blockchain platform by locally assigning and creating private keys for processing in the blockchain network. These interactions between users and the Ethereum blockchain platform incurs a transaction cost, which is also called transaction fees. This transaction fee is measured in a unit called gas. Every transaction and instruction in the Ethereum virtual machine have a gas cost, which is predetermined. At the end of the transaction process, this gas cost is calculated to know the total transaction cost. These transactions are gathered by the Ethereum blockchain network to validate and add to a body of new blocks to be further inducted into the blockchain ledger [13].

Ethereum mining is a consensus mechanism responsible for pushing the block into the blockchain ledger. This mechanism helps in minimizing trust issues in the blockchain Ethereum network. The node in these networks goes through a cryptographic pattern known as mining.

The Ethereum Virtual Machine (EVM) is a global device that anybody may utilize for a modest charge in Ether. It is technically a machine made up of numerous different machines.

Ethereum Virtual Machine (EVM)

The term "distributed ledger" is frequently used to describe blockchains like Bitcoin, which allow a decentralized currency by utilizing core cryptographic techniques [14]. Although Ethereum does have its local cryptocurrency (Ether), it offers an even more powerful function: smart contracts. Ethereum is a major data structure that contains not just the accounts and amounts but also a machine state that may vary from block to block according to a defined set of rules and can conduct arbitrary bytecode. The EVM defines the rules for altering the state from block to block.

Smart contracts on Ethereum, accounts, transactions, and EVM are three essential components discussed in detail in [15]. The EVM is a stack-based design instead of a register-based architecture that offers a distinct execution environment to protect contract execution from external assaults and prevent bad contracts from impacting the existing system. Once calls to the contracts are received (by message calls or transactions), the EVM will first search and then load the contract code from the local database. Unlike conventional software with only one entry point (main ()), all public functions in smart contracts can be entry points.

2.3. Blockchain Technologies in Secure Cloud Computing

The purpose of blockchain technology in cloud computing is to minimize the possible risks and obstacles in using blockchain technology in the field of cloud computing. There are two approaches for integrating blockchain with cloud systems:

- Integrating blockchain with the cloud facilitates enterprise networks such as storage, replication, and access to transactional databases.
- Integrating security ideas into cloud tasks, user, and data management.

In [16], The article demonstrates the usage of blockchain technologies in developing secured cloud computing infrastructure. It is a linear data idea that is constantly updated, duplicated, and distributed to all nodes in the network. The hash key is used to build an interconnection between blocks, resulting in a chain of blocks or blockchains. The effective

use of hashing and proof of work increases the security of the blockchain. Another critical security protection is that blockchain is peer-to-peer in origin, requiring agreement from more than 50% of peers to approve any modification. The most recent development is the constantly evolving blockchain and the notion of smart contracts. They addressed blockchain categories as well as their implementations. A comparison of three blockchain platforms: Ethereum, Hyperledger, and R3 Corda. Blockchain is quickly growing in fintech, including financial services, insurance, online payments, and many others. However, they think that the continuing integration of blockchain in services for improving security, especially the availability and integrity of data on the cloud, has still to be investigated. This article invites researchers to use the platforms in extreme scenarios and examine how they may be used to improve data availability across a distributed ledger utilizing blockchain technology to provide a more secure system solution. It is a linear data idea that is constantly updated, duplicated, and distributed to all nodes in the network. Blockchain is quickly growing in fintech, including financial services, the insurance sector, online payments, and many others. Therefore, it was necessary to use the platforms in extreme scenarios and investigate how they may improve data availability across a distributed ledger utilizing blockchain technology.

2.4. Decentralized Cloud Storage

Decentralized, safe storage of data, high data availability, and effective storage resources are used. The suggested system is divided into four components, as illustrated in Figure 2.

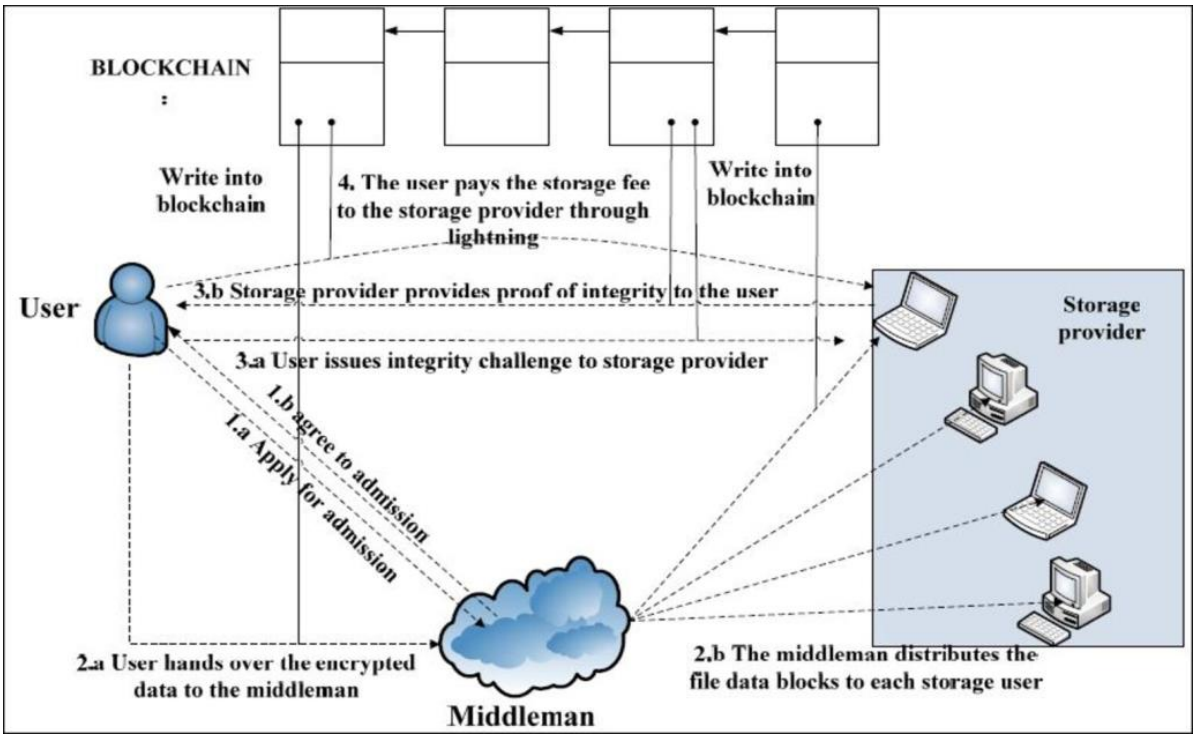

Figure 2. Decentralized cloud storage system design [10].

By encrypting and spreading data across various users in the system, the suggested approach improves data security. The implemented system encrypts data using the AES 256-bit encryption method, assuring the security of the user's data. The solution not only addresses the privacy and security problems associated with centralized cloud storage but also provides a platform for users to rent out unused storage and earn cryptocurrency in exchange, optimizing storage resource usage. Furthermore, a flexible scheduling method can be implemented, with which files can be viewed several times by the user instead of

only once. A payment system can also be introduced, with each member receiving a default credit of 100. Group members with greater credits are prioritized for data storage [10].

Cloud computing systems are popular for data sharing throughout many apps and network components. However, multiple copies of data follow different pathways to guarantee resilience, making it difficult for administrators to identify the origin of the assault, its impact, and its tool [3]. The primary foundations of blockchain architecture are a combination of cryptography mechanisms and distributed public ledgers. This mixture enables the creation of any type of structure on a blockchain without causing any trust issues on the network. The same holds for blockchain-enabled cloud solutions. Blockchain Ethereum benefits the cloud by ensuring data provenance (verifying the data's source) and enabling cloud monitoring. If genuine data provenance exists in the cloud, with all data gathered on cloud servers, distributed data calculations, data transfers, and transactions, it detects insider threats, replicates test findings, and identifies the specific source of the system or network breach [3].

In 2018, with the induction of the general data protection regulation (GDPR), every big organization must be compliant to preserve the privacy of the sensitive data of the customers. In a blockchain ecosystem, data sharing among users and service providers is overloaded and vulnerable. The GDPR compliance data sharing scheme ensures restricted and secured transactions of data using interaction tools between users and service providers. Also, to protect the data privacy of medical records, HIPAA (Health Insurance Portability and Accountability Act) was introduced. The main purpose of this compliance act was to protect the privacy and ownership of patient data [17,18].

3. Literature Review

Several existing studies on blockchain technology have lately gained popularity for cloud computing security. This section briefly discusses ongoing efforts to combine blockchain technology with a scalable cloud environment to increase trust, server service, data security, and user data management.

In [5], the authors proposed a decentralized service monitoring strategy based on the Ethereum blockchain. The suggested solution enabled consumers to assess CSP compliance with the contractual services under service-level agreements (SLAs) and "autonomously" reimbursed users in the event of security breaches. Simultaneously, the suggested method inhibited consumers from falsely reporting for financial advantage. The technique leverages the Ethereum blockchain architecture to implement security monitoring data and include SLAs as smart contracts. The implemented smart contract combined and abstracted measurable SLOs and monitored cloud services' compliance with contractual SLOs. Also, it autonomously recompensed the consumer in the event of a violation. The suggested monitoring method was tested on a commercial IaaS (Infrastructures as a Service) cloud service. The findings demonstrated that the technique is appropriate for assessing the values of SLOs and discovering violations of contractual SLO standards.

In [6], this study presented a hybrid cloud-blockchain system that guaranteed data integrity for all homomorphic encryption techniques. To obtain the cloud service provider's (CSP's) ultimate authority over the data, the suggested strategy used byzantine fault tolerance consensus to create a distributed network of processing CSPs based on the client's needs. After performing specific operations, all CSPs provided a master hash value for their database. In Bitcoin and Ethereum blockchain networks, master hash values were stored to ensure immutable data was produced. The master hash values can be acquired for verification by tracing the block header address. A theoretical analysis of the overhead costs of creating master hash values was offered for each cryptocurrency.

They discovered that Ethereum has lower client financial costs and superior online performance compared with Bitcoin. They also outlined the proposed scheme's data security requirements, ground-level implementation, and future work. The proposed technique uses homomorphic encryption to offer data security and privacy during outsourced computations.

The study described in [10] focused on decentralized and safe data storage, high availability, and effective storage resource use. By encrypting and spreading data across various users in the system, the suggested approach improved data security. The implemented system encrypted data using the AES 256bit encryption method, assuring the security of the user's data. The solution not only addressed the privacy and security problems associated with centralized cloud storage but also provided a platform for users to rent out unused storage and earn cryptocurrency in exchange, optimizing storage resource usage. Furthermore, a flexible scheduling method can be implemented, with which files can be viewed several times by the user as opposed to only once. A payment system can also be introduced, with each member receiving a default credit of 100. Group members with greater credits would be prioritized for data storage.

In [11], the authors reviewed the use of blockchain in the cloud computing system. First, the idea of blockchain was briefly explored, along with its benefits and drawbacks. Then, using blockchain technology, the concept of cloud computing was briefly presented. According to the evaluation, the research on blockchain-based cloud platforms was still in the initial stages. One of the most pressing concerns affecting the researchers was access control. To reward the data, communication between multi-party calculations disturbed networks and caused unanticipated monetary loss. The establishment of fake accounts also reduced the system's scalability. The intended model should attempt to address the difficulties mentioned above in the future.

The study in [12,13] provided a blockchain-based solution and platform for file sharing and changed the control to promote multi-user collaboration and monitor changes in a trustworthy, secure, and decentralized way without the intervention of a centralized trusted institution or third party. They developed and discussed their solution using the Ethereum blockchain and smart contracts to authorize, track, and conduct versioning operations for the IPFS-stored file. This solution eliminated the necessity for a trusted centralized authority and enabled transactions and files with high integrity, resilience, and security to exchange and monitor multiple versions of online content. Remix IDE was used to implement and test the smart contract. All procedures were tested to confirm that the concept and contract state were accurate. In addition, prominent security analytical techniques, such as ChainSecurity and Oyente, were used to evaluate and show the robustness and security of the built smart contract against widely known threats. In the future, developers can create smart contracts for various file management functions. In addition, developers may also create smart contracts for domain-specific decentralized data procedures.

In [16], the authors demonstrated the usage of blockchain technologies in developing secure cloud computing infrastructure. It was a linear data idea that was constantly updated, duplicated, and distributed to all nodes in the network. The hash key was used to build an interconnection between blocks, resulting in a chain of blocks or blockchains [19]. The effective use of hashing and proof of work increased the security of the blockchain. Another key security protection was that blockchain is peer-to-peer in origin, requiring agreement from more than 50% of peers to approve any modification. The most recent development is the constantly evolving blockchain and the notion of smart contracts. They addressed blockchain categories as well as their implementations and compared three blockchain platforms: Ethereum, Hyperledger, and R3 Corda. Blockchain is quickly growing in fintech, including financial services, the insurance sector, online payments, and many others. However, they thought that the continuing integration of blockchain in services for improving security, especially the availability and integrity of data on the cloud, was still to be investigated. This article invited researchers to use the platforms in extreme scenarios and examine how they may be used to improve data availability across a distributed ledger utilizing blockchain technology to provide a more secure system solution.

In the study [20], they looked at blockchain technology and related core technologies and the current state of research to see what else needs to be looked at. Several existing concerns must be considered in employing blockchain in a cloud computing environment. Several challenges arose with blockchains, such as transaction security, wallet security, and

software security, and various research has been undertaken to address these issues. As a result, this research presented a secure blockchain use and removal protocol to explore the method of providing security. Considering the environment in which a vast volume of data was sent, efficiency studies were also required in addition to security studies.

In [21], cloud computing and blockchain technologies were briefly discussed. The advantages of combining the blockchain network with a scalable cloud environment to improve trust, server service, data security, and user data management were explored. The authors tended to conduct a quick survey on previous research focused on blockchain merging with the cloud to demonstrate their influence. Architecture for integrating blockchain with the cloud was established in the article, showing communication between blockchain and the cloud. There were numerous benefits in terms of usability, trust, security, scalability, data management, and other factors if blockchain and cloud computing were combined. Perhaps data privacy and security problems can be solved by integrating blockchain technology to support cloud computing growth. The architecture of cloud computing and blockchain technology integration is depicted in Figure 3. The application layer allows the user to interact with the server. Assume that when a user requests a transaction via the application layer, the transaction details are saved by generating a block for each transaction.

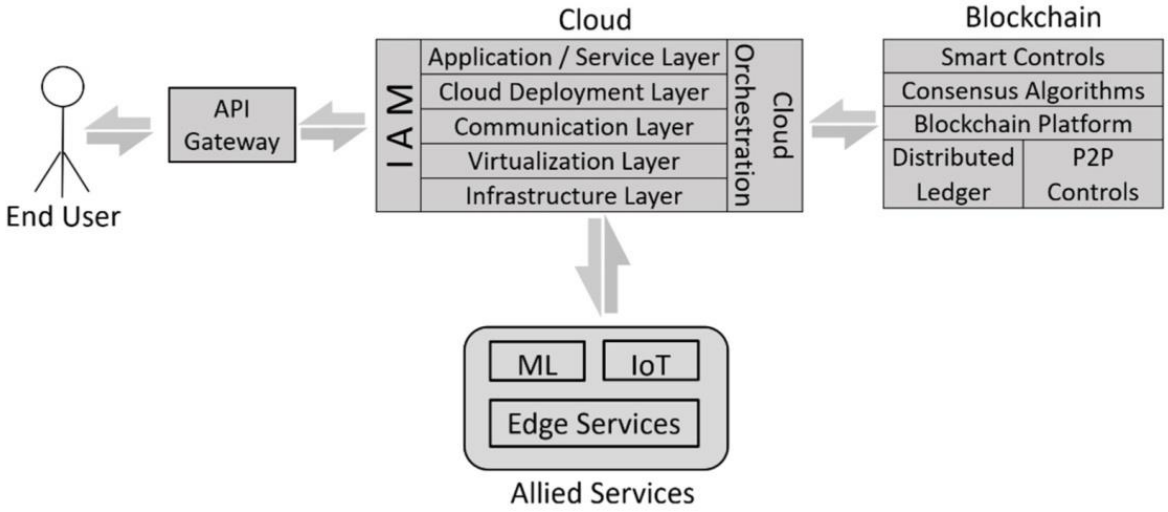

Figure 3. The architecture of cloud integrated with blockchain [20,21].

In [22,23], the authors examined the many security issues in blockchain and cloud computing and the use of blockchain in the security of cloud computing. All users employed blockchain technology linked to computer-generated and virtual money. However, several blockchain security issues were reported, including the following:

1. Agreement on the blockchain
2. Transactional protection
3. Security of the wallet
4. Security of software

Blockchain could be a useful and powerful tool for ensuring security in the cloud computing environment. In addition, this article examined the various blockchain implementations for cloud security that are now available.

In [24], the authors designed and built ProvChain, a system for collecting and verifying cloud data provenance by embedding provenance data into blockchain transactions. ProvChain works in three stages: (1) provenance data gathering, (2) provenance data storage, and (3) provenance data validation. Figure 4 depicts an overview of the ProvChain architecture. The following are the key components of ProvChain: Cloud User, Cloud Service Provider (CSP), Provenance Database, Provenance Auditor (PA), and Blockchain Network.

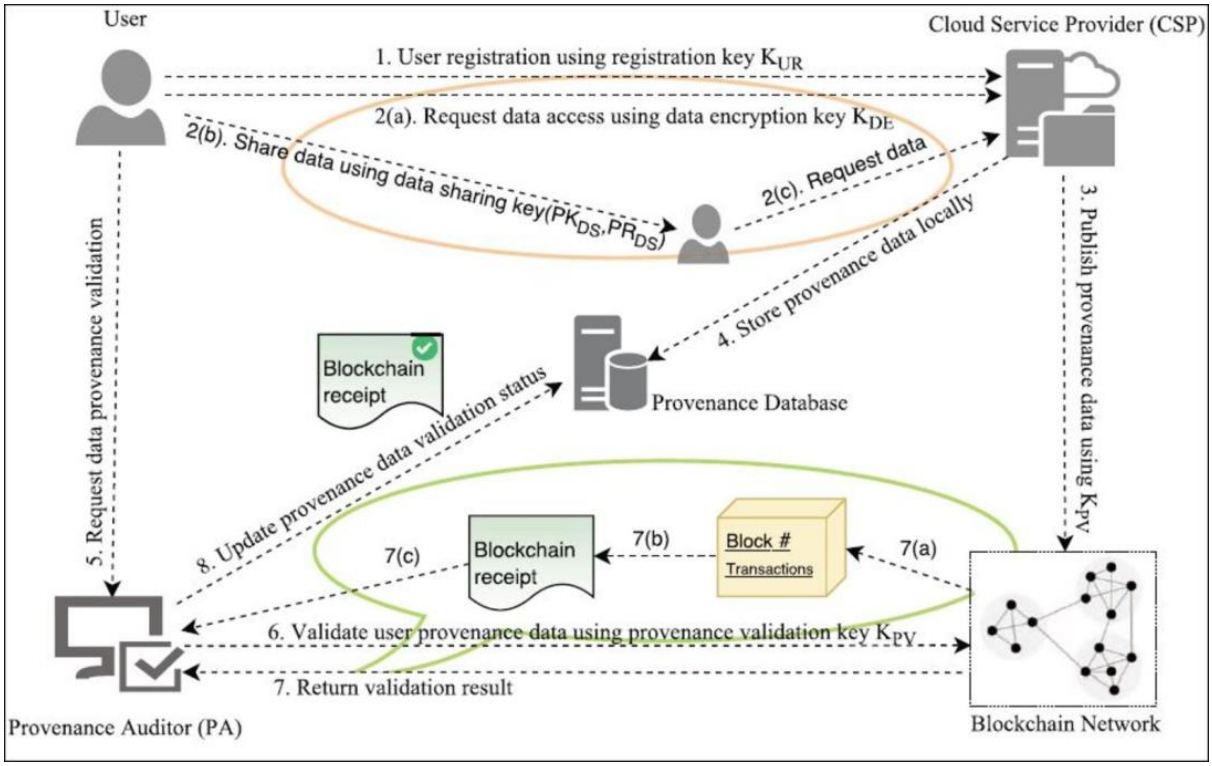

Figure 4. Interaction of the ProvChain System.

According to the performance study results, ProvChain delivers security characteristics such as tamper-proof provenance, user privacy, and dependability with little overhead for cloud storage applications. In the future, the validation will be built on top of an open-source architecture, which will enhance overall efficiency, security, and flexibility.

Table 1 summarizes the related studies that applied blockchain technologies in developing secure cloud computing. Specifically, we discuss the studies currently used for decentralized service monitoring strategy based on the Ethereum blockchain to provide a more secure system solution.

Table 1. Summary of related work.

Ref	Purpose or Motivation	Methodology	Result	Limitations and Future Directions
[5]	Propose a decentralized service monitoring strategy based on the Ethereum blockchain.	The implemented smart contract combined and abstracted measurable SLOs and monitored cloud services compliance with contractual SLOs.	The findings demonstrated that the technique was appropriate for assessing the values of SLOs and discovering violations of contractual SLO standards.	
[6]	Present a hybrid cloud-blockchain system.	The design was built on CSP and BC technologies, which were equally important.	The proposed technique used homomorphic encryption to offer data security and privacy during outsourced computations.	

Table 1. *Cont.*

Ref	Purpose or Motivation	Methodology	Result	Limitations and Future Directions
[10]	Decentralize safe storage of data.	Encrypted data using the AES 256-bit encryption method.	Privacy and security problems associated with centralized cloud storage also provided a platform for users to rent out their unused storage and earn cryptocurrency.	Flexible scheduling method, in which files can be viewed several times and a payment system.
[11]	Determine the purpose of blockchain technology in the field of cloud computing.		Integrated system to ensure and improve confidence between the cloud servers, data consumers, and data security.	The planned model should address access control; the connection between multilateral accounts broke the networks and caused unexpected economic loss.
[12]	Provide a blockchain-based solution and platform for file decentralized way.	Used the Ethereum blockchain and smart contracts to authorize, track, and conduct versioning operations for the IPFS-stored file.	Enabled transactions and files with high integrity, resilience, and security to exchange and monitor multiple versions of online content.	Developers were able to create smart contracts for various file management functions. Create smart contracts for domain-specific decentralized data procedures.
[15]	Demonstrate the usage of blockchain technologies in developing secure cloud computing infrastructure.		Blockchain is quickly growing in fintech, such as financial services, the insurance sector, online payments, and many others.	Used the platforms in extreme scenarios and investigated how they may improve data availability across a distributed ledger utilizing blockchain technology.
[17]	Blockchain technology and related core technologies.	Explored the method of providing security.	Presented a way of securing blockchain use and removal protocol to explore the method of providing security.	Considering the environment in which a vast volume of data is sent, efficiency studies were also required in addition to security studies.
[18]	Cloud computing and blockchain technologies were briefly discussed	Established an architecture that integrated blockchain with the cloud, disclosing the connection between the two.	There were numerous benefits in terms of usability, trust, security, scalability, data management, and other factors if blockchain and cloud computing were combined.	We can solve data privacy and security problems by integrating blockchain technology to support cloud computing growth.
[19]	Examine the many issues of security in blockchain and cloud computing.		Blockchain could be a useful and powerful tool for ensuring security in the cloud computing environment.	
[20]	Collect and verify cloud data provenance by embedding provenance data into blockchain transactions.	Designed and built ProvChain, a system.	ProvChain delivered security characteristics such as tamper-proof provenance, user privacy, and dependability with little overhead for cloud storage applications.	Built ProvChain on top of an open-source architecture.

Table 1. *Cont.*

Ref	Purpose or Motivation	Methodology	Result	Limitations and Future Directions
[25]	VerfiyMed: A blockchain platform for transparent trust in virtualized healthcare: proof of concept.	Design, implementation, and evaluation of VerifyMed.	A contemporary solution for the healthcare domain to incorporate a robust blockchain platform for transparent trust.	Open source with limited trust boundaries.
[26]	A blockchain-based framework to enhance anonymous services with accountability guarantees.	Proposed a framework to deal with an identification issue of unauthorized access of anonymous entities.	Strengthened access control system, a private key and user credentials were generated using secured third-party protocols, which enabled the framework to utilize group signatures.	Real-life implementation issues due to lack of validation and industrial support.

Based on our review, we can derive the following observations:

- The current impediment to providing a model of secure storing data in cloud computing by utilizing blockchain Ethereum is, according to the assessment, still in the initial stages.
- One of the primary challenges that researchers confront in developing a viable system for assessment is access control.
- To reward the data, transmission between multi-party calculations interrupts networks and causes unanticipated monetary loss. The creation of false accounts also reduces the system's scalability.
- A need for a secure and workable model to mitigate real-life complications for uninterrupted services.

Based on the linked works in [24], it was proposed to use blockchain technology to create a decentralized and trustworthy cloud data provenance architecture. ProvChain is a data provenance collection and verification framework. It failed to create a trustworthy atmosphere. Overhead requires high computational complexity as the file size grows. The article [27] showed that a decentralized environment was met by integrating blockchain and edge computing platforms. Edge computing mobility brings cloud services and resources to the edge for denial of service. During scalability analysis, it incurred significant overheads. The study [24] examined blockchain technology as well as potential solutions for permanently distributed ledgers in operations and supply chains. Performance and data privacy were insufficient to enable blockchain transactions.

Furthermore, in [16], they demonstrated the usage of blockchain technologies in the development of secure cloud computing infrastructure. It was a linear data idea that was constantly updated, duplicated, and distributed to all nodes in the network. Blockchain is quickly growing in fintech, such as financial services, the insurance sector, online payments, and many others. Therefore, it was necessary to use the platforms in extreme scenarios and investigate how they may improve data availability across a distributed ledger utilizing blockchain technology.

By introducing our proposal, we plan to give a far more solid solution to gaps by providing or modifying models for secure storage data using blockchain Ethereum. The designed model for resolving the issues ensures data security and integrity in cloud computing. Ethereum is a decentralized network blockchain that utilizes the Ether currency and operates on the PoW (Ethash) compromise mechanism. At the same time, Hyperledger is a statement that does not use any currency and operates on the PBFT (excluding Corda) excluding mechanism. Furthermore, the Hyperledger blockchain is powered by both validating and non-validating peers.

The article [28,29] concentrated on decentralized, secure storage of data, high data availability, and optimal storage resource use. The proposed approach associated the cus-

tomer's wallet address with the user's file, allowing only the file's rightful owner to access the contents. The information of customers was saved in the Ethereum blockchain [30–32]. The Ethereum blockchain network supported the usage of smart contracts, which stored information from files provided by users on the blockchain. Every time data was uploaded or downloaded, the suggested system encrypted and decrypted it. The system used the IPFS protocol to distribute files effectively across several network peers [33]. An adaptive scheduling technique may be used, with which files can be visited more frequently by the user than those only infrequently accessed. This guaranteed that commonly accessed files were readily available to the user whenever needed. A credit system can also be introduced. Each peer was allocated a default 100 credit depending on their system uptime, and multiple successfully served file access requests had their credits withdrawn or added. Peers with greater credits were prioritized for data storage. The most important gap is the little research on this topic. In addition, the available research is new and has some gaps. That is, it needs analysis, development, and the acquisition of reviews.

4. Proposed System

4.1. EVM-ECC Model

The primary issue when storing data in cloud computing for security, privacy, and decentralization. The Ethereum blockchain supports the use of smart contracts, which store data about files provided by users on the blockchain. The suggested system encrypts and decrypts data that is uploaded or downloaded. The proposed method is divided into parts. The client first registers for a MetaMask account. The program retrieves the user's account address and wallet amount from MetaMask using web3.js [26]. The file to upload is selected by the user using the file choice. Next, the system determines the number of available peers. Before that, the ECC method encrypts the extracted data using the user's public key as the identifier. The ECC (elliptic curve cryptography) method is used to improve the security of user data saved in data storage. Using the IPFS protocol, the user's data is distributed across accessible peers. IPFS then delivers a hash value containing the file's path. The concept is explored as follows:

- MetaMask is a cryptocurrency wallet that uses software to interface with the Ethereum network. It gives users access to their Ethereum wallet via a browser extension or mobile app, which they can connect with decentralized apps.
- Ethereum is a smart contract with a currency, Ether (ETH), and a programming language, Solidity. Ethereum, as a blockchain network, is a decentralized public ledger used for transaction verification and recording.
- The Interplanetary File System (IPFS) is a protocol and peer-to-peer network that allows data to be stored and shared in a distributed file system. IPFS employs data to identify each file in a facility that connects various computing devices.
- Elliptic-curve cryptography (ECC) is a public-key encryption technique based on the integral equation of elliptic curves over finite fields. Because ECC employs fewer keys and signatures than RSA for the same degree of security and enables rapid key generation, key agreement, and signatures, it is considered a natural contemporary successor to the RSA cryptosystem.

4.1.1. Architecture

Figure 5 depicts the general framework of the proposed model's design.

Phase 1: Encryption and Decryption

The encryption phase aims to encrypt the user's document by using ECC. For existing cryptography algorithms, an elliptic curve is a plane curve beyond a finite field made up of points that meet Equation (1):

$$y^3 = x^3 + ax + b \quad (1)$$

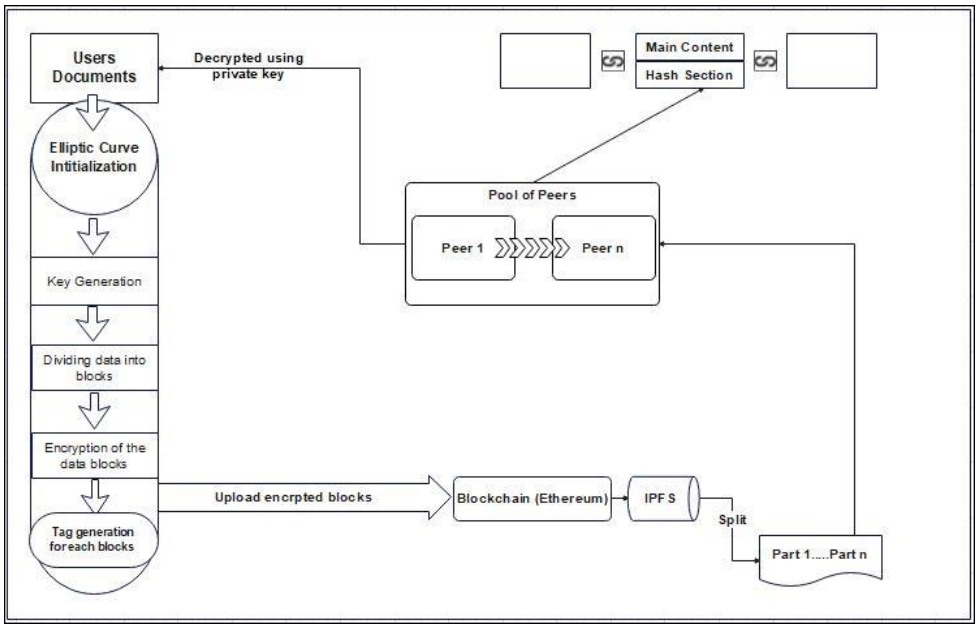

Figure 5. EVM-ECC Architecture.

This equation, along with the group operation of elliptic curves, forms an algebraic group, with the object at infinity serving as an identity member. The group's architecture is obtained from the underpinning algebraic variety's component group [27].

$$Div^0(E) \rightarrow Pic^0(E) \cong E \tag{2}$$

Equation (2) provides key generation as well as encryption and decryption techniques, along with pertinent data [28].

1. Key Generation: This is the most crucial step in which an algorithm generates both public and private keys

$$Private\ Key\ \leftarrow d_A Q_A \tag{3}$$

i. Encryption: (Input: Document, Output: Encrypted blocks)

- Choose the document to be uploaded
- Divide documents into data blocks.
- Encrypt each block with a secret key

$$S_k = d_A Q_B = d_B Q_A \tag{4}$$

- Generate the tag index for each block.
- Upload the data into the blockchain Ethereum.

ii. Decryption: (Input: Encrypted Document, Output: Decrypted Document

- Select the document from Ethereum storage by the pool of peers
- Retrieve all blocks of that file from the pool of peers(miners).
- Decrypt each block with the secret key S_k
- Combine all blocks and download them as a single file.

2. Phase 2: Uploading Document

The user first registers for a MetaMask account. The program retrieves the user's account address and wallet amount from MetaMask using web3.js. The file to upload is selected by the user using the file picker. The system will determine the pool of available peers. Next, the system verifies the file size and data storage availability. When adequate

storage space allows, the file is uploaded. When sufficient storage is unavailable, users are alerted to try again.

- There is enough space on the network to store files.
- The user has a sufficient wallet balance to pay the peers.

3. Phase 3: Document storage across several peers

After that, the encrypted file is split into 64 KB parts and sent to multiple peers throughout the network using the IPFS protocol. To allow registered peers to store the document in the network, the proposed model uses a private IPFS network. Using the IPFS cluster, the file block is duplicated on several peer storages for high availability. IPFS produces a hash value indicating the file's path. The hash value, together with information, is associated with the user's wallet address and saved in the blockchain via a smart contract. Smart contracts are like agreements in that they are used to eliminate the requirement for a third party. Under specific settings, they control transactions between nodes or assets between parties. This is a set of code lines stored on a blockchain network and automatically executed when certain terms and circumstances are achieved.

4.2. Ethereum

Defined blockchains, such as Ethereum, include much functionality and save time and energy in integration testing. Furthermore, from a cyber-security standpoint, because the Ethereum platform has a lot of miners' distributed networks, the security measures will be strong, which is a great reason to use it.

4.2.1. Remix: Smart Contracts Editor

Remix is a sophisticated open-source application that allows you to create code in the Solidity programming language from the browser. Remix was developed in JavaScript and may be used in the browser and locally. Remix also offers smart contract testing, debugging, and deployment, among other things [29].

Below is a summary of some defining attributes:

- It shows source files as tabs;
- In the garbage, compilation warnings and error messages are shown;
- Remix constantly stores the existing file (5s after the last changes)

```
pragma solidity >-0.7.0 <0.9.0;
/**
* @title Storage
* @dev Store & retrieve value in a variable
*/
contract Storage {
unit256 number;
/**
* @dev Storage value in variable
* @param num value to store
*/
function store(uint256 num) public {
number - numb;
}

/**
* @dev Return value
* @return value of 'number'
*/
function retrieve() public view returns (uint256){
return number;
}
}
```

4.2.2. Solidity: Smart Contracts Language

Solidity is a language with curly brackets. It is inspired by C++, Python, and JavaScript and is intended for use with the Ethereum Virtual Machine (EVM). The run code has no access to other processes on the machine and has restricted access to other smart contracts. Contracts produced in Solidity are analogous to objects defined in object-oriented languages such as JavaScript [34,35]

4.3. Proposed System Model

Figure 6 depicts the processes of the suggested system for storage encryption documents on blockchain Ethereum. The flowchart depicts three fundamental system steps: encryption by ECC, upload of documents on the blockchain Ethereum and split and store by IPFS on a pool of peers.

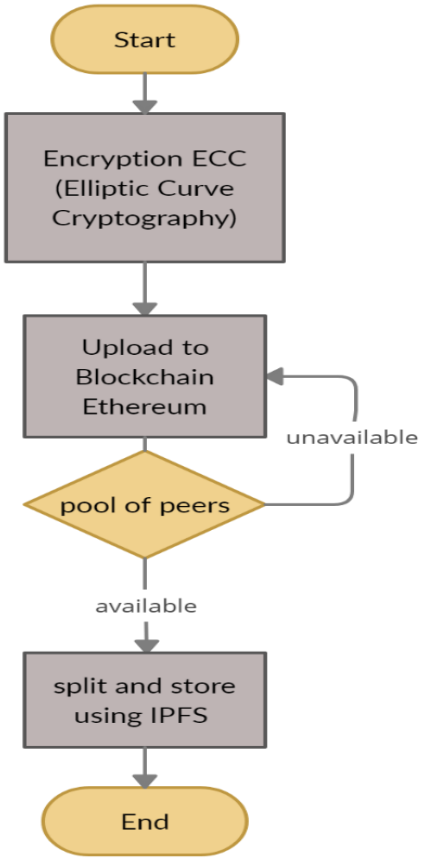

Figure 6. Flow chart of the proposed model.

5. Implementation and Results

During the implementation phase, encryption and decryption using the ECC algorithm created a smart contract utilizing the Solidity programming language, Remix as the editor, and MetaMask to establish a wallet for deploying smart contracts and other transactions in the Ethereum network.

5.1. Encryption and Decryption Using ECC Algorithm

Work environment: Microsoft visual studio.
Used Libraries: using nsoftware.IPWorksEncrypt
using Microsoft.Office.Interop.Word.

1. Work mechanism: Generate two keys, the pair (public and private keys) of the sender: generated and saved at the beginning for use each time the text will be encrypted or

decrypted (the public key will be used to encrypt the file and the private key will be used to decrypt the file).

Private Key:string ecc1_prive = ecc1.Key.PrivateKey;
Public Key: string ecc1_publ = ecc1.Key.PublicKey;

2. Reading the file will be performed by creating a file dialog to choose a file to be encrypted (the file in .docx format).

```
OpenFileDialog file = newOpenFileDialog();
file.ShowDialog();
tring path = file.FileName.ToString();
Microsoft.Office.Interop.Word.Application app = new
Microsoft.Office.Interop.Word.Application();
Document doc = app.Documents.Open(path);
Encryption is done by using the public key:
ecc1.InputMessage = textBox1.Text;
ecc1.RecipientKey.PublicKey = ecc2_pub;
ecc1.UseHex = true;
ecc1.Encrypt();
```

3. The encrypted file is saved as a text file, where its content can only be obtained by knowing the private key corresponding to the public key that was used to encrypt.

```
Microsoft.Office.Interop.Word.Application app1 = new
Microsoft.Office.Interop.Word.Application();
Microsoft.Office.Interop.Word.Documentdocc = app1.Documents.Add();
Microsoft.Office.Interop.Word.Paragraph para;
para = docc.Paragraphs.Add();
para.Range.Text = textBox2.Text;
docc.SaveAs2("c:\\Users\\public\\Desktop\\enc1.docx");
docc.Close();
```

4. For decryption, the encrypted file is selected like in the previous stage via file dialog, and then decryption is done by using the private key.

```
OpenFileDialog file = newOpenFileDialog();
file.ShowDialog();
string path = file.FileName.ToString();
Microsoft.Office.Interop.Word.Application app = new
Microsoft.Office.Interop.Word.Application();
Document doc = app.Documents.Open(path);
x = doc.Content.Text;:
ecc2.Key.PrivateKey = ecc2_priv;
ecc2.InputMessage = x;
ecc2.UseHex = true;
ecc2.Decrypt();
```

5.2. Create the Smart Contract

Work environment: Solidity on Remix.

1. Define The Version: pragmasolidity^0.4.17;
2. Define two variables to identify the account that sent the request and the hash of each piece of file.

```
string ipfsHash;
addressprivate owner;
```

3. Define the constructer to set the account: Create an address owner to store the address of the person who deployed the smart contract to Ethereum. When the constructor runs, it may utilize the owner's address in the modifier.

```
constructor()public{
owner=msg.sender;
}
```

4. Define three functions to receive the account balance and to send and receive hashes.

```
function getBalance()publicviewreturns(uint256){
 return owner.balance;
 }
 function sendHash(string x)public{
 ipfsHash = x;
 }
 function getHash()publicviewreturns(string x){
 return ipfsHash;
 }
```

5. Compile using Remix by clicking the deploy button after developing a smart contract; it can be tested in a simulated environment. The last step is deploying the smart contract on the web3 provider. After compiling it without errors, we can call the smart contract MetaMsk to be automatically opened to accept the transaction. Then, we will save the contract address and Application Binary Interface (ABI).

5.3. Metamask and Ganache (Create Address and Get ETH)

Ganache is a tool for deploying contracts, developing apps, and running tests on your own Ethereum BC. It makes it possible to complete all necessary tasks across the main chain. It has useful features, including enhanced mining controls and a built-in block explorer, and can either be imported or created from a new wallet in MetaMask. It may be possible to test the distributed application on a test network and receive some ETH that works within the test network for testing purposes. We used a Ganache account to connect to MetaMask, providing a virtual network with free gas to use our contract.

1. Open Ganache to get an account: Figure 7 shows the accounts and private key to use a Ganache account.

Figure 7. Available accounts to Ganache.

2. Open MetaMask and then choose to import an account and input one private key from an account on Ganache, as shown in Figure 8:

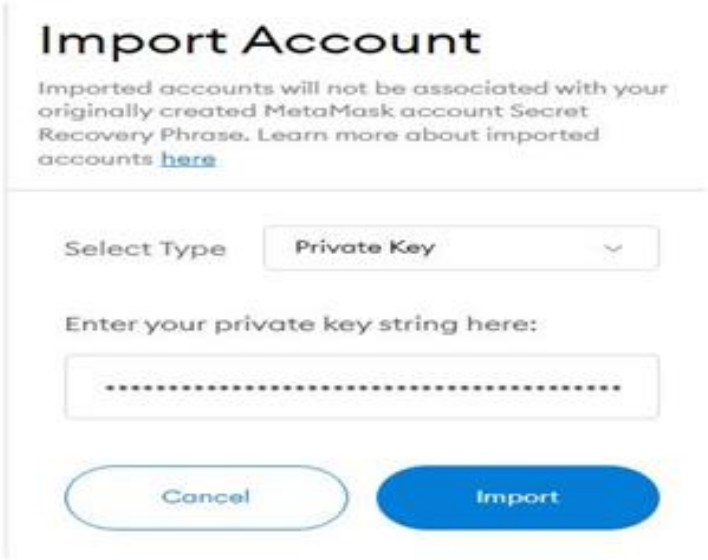

Figure 8. Import account on MetaMask.

3. 100 ETH added to our account, as shown in Figure 9:

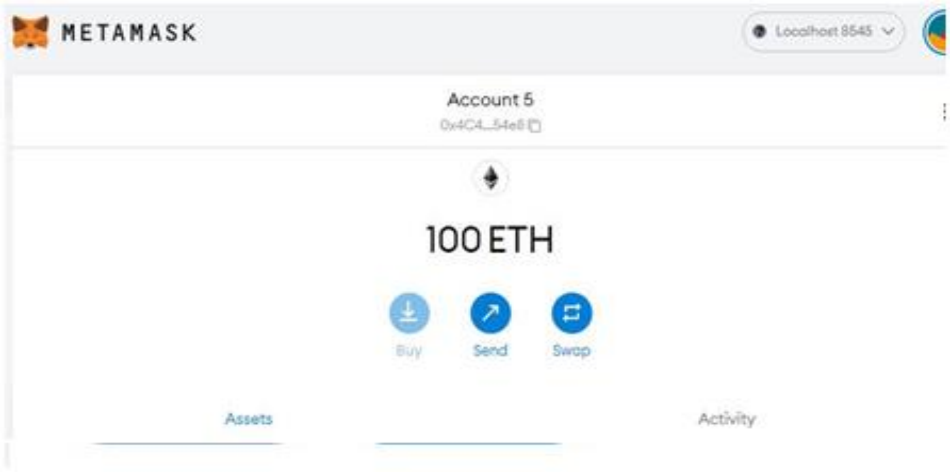

Figure 9. Metamask account with 100ETH.

5.4. Uploading Cithered File on Ethereum Using IPFS

Environment: NodeJS, React.
Used libraries:

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';
import web3 from './web3';
```

1. The first step was to install the dependencies (using npm on NodeJS) for this library, which was needed to complete our project work.

```
npm I create-react-app
npm install react-bootstrap
npm install fs-extra
npm install ipfs-api
```

```
npm installweb3@^1.0.0-beta.26
npm install –g create-react-app
const web3 = new Web3(window.web3.currentProvider);
```

In the directory we needed to install a react app and allow users to use async and await instead of adding promises to our JavaScript.

2. Next, we defined our smart contract address and ABI threat and we protected them against before developing our smart contract for use.

```
const address = 'our address';
abi = [our abi];
```

3. Building the application:

- Set the state variables:

```
State = {
ipfsHash:null,
buffer:'',
ethAddress:''
blockNumber:'',
transactionHash:'',
gasUsed:'',
txReceipt: ''
};
```

- Set the state variables:

```
captureFile =(event) => {
event.stopPropagation()
event.preventDefault()
const file = event.target.files [0]
let reader = new window.FileReader()
reader.readAsArrayBuffer(file)
reader.onloadend = () =>this.convertToBuffer(reader)
};
convertToBuffer = async(reader) => {
//file is converted to a buffer for upload to IPFS
const buffer = await Buffer.from(reader.result);
//set this buffer -using es6 syntax
this.setState({buffer});
};
```

- Uploading the buffered file to IPFS to block it and get block hash:

```
this.setState({blockNumber:"waiting . . . "});
this.setState({gasUsed:"waiting . . . "});
```

- Getting the user account and connecting it with the smart contract:

```
//bring in user's metamask account address
constaccounts =awaitweb3.eth.getAccounts();
//obtain contract address from storehash.js
constethAddress=awaitstorehash.options.address;
```

- Sending the file to IPFS and generating hashes:

```
awaitipfs.add(this.state.buffer, (err, ipfsHash) => {
console.log(err,ipfsHash);
//setState by setting ipfsHash to ipfsHash [0].hash
this.setState({ ipfsHash:ipfsHash [0].hash });
```

- Sending the file blocks to Ethereum:

```
storehash.methods.sendHash(this.state.ipfsHash).send({
from: accounts [0]
}, (error, transactionHash) => {
console.log(transactionHash);
this.setState({transactionHash});
});
```

5.5. Results

This section presents and explains the results. Service delivery over permissioned and private infrastructures is now possible due to advancements in BC-based technologies. Consequently, businesses or groups of people can now share information and services even if they do not trust each other to assist them with infrastructure management responsibilities [24].

File encryption ensures the confidentiality of data, protecting it from being accessed by people who are not permitted to do so. Even if an attacker manages to obtain data, they will not be able to understand its content without decrypting it, and encryption cannot be decrypted unless the sender's private key is known, as shown in Figure 10.

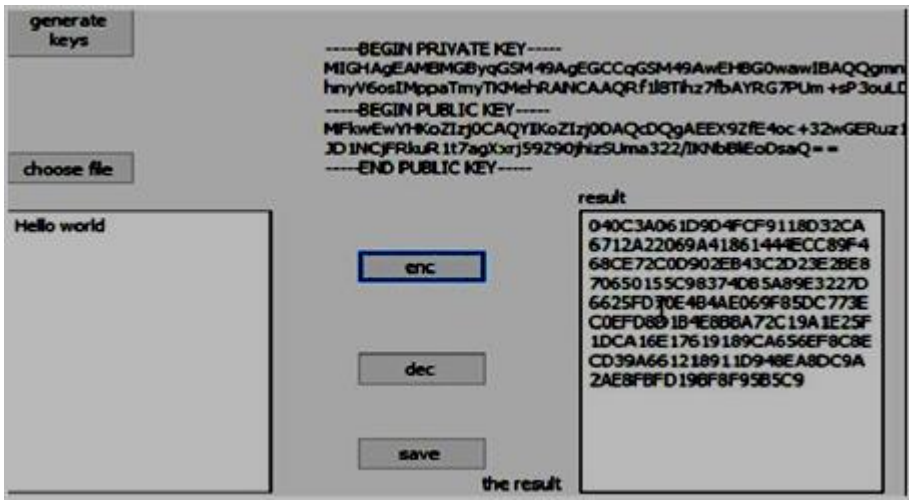

Figure 10. Encrypted file.

To utilize the system, users must first create an account on MetaMask and then log in using their credentials. After a successful log-in, users are then sent to the home screen, where they may choose which file to upload, as shown in Figure 11.

Figure 11. Uploading file by IPFS.

Based on the specified file size, the system will run checks to determine the available storage. Once enough storage has been made available, it would then be possible to upload a file encrypted using the ECC algorithm. The overall cost of storing this file would then be calculated by the system, which will be used to verify if the user's wallet balance is greater than the computed amount after the cost has been determined, as shown in Figure 12. If the user has sufficient funds, they will be prompted to pay cryptocurrency to store the file.

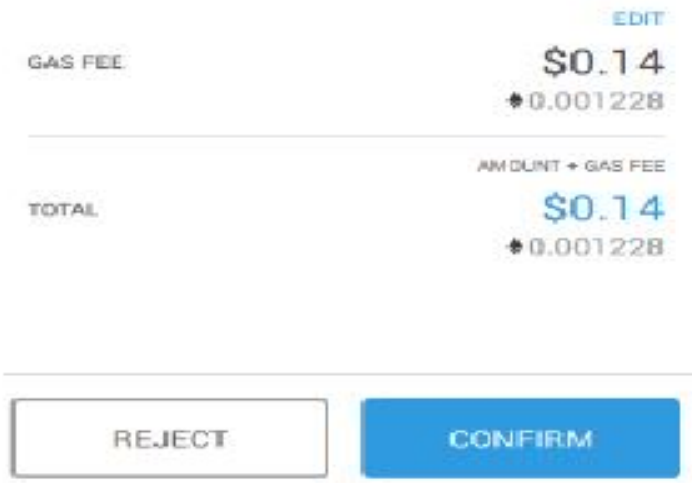

Figure 12. System verifies user's wallet balance.

We separately generated the hash for each block and uploaded the hashes to Ethereum because each upload incurred a fee. In addition to the fact that the hash always provides the best representation of the current file with a fixed length, by uploading the hash, we ensured that the file was not tampered with because changing any characteristics of the file would lead to significant changes in the hashtags.

Following successful payment, the file is divided into blocks and stored across peers using the IPFS protocol, with the hash value kept in the BC. Each Ethereum block is then saved to the server so that no attacker will be able to obtain the full file because they do not know where the file is being distributed.

6. Conclusions

Cloud computing systems are popular for data sharing throughout many apps and network components. However, multiple copies of data follow different pathways to guarantee resilience, making it difficult for administrators to identify the origin of the assault, its impact, and its tool. The primary foundations of blockchain architecture are a combination of cryptography mechanisms and distributed public ledgers. This mixture enables the creation of any type of structure on a blockchain without causing any trust issues on the network. The same holds for blockchain-enabled cloud solutions. The blockchain Ethereum benefits the cloud by ensuring data provenance (verifying the data's source) and enabling cloud monitoring. If genuine data provenance exists in the cloud, with all data gathered on cloud servers, distributed data calculations, data transfers, and transactions, it would detect insider threats, replicate test findings, and identify the specific source of the system or network breach. Blockchain has developed a critical technique for ensuring security, particularly integrity, authenticity, and secrecy. These benefits prompted us to attempt to provide a security model using blockchain. We proposed a model for data storage security in cloud computing by using the blockchain Ethereum.

The developed system encrypted data using the ECC encryption algorithm, assuring the secrecy of the user's data. The IPFS protocol distributed and stored this encrypted data among network peers. The model addressed privacy and security issues associated with centralized cloud storage. Moreover, it provided a platform for peers to rent out underused

storage and earn currency in the form of exchange, optimizing storage resource usage. The gap addressed was the amount of time taken to upload files, as determined by file size and the availability of peers. The time taken to upload a file grew proportional to its size. This proposed system provided a distributed model for a cloud access control system that employs role-based access control. For available resources, we leveraged ECC encryption and Ethereum BC smart contracts. This then improved data security by encrypting and distributing data across different peers in the system.

As a part of future work, a dynamic scheduling mechanism might be implemented to enable files to be viewed several times by a user rather than supporting one-time access only. This will make it easier for the user to access commonly used files as needed. When developing a model, it is highly beneficial to establish a local Ethereum BC in which the system can be tested in a safe environment. This is precisely what Ganache does, which is why it is vital to consider what Ganache is and the reasons for requiring a local BC in future studies. Ganache is a worldwide software platform with no host on which developers build thousands of BC-based applications.

Author Contributions: N.K. and M.T. conducted this research as part of their research project. N.K. and M.T. were involved in planning the work, performing the analysis, drafting the manuscript, and designing the figures. H.A. assisted in collecting the related article and drafting the paper. A.F., T.S. and Y.H.T. participated as external team members to moderate and guide the research toward positive outcomes. The case study, analysis, and discussion were written by all participants. All authors discussed the results and commented on the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the Faculty of Engineering, and Research Management Center of Universiti Malaysia Sabah.

Data Availability Statement: This research is solely conducted based on published data from different online platforms. No individual or person was affected in the procedure.

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

References

1. Atieh, A.T. The Next Generation Cloud technologies: A Review On Distributed Cloud, Fog And Edge Computing and Their Opportunities and Challenges. *Res. Rev. Sci. Technol.* **2021**, *1*, 1–15. Available online: https://researchberg.com/ (accessed on 5 March 2022).
2. Bacis, E.; Vimercati, S.D.C.D.; Foresti, S.; Paraboschi, S.; Rosa, M.; Samarati, P. Securing Resources in Decentralized Cloud Storage. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 286–298. [CrossRef]
3. Baalamurugan, K.M.; Kumar, S.R.; Kumar, A.; Kumar, V.; Padmanaban, S. *Padmanaban, Blockchain Security in Cloud Computing*; Springer: Cham, Switzerland, 2022.
4. Kumar, A.; Abhishek, K.; Nerurkar, P.; Ghalib, M.R.; Shankar, A.; Cheng, X. Secure smart contracts for cloud-based manufacturing using Ethereum blockchain. *Trans. Emerg. Telecommun. Technol.* **2020**, *33*, e4129. [CrossRef]
5. Taha, A.; Zakaria, A.; Kim, D.; Suri, N. Decentralized Runtime Monitoring Approach Relying on the Ethereum Blockchain Infrastructure. In Proceedings of the 2020 IEEE International Conference on Cloud Engineering (IC2E), Sydney, NSW, Australia, 21–24 April 2020; pp. 134–143. [CrossRef]
6. Awadallah, R.; Samsudin, A.; Teh, J.S.; Almazrooie, M. An Integrated Architecture for Maintaining Security in Cloud Computing Based on Blockchain. *IEEE Access* **2021**, *9*, 69513–69526. [CrossRef]
7. Dannen, C. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*; Apress: Berkeley, CA, USA, 2017; Volume 318.
8. Goldin, P. *State of the Cloud Report: DevOps Trends*; Ferreira, D.M.P., Ed.; RightScale Inc.: Santa Barbara, CA, USA, 2021; pp. 1–19.
9. Monti, M.; Rasmussen, S. RAIN: A Bio-Inspired Communication and Data Storage Infrastructure. *Artif. Life* **2017**, *23*, 552–557. [CrossRef] [PubMed]
10. Zhu, Y.; Lv, C.; Zeng, Z.; Wang, J.; Pei, B. Blockchain-based Decentralized Storage Scheme. *J. Phys. Conf. Ser.* **2019**, *1237*, 042008. [CrossRef]
11. Sarmah, S.S. Application of Block chain in Cloud Computing. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 4698–4704. [CrossRef]
12. Nedakovic, A. Analysis and Improvements of VerifyMed-The Blockchain Solution for Virtualized Healthcare Trust Relations.Security and Cloud Computing (SECCLO). Master's Thesis, Aalto University, Espoo, Finland, 2022.
13. Nizamuddin, N.; Salah, K.; Azad, M.A.; Arshad, J.; Rehman, M. Decentralized document version control using ethereum blockchain and IPFS. *Comput. Electr. Eng.* **2019**, *76*, 183–197. [CrossRef]

14. Chacon, S.; Straub, B. *Pro Git Book*, 2nd ed.; Apress: Berkeley, CA, USA, 2014.
15. Available online: https://ethereum.org/en/developers/docs/evm/ (accessed on 12 October 2021).
16. Sharma, S.G.; Ahuja, L.; Goyal, D.P. Building Secure Infrastructure for Cloud Computing Using Blockchain. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; pp. 1985–1988. [CrossRef]
17. Piao, Y.; Ye, K.; Cui, X. A Data Sharing Scheme for GDPR-Compliance Based on Consortium Blockchain. *Future Internet* **2021**, *13*, 217. [CrossRef]
18. Hasselgren, A.; Wan, P.K.; Horn, M.; Kralevska, K.; Gligoroski, D. GDPR Compliance for Blockchain Applications in Healthcare. In Proceedings of the International Conference on Big Data, IOT and Blockchain (BIBC 2020), Dubai, United Arab Emirates, 24–25 October 2020. [CrossRef]
19. Basnet, S.R.; Shakya, S. BSS: Blockchain security over software defined network. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 720–725. [CrossRef]
20. Khanna, A.; Sah, A.; Bolshev, V.; Burgio, A.; Panchenko, V.; Jasiński, M. Blockchain–Cloud Integration: A Survey. *Sensors* **2022**, *22*, 5238. [CrossRef] [PubMed]
21. Bharathi Murthy, C.V.N.U.; Shri, M.L.; Kadry, S.; Lim, S. Blockchain Based Cloud Computing: Architecture and Research Challenges. *IEEE Access* **2020**, *8*, 205190–205205. [CrossRef]
22. Srilakshmi, K.; Bhargavi, P. Cloud Computing Security Using Cryptographic Algorithms. *Asian J. Comput. Sci. Technol.* **2019**, *8*, 76–80. [CrossRef]
23. Tabassum, M.; Perumal, S.; Mohanan, S.; Suresh, P.; Cheriyan, S.; Hassan, W. IoT, IR 4.0, and AI Technology Usability and Future Trend Demands: Multi-Criteria Decision-Making for Technology Evaluation. In *Design Methodologies and Tools for 5G Network Development and Application*; IGI Global: Hershey, PA, USA, 2021; pp. 109–144.
24. Liang, X.; Shetty, S.; Tosh, D.K.; Kamhoua, C.A.; Kwiat, K.A.; Njilla, L. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. In Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, Spain, 14–17 May 2017; pp. 468–477. [CrossRef]
25. Rensaa, J.-A.H.; Gligoroski, D.; Kralevska, K.; Hasselgren, A.; Faxvaag, A. VerifyMed-A blockchain platform for transparent trust in virtualized healthcare: Proof-of-concept. In Proceedings of the 2nd International Electronics Communication Conference, Singapore, 8–10 July 2020; pp. 73–80. [CrossRef]
26. Buccafurri, F.; De Angelis, V.; Lazzaro, S. A Blockchain-Based Framework to Enhance Anonymous Services with Accountability Guarantees. *Future Internet* **2022**, *14*, 243. [CrossRef]
27. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. Integration of Blockchain and Cloud of Things: Architecture, Applications and Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2521–2549. [CrossRef]
28. Shah, M.; Shaikh, M.; Mishra, V.; Tuscano, G. Decentralized Cloud Storage Using Blockchain. In Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), Tirunelveli, India, 15–17 June 2020; pp. 384–389. [CrossRef]
29. Perumal, S.; Tabassum, M.; Narayana, G.; Ponnan, S.; Chakraborty, C.; Mohanan, S.; Basit, Z.; Quasim, M.T. ANN Based Novel Approach to Detect Node Failure in Wireless Sensor Network. *Comput. Mater. Contin.* **2021**, *69*, 1447–1462. [CrossRef]
30. Hankerson, D.; Vanstone, S.; Menezes, A. *Guide to Elliptic Curve Cryptography*; Springer: New York, NY, USA, 2006.
31. Wikipedia. Elliptic-Curve Cryptography. Available online: https://en.wikipedia.org/wiki/Elliptic-curve_cryptography (accessed on 10 December 2021).
32. Remix. Welcome to Remix's Documentation, 2019–2022. Available online: https://remix-ide.readthedocs.io/en/latest/ (accessed on 25 November 2021).
33. Ethereum Revision. Solidity, 2016–2019. Available online: https://docs.soliditylang.org/en/v0.5.3/ (accessed on 4 December 2021).
34. Liang, C.B.; Tabassum, M.; Kashem, S.B.A.; Zama, Z.; Suresh, P.; Saravanakumar, U. Smart home security system based on Zigbee. In *Advances in Smart System Technologies*; Springer: Singapore, 2021; pp. 827–836.
35. Park, A.; Li, H. The Effect of Blockchain Technology on Supply Chain Sustainability Performances. *Sustainability* **2021**, *13*, 1726. [CrossRef]