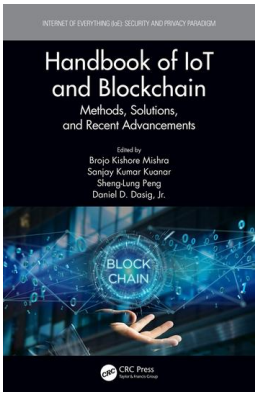

Handbook of IoT and Blockchain
Methods, Solutions, and Recent Advancements

Brojo Kishore Mishra, Sanjay Kumar Kuanar, Sheng-Lung Peng, Daniel D. Dasig

Security and Privacy in IoT

Publication details
https://test.routledgehandbooks.com/doi/10.1201/9780367854744-9
Neelamani Samal, Debasis Gountia
Published online on: 26 Nov 2020

9 Security and Privacy in IoT

Neelamani Samal and Debasis Gountia
CET (BPUT) Bhubaneshwar, India

CONTENTS

The Internet of Things (IoT) plays a remarkable role in our modern daily life. With the use of IoT our daily life has become easier and dynamic. The IoT covers almost all fields, including home automation, office automation, healthcare, sports, industrial applications, and transport, Smart Cities, agricultural, Smart Energy and the Smart Grid, etc. Therefore this technology requires security management and privacy control over the user data which is in the cloud. Security in IoT is the technology area

concerned with securing connected devices and networks in the Internet of Things environment where we have a number of devices connected over the internet with IP addresses. There are so many devices, sensors, things that we are wearing or using, things that we are interacting with that we will not even sense it in future. The IoT enables development of helpful tools and technological partners to resolve security issues which may occur in the future. Due to the small size of modern CPUs, the possible features of IoT and its applications are almost unlimited. IoT devices [16] are connected to the internet, so they are vulnerable to the same kinds of cyber-attack that can affect the user's personal and commercial information.

Through IoT the user is connected to a number of devices for gathering and controlling information. Therefore protecting consumer privacy of information has become difficult as the IoT becomes available to all involved. A number of devices are connected to different kinds of devices and this increase in connectivity and data collection results in less control of privacy of information to the public. Both the control of data and control of the different devices that are connected has become a major area of concern in the modern age of internet access. Therefore maintaining control over the data and device with maintaining privacy is of more concern as control can be lost if someone hacks into a smart phone or computer. Control can be lost when more and more companies collect data about users connected to the internet. Everything that we search, and all of our online activities, is being tracked by many companies that provide us with the services for their data-mining purposes and for improving user experience over the IoT. But sometimes these activities breach user security and privacy of data.

9.1 INTERNET OF THINGS (IOT) SECURITY TECHNOLOGIES

Creating a security framework [6] in IoT is purely problem specific i.e. depending on the case a specific security measure may be taken for improving security in IoT and to fix the shortcomings. Based on usage, it is possible to differentiate six main directions of security solutions for IoT.

In general among the greatest of the Internet of Things security issues, scientists suggest the following points are maximal in maintaining security. These issues include authentication problems, lack of proper data encryptions and security analytics, vulnerability of networks, and problems with the application program interface. The best part of security management is that these issues are successfully handled by the IoT experts to make it more user friendly and reliable, and each of them has a specific approach in terms of IoT security.

9.2 IOT SECURITY FOR NETWORKS

The best approach to creating a safe network with the collaboration of IoT is to connect it to a pre-installed backend system while maintaining security. To achieve the highest level of sophistication in the environment we combine traditional approaches of data protection such as antivirus and firewalls with protocols, standards, and complex device capabilities for specified uses.

9.3 AUTHENTICATION FIXING FOR IOT SECURITY ISSUES

Managing the authentication of accessing devices will definitely add a security benefit and can resolve most of the security threats. The functionality of these IoT security solutions ensures a safe authentication of one user into multiple networks or devices. Apart from the standard password-based procedure of maintaining security, the developers in IoT generally advise use of two-factor [15] authentications, biometrics, and digital certificates and other components that may be required to increase security.

9.4 IOT SECURITY TECHNOLOGIES FOR DATA ENCRYPTION

Data encryption can be considered one of the best ways to establish a secure environment or ecosystem for maintaining security. In this approach information becomes almost unreachable or unreadable for hackers with the help of standard and effective cryptographic algorithms designed and developed by the IoT expert. These tools are commonly used in a large range of applications. Encryption and technologies protect the data from the hacker and the full encryption key lifecycle management process establishes a powerful security system which can increase IoT security. In this context blockchaining can be considered one of the best approaches to establishing security.

9.5 SECURITY ANALYTICS AS A DIMENSION OF IOT SECURITY SOLUTIONS

To facilitate the necessary measures in connection with the Internet of Things security, we need to monitor all the smart devices and carry out regular monitoring and check their performance of the security standard. For this purpose various IoT security vendors offer their analytical capabilities and the tools. For an authentic solution, vendors assist in collecting, monitoring, processing and reporting on data that is given to the IoT devices. And to address the issue properly, analytics toolkits are frequently used which can match recent trending technologies, which also include AI, machine learning, deep learning and big data. In general security analytics can be considered an integral part of the software solution for security analytics.

9.6 IOT SECURITY TECHNOLOGIES: CORE PROTECTION METHODS (IOT API)

Application program interface (API) security capability includes the ability to authenticate and authorize the flow of information inside the fully protected IoT network, which may include smart devices, backend systems or any third-party applications. The API must provide end to end service which helps the client to proceed with the required platform [13] and the deserved API control ensuring the security of the information and the data. This should guarantee safe storage of sensitive information, and should have a system of authentication and authorization.

9.7 HOW TO BUILD TRUST IN IOT

To build trust in IoT environment we have to ensure the following:

9.7.1 Enable Device Authentication

To ensure secure participation in the Internet of Things, each and every connected device needs a unique identification in the network. We have a number of methods to prove an identity to the IoT devices such as passwords, biometrics, digital certificates etc. However, when we focus on providing the secure ID the choice of device depends on the capabilities of the device [9].

In environments where our main point of consideration is only security and safety, a hardware-based authentication provides the best means to establish and maintain authentication of the device identity. The digital certificates issued from a trustworthy [5] vendor of key management can be the proven mechanism for security, whereas the storage and processing of the information received demands the traditional RSA keys and management by elliptic curve cryptography (ECC). The combination of both RSA and ECC makes device authentication more reliable.

9.7.2 Encryption to Protect Data

With the use of a proper authentication device a proper encryption mechanism is essential to protect the IoT data which is confidential and has high significance. In IoT the device must protect user data from the device itself and the environment and cloud storage.

This requires process steps to identify the data to be encrypted, and also a key management scheme to distribute and manage the keys that are used to encrypt the data. The secure storage and access control for keys need proper investigation and planning before giving permission or the key to the user. Keys need to be properly managed and integrated in order to ensure security. The IoT and its applications are increasing day by day so for more authentic use the key management must be scalable and dynamic all the time.

9.8 BLOCKCHAIN TECHNOLOGY

A blockchain technology based system is a classical distributed system [11] where all the participating entities are geographically scattered but connected through different types of networks [2]. Decentralization is the fundamental characteristic of blockchain which can be employed to solve the issues of traditional transaction management systems. Blockchain basically provides a platform where multiple entities which do not trust each other can work or share information in a common platform.

This technology is a decentralized [12] architecture having all transactions recorded as a digital ledger. All the nodes are connected in a distributed manner such as mesh topology. Transactions occurring between any nodes are passed through verification by the blockchain network, then after the mining process the completed transaction is recorded in a block. The block consists of numbers of valid transactions between

TABLE 9.1
Comparison of permissionless and permissioned blockchain

	Permissionless Blockchain	Permissioned Blockchain
Environment Type	Open	Closed
Participation in Consensus	All nodes	Selected nodes
Identity	Pseudo-Anonymous	Registered Participants
Consensus Type	Lottery based	Voting based
Transaction Processing Speed	Slow	Fast
Consensus Algorithms	Proof of Work, Proof of Stake, Proof of Burn, Proof of Delegated Stake, etc.	Paxos, Practical Byzantine Fault tolerance, Raft, etc.

different nodes. Once recorded in a block the transaction can never be changed. So blockchain provides an immutable digital ledger among all the nodes present in the network. It builds trust [7] among all users as all the users have the same set of digital record present among themselves and whatever happens in the network one can see it. Blockchain technology has security, and privacy issues of the users are addressed using public and private key concepts also using a digital signature. Blockchain can mainly be used in two different ways; permissionless and permissioned. Table 9.1 provides a brief comparison of both the technologies. Permissionless design which is generally established on an open environment allows anyone to join the system as well as allowing writing to shared blocks. Permissionless design also gives equal privilege to all the nodes in case of consensus process. A permissioned blockchain design is managed by a known set of entities and is established in a closed environment. Though all the entities are allowed to perform transactions, only a fixed set of predetermined nodes can take part in the consensus process in a permissioned blockchain. Consensus algorithms [4] play an important part in managing an efficient and secure blockchain system.

9.9 DISTRIBUTED CONSENSUS ALGORITHMS

The concept of consensus is an engrossing topic in a decentralized or distributed network. Consensus means a procedure to arrive at a common agreement in a decentralized or distributed multi-agent platform. In a conventional distributed system, we apply consensus to ensure reliability which ensures correct execution in the presence of faulty individuals and fault tolerance. In addition to this, a distributed consensus mechanism should satisfy certain properties such as termination, validation, integrity and agreement. An example of consensus is state machine replication, which is a key aspect of any distributed consensus protocol. For example, if we want to run some kind of distributed protocol over a network, every individual entity runs the current protocol and they store the state of the protocol in different state machines. So the entire execution part of the protocol can be represented as a state machine. Now this state machine needs to be replicated to multiple entities so

that every individual entity can reach a common output of the protocol. Achieving consensus can be easy and straightforward for certain architectures under certain scenarios. The scenarios could be either that the entire system is faultless or that there is not be any failure in the system, so that every entity can receive the message [19] correctly or the system behaves in a synchronous way, i.e., it is expected that you will receive all messages within some predefined time interval.

However, achieving consensus can be non-trivial in the case of a distributed environment [13] due to the presence of multiple types of failure. Typically in a distributed system, we consider three different types of failure:

1) Node Failure: A node suddenly crashes or becomes unavailable in the middle of communication. Therefore we are not expected to receive any message from that particular node. This can be a hardware or software fault.
2) Partitioned Faults: This type of fault occurs whenever links fail, which results in partition in the network. This can hamper reaching of consensus.
3) Byzantine Faults: This kind of fault is more difficult to handle in a distributed environment. Here the entity starts behaving maliciously. In both the above faults, we can expect an effect on the network, but in this case, it is difficult to guess because it completely depends on how maliciously the entity is behaving and what the entity is doing.

The correctness of a distributed consensus protocol can be characterized by the following two properties; safety and liveliness. Safety ensures that one will never converge to an incorrect state. And liveliness ensures every correct value must be accepted eventually.

9.10 CONSENSUS IN PERMISSIONLESS BLOCKCHAIN SYSTEM

Conventional consensus mechanisms will not be applicable in an open or permissionless blockchain system. Therefore the following section shows how consensus has been achieved in bitcoin-like open environments along with their shortcomings.

9.10.1 Bitcoin Consensus

The main purpose of consensus in bitcoin is to add a new block to the existing blockchain. There can be multiple miners in the bitcoin network and these miners can propose their new blocks based on the transactions they have heard of. It is not necessary that every miner propose the same block. Generally the miners include those transactions in the new block that they have heard of since the last time a block has been added. The miners also need to ensure that size of the newly proposed block does not exceed a certain threshold. We should focus on the following two observations before designing the algorithm:

Any valid block (a block with all valid transactions) can be accepted even if it is proposed by only one miner.

The entire protocol can work in rounds. Broadcast the accepted block to the peers and collect the next set of transactions.

Based on the above two observations, we can design solutions so that every miner independently tries to solve a problem and the block is accepted for the miner who can prove first that the challenge has been solved.

9.10.2 Proof of Work (PoW)

The idea of PoW came in the year 1992 from Dwork and Naor to combat junk emails where we have to do some work to send a valid email. The attacker this way will be discouraged from sending junk emails because in that case they have to do some work of some complexity in order to forward the junk emails that does not prove beneficial for them. A blockchain-based PoW system must have certain features such as asymmetry: The task must be relatively hard but feasible for the service requester; the task must be easy to verify for the service provider. In this way the service requester will get discouraged from carrying out the work but the service provider can easily check the validity of the work because of the asymmetric nature of the work. Bitcoin-based PoW systems extend the hashcash-based PoW system and develop a methodology to protect the blockchain by applying the distributed consensus mechanism. The hashcash system was proposed by Adam Back and uses the puzzle-friendliness property of the cryptographic hash function. Now coming to bitcoin-based PoW, miners who take part in the consensus process need to give a proof that they have done some work before proposing a new block. The attacker will be discouraged from proposing a new block or making change in existing blocks. Because in that case they have to do the entire work of the blockchain which is computationally difficult in a generic environment. Every miner will try to find out a nonce value which will satisfy a certain hash equation. Most implementations of bitcoin PoW use a double SHA256 hash function. In the default set up, all miners wait for around ten minutes and look for all transactions which have taken place within that duration. Then they start the mining process. As we have seen, the probability of getting a PoW is very low; hence it will never happen that any miner will be able to control the bitcoin network exclusively. In case any attacker wants to make some changes in one block, they have to do more work compared to the collective work of all the blocks in the longest chain, which is computationally difficult, though not impossible, thus making an attack difficult with current hardware and hence the bitcoin system is tamper-proof. These tamper-proof characteristics of PoW also ensure that double spending does not happen in the case of a blockchain network. Apart from this, PoW-based systems suffer from a number of security attacks such as sybil attack, DOS attack, etc. In the case of a sybil attack, the attacker tries to fill the network with clients under its control. This helps the attacker to control the entire network. These clients work according to the instructions of the attacker which compromises the PoW mechanism. In DOS, the malicious node will send a lot of data to a particular node or nodes which will hamper the normal functioning of the bitcoin transactions. Another major flaw in a bitcoin system is the monopoly problem. This happens when a particular miner gains control of the network by deploying huge servers for the mining process. So it may happen that this particular miner will gradually generate a huge number of blocks and hence the miner can control the entire flow of transactions. As a result,

other nodes will get discouraged from joining as miners and only a few miners with large computing resources [10] will control the network.

9.10.3 Proof of Stake (PoS)

The PoS mechanism is an improved version of PoW to reduce the excessive electricity consumption of PoW-based systems. As a substitute to the computationally expensive PoW mechanism, PoS aims to stake nodes' economic share in the network. Like PoW, a node (miner) is selected to add a block to the blockchain. But here the miner selection procedure is different from PoW. In PoS, the selection of a miner is proportional to the amount of bitcoin it holds. Block finality in the PoS-based system is faster compared to PoW blockchain, as computationally difficult puzzle solving is not within PoS. The PoS-based algorithm developed by Ethereum is known as Casper. The PoS algorithm pseudo-randomly chooses miners to create blocks of the blockchain, so that no miner can predict its turn in advance. It also solves the monopoly problem of PoW-based consensus. Naive PoS algorithms are prone to an attack known as Nothing-at-Stake, and thus require some improvements in order to provide safety.

9.11 CONSENSUS IN PERMISSIONED BLOCKCHAIN SYSTEM

Permissioned blockchain consensus is achieved through the help of a smart contract, which is basically an extension of bitcoin scripts. Smart contracts are self-executing software programs in which the terms and conditions among the negotiating parties are written down. Generally, the concept of state machine replication mechanism is used to ensure consensus in the permissioned blockchain environment because of the following reasons such as the network being closed and the nodes knowing each other, so state replication is possible among the known nodes. And it avoids the overhead of mining.

9.11.1 Paxos

The first ever consensus algorithm, Paxos, was proposed by Lamport with the objective of choosing a single value under crash or network fault. The idea behind Paxos is very simple: All the nodes in the network have been categorized into three types; namely the proposers, the acceptors and the learners. Everyone is a learner in the network that learns the consensus value. The proposer initially prepares a proposal with a proposal number known as a prepare message and sends it to the acceptors. This proposal number forms a timeline and the biggest number is considered up to date. Each acceptor compares received proposal number with current known values for each proposer's proposal message. If it gets a higher number, then it accepts the proposal or otherwise declines it. Then the acceptor prepares a response message of the following form: Prepare response where the proposal number is the biggest number the acceptor has seen and accepted values are the already accepted values from another proposer. Next a vote is taken based on the majority decision. The proposer checks whether the majority of the acceptors have rejected the proposal. If yes, then the proposer updates

it with the latest proposal number. If no, then the proposer further checks whether the majority of the acceptors have already accepted values. If yes then the proposer's value cannot be selected or otherwise it sends an accept message. Finally, the proposer sends an accept message with the following format to all the acceptors. Whenever the acceptor accepts a value, it informs the learner nodes about it so that everyone will learn about the accepted value. If more than (Nonce/2 – 1) acceptors fail, then no proposer will get enough reply messages to form a conclusion and we cannot reach consensus. Even though the concept behind Paxos is very simple to understand, the theoretical proof is very complicated. Real-life application of this may need to go for a sequence of selections which is known as a multi-Paxos system. Multi-Paxos systems need a repeated application of Paxos which increases the system complexity. This is because the number of messages exchanged between the nodes also increases. There is also a chance of livelock or starvation in Paxos-based systems.

9.11.2 Raft

The Raft algorithm is designed as an easy alternative to Paxos. The basic idea behind Raft is that nodes collectively select a leader and rest of the nodes become followers. The leader is responsible for state transition log replication across the followers. Record entries flow in only one direction from the leader to the followers. Unlike Paxos, here each node can be at any of the three states namely leader, candidate and follower at any particular time. The Raft algorithm runs in rounds which are known as term. Each term starts with an election where one or more candidates strive to become leader. Initially, we have a set of follower nodes, who look out for a leader. If within a certain time interval they do not find one, then the leader election process starts. In this election phase, some of the followers volunteer to become a leader and request votes from all other nodes. Then these candidate nodes send request messages to other followers of the system for a vote. When a node receives the request, it compares the term and index in the received message with corresponding current known values. Like Paxos, the followers vote for one of the candidates and based on the majority of votes, a leader is selected. Then in the next part, the elected leader will propose values which the followers will choose so that the system can reach consensus. The leader sends out heartbeats (signals) to all followers at regular intervals in order to maintain its authority. Compared with Paxos and PBFT, this algorithm has high efficiency and clarity. Hence it has been extensively employed in distributed systems. The Raft algorithm achieves the same safety performance as Paxos and is better suited in real-life implementation and comprehension. As mentioned, the Raft algorithm cannot support byzantine nodes and can stand up to failure of 50% of nodes. As in the case of permissioned blockchain, nodes are verified members. Hence, it is more essential to resolve crash faults rather than Byzantine faults for private blockchain.

9.11.3 Byzantine Fault Tolerance and Its Variant

In relation to distributed systems, Byzantine Fault Tolerance is the capability of a distributed network to execute as required and correctly reach a consensus despite malicious nodes. It is derived from the Byzantine General Problem, where the

general sends an attack message to one group of lieutenants whereas he sends a retreat message to another group of lieutenants. Hence it becomes difficult for the system to find out what action to take. Byzantine fault-tolerant systems are typically built using replication. For this, the state machine approach is used which helps to implement fault-tolerant services. The variant of BFT that has been designed for synchronous distributed systems is called the "Lamport–Shostak–Pease" algorithm. This ensures consensus in presence of a number of faulty nodes, provided we have (2f + 1) number of lieutenants apart from the commander. But our real systems behave in an asynchronous way as there is no guarantee that a message will be received within a certain time interval. For this reason, a variant of BFT known as Practical Byzantine Fault Tolerance (PBFT), has been developed for real-life asynchronous systems. As in the case of a pure asynchronous system, achieving consensus is impossible even in the presence of a single faulty node. So to ensure liveness property, instead of pure asynchronous system, a weak asynchronous system has been considered. Coming to the algorithm, the byzantine model consists of three types of nodes: the clients, a commander and the lieutenants. The entire algorithm runs in three phases: pre-prepare, prepare and commit phase. In the pre-prepare phase the commander assigns a sequence number to the request submitted by a client and multicasts it to the network. Among other data, the request message also contains the digital signature and message digest for verification. The lieutenants of the network confirm the block by verifying the digital signature and message digest. Once the validating lieutenants accept the pre-prepare message, they enter the prepare phase by multicasting the message to the rest of the network. Once again both the commander and lieutenants verify the prepare messages before accepting them. The messages commit, when (2f) prepare message from different backups match with the corresponding pre-prepare messages. Hence, the total (2f + 1) votes one from primary from the non-faulty replica help the system to reach to a consensus. With the evolution of ICT, the blockchain technology has attracted interest from various directions. The consensus algorithm is the main technology of blockchain. In the case of permissionless systems, it is easy to achieve robust consensus among large number of untrusted nodes using complex computations though transaction, finally remains non-deterministic. On the contrary, permissioned blockchain provides high throughput in less time while sacrificing a degree of decentralization.

Since the concept of IoT arose, much development has taken place in a different application. The architecture of IoT is basically three layer perception, network, and application layer to make real use of the IoT technology for efficiency and reliability of the system. Security and privacy issues are the challenges [20] to implement a different application using the IoT concept. Some of the challenges are access control, authentication, centralized or distributed network, and the identity of the things. Visualization is one the important aspects [21] of the IoT application. The end-user monitors the environment using either mobile phone or smart tablets, although IoT applications use both centralized and decentralized architecture. The decentralized architecture has an advantage over centralized architecture. The IoT application system has different stages. Processing and doing computation may depend on cloud computing or the recent development of fog computing. The storage purpose IoT

application can store at the edge device or at the cloud server depending upon the storage space availability. In IoT / physical layer all the required sensors are deployed in the application area. The sensors are connected to the next layer using WiFi or through a wire. IoT application demands faster processing and quick response for better utilization of the system. Most of the application collecting data from sensors need to process in the edge of the network for faster processing and temporary storage for quick analysis of the data. The decision can be taken by analyzing the data either in a collaborative way or a distributed way by mutual agreement. All these activities can be done in the Fog Layer / Processing layer. The Cloud Layer provides different types of services such as infrastructure, platform, and software as a service. In an IoT application as different sensors collect information from environment continuous way, it is not possible to provide large memory space. Also, IoT devices are low in memory capacity. Similarly, the storage capacity of a fog device is also limited, so the final storage must be provided by cloud computing. For establishing security and privacy in communication and user data, authentication of IoT devices is highly essential. When devices perform any operation in an application, the devices need to be identified using their unique ID. Using the unique ID, devices can connect to the next layer also performing different computation in a collective way. The authentication of the devices means the communication can be done securely; all the nodes are identified using their ID. The decentralized scheme for IoT device authentication can be considered one of the best approaches for secure communication. The process of device authentication is done in blockchain-based technology. Implementation takes place using the Ethereum platform and Web3 client is used to integrate block chain smart contract with Frontend.

9.11.4 Security Analysis

IoT implementation requires many security and privacy issues to be addressed. Here a decentralized approach is proposed for authentication to increase IoT network connectivity and build trust among all the devices. In an IoT system not only information needs to communicate securely it also needs to identify which devices are authenticated devices. Each of these devices identify using their public key which is a unique key generated from the system. Security and integrity of the IoT system are ensured without using a centralized system. The proposed authentication protocol [17] is secured against some of the attacks described below:

1) **MITM (Man-In-The-Middle)**: In the proposed scheme, MITM attack is not possible as whatever message is sent from one node to another node, it is sent using hash techniques and public-private key concept so that only the authorized user will get the message. Any modification done on the message will be ignored by the receiver side. The message communication also uses a digital signature as one of the concepts of blockchain technology.
2) **Impersonation Attack:** Whatever transaction is done in the blockchain network, all are verified and mined by the network using mining process and digital signature concept. Each user has their own identity added to the system using the authentication process.

3) **Replay Attack:** In a blockchain environment, no node can capture more than half of the network power. Each transaction sent or received is recorded in a digital ledger after verification and mining. So no transaction can transmit multiple times in the network.
4) **Denial-of-Service (DoS) Attack:** As each message is broadcast in the network and it goes through a verification process to check valid or invalid transaction, there is no chance that a DoS attack is possible.

Due to all the above security issues, a decentralized web authentication system using blockchain is proposed which is not a password-based authentication and authentication is done using AuthKey which is a 160-bit hash and is secured enough to prevent all the above attacks.

9.11.5 Decentralized Authentication

Decentralized authentication is an attempt to make a decentralized site login and authentication protocol. It is analogous to the "Log In with Facebook" button that we have probably become accustomed to. It is a smart contract that will store user IDs and their associated wallet addresses. The user ID is simply a UTF-8 string with size ranging between 2 and 32 bytes. The user himself creates it on inception of the wallet and will later use it to enter any site that supports decentralized authentication. It would also be possible to add a restriction on the possible characters included in the string. One could restrict it to Latin characters and Arabic numerals in order to limit the possibility of creating visually similar IDs. When creating an account with decentralized authentication, a pair of keys is created. We will create an authorization key and a key to restore access. When created, both addresses are the same as the address of the wallet which first made the transaction. Users who care about their security should create a separate Master Key and store it in a place that is inaccessible from online. Recovery seeds are a set of 12 mnemonic words that when used can recreate the key pairs for our wallet. If we are going to be using a wallet for authentication then it is also recommended we use a separate address from the one that keeps all of our ether. Doing this allows us to avoid any hackers from tracing our Authkey to the wallet with our assets. Decentralized Web Authentication System using Ethereum-based blockchain protects our assets.This is something that could be updated in later iterations of the smart contract. If we want to further protect ourselves, then we consider VPN services.

9.11.6 Using Decentralized Authentication

There is a dedicated web page intended for user interaction with the smart contract. We can create an account there, change the keys or delete it. To work with it, the user will need to install the well-known browser plugin called MetaMask. Of course, if we are already an experienced user of the Ethereum network then we will already have used MetaMask and will probably have an idea of how it interacts with the network. The overall user authentication process using DecAuth looks as follows: The site (backend) contacts the smart contract and receives the user's Ethereum address.

Then the site (backend) generates and records a message, and asks the user to sign this message with the help of the authKey address. And the user being on the site (frontend) signs the message using the MetaMask plugin and sends it to the backend. Finally the site (backend) verifies the signature, and if everything is correct, it activates the user's session. It is important that authentication checks should take place in a user-uncontrolled environment. So, in other words, all of the checks should be completed on a server instead of on a user's browser.

The decentralized system is not subject to censorship by the large entities, such as Google or Facebook. If it is necessary to censor something, each website should implement it independently. Yet this would only affect the user's interaction with that site and not any others. The Ethereum network currently has quite slow transaction speeds when creating an account; the user may have to wait a few minutes, but sites can get the data and verify users quite quickly. This solution scales well, because there are a lot of data nodes, and anyone can add another one at any time. The complexity of implementing such a solution for site owners is no higher than the complexity of implementing OAuth 2.0. Using blockchain-based authentication, all devices are connected in a peer-to-peer way. Each of these devices identifies using their public key which is a unique key generated from the system. Security and integrity of the IoT system is ensured without using a centralized system. Due to different kinds of security issues, a decentralized web authentication system using blockchain is proposed which is not a password-based authentication and authentication is done using AuthKey which is a 160-bit hash and is secure enough to prevent all the above attacks and the blockchaining method makes it secure against all kinds of attack.

REFERENCES

[1] Mahmoud Ammar, Giovanni Russello, Bruno Crispo (2018) Internet of Things: A Survey on the Security of IoT Frameworks, Journal of Information Security and Applications (JISA), vol. 38, pp. 8–27.

[2] Sachi Nandan Mohanty, K.C. Ramya, S. Sheeba Rani, Deepak Gupta, K. Shankar, S.K. Lakshmanaprabu, Ashish Khanna (2020) An Efficient Lightweight Integrated Blockchain (ELIB) Model for IoT Security and Privacy, Future Generation Computer Systems, vol. 102, pp. 1027–1037.

[3] Bhabendu K. Mohanta, Anisha Sahoo, Shibasis Patel, Soumyashree S. Panda, Debasish Jena, Debasis Gountia (2019) "DecAuth: Decentralized Authentication Scheme for IoT Device Using Ethereum Blockchain", in Proc. of the 2019 IEEE Region 10 Conference (TENCON), pp. 1–6

[4] Soumyashree S Panda, Bhabendu Ku. Mohanta, Utkalika Satapathy, Debashis Jena, Debasis Gountia, Tapas Kumar Patra (2019) "Security (In-Depth) Analysis of Blockchain Based Decentralized Consensus Algorithms", in Proc. of the 2019 IEEE Region 10 Conference (TENCON), pp. 1–6

[5] Bhabendu K. Mohanta, Utkalika Satapathy, Soumyashree S. Panda, Debasish Jena, Debasis Gountia (2019) "Trustworthy Management in Decentralized IoT Application using Blockchain", in Proc. of the IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–5

[6] Soumyashree S. Panda, Utkalika Satapathy, Bhabendu K. Mohanta, Debasish Jena, Debasis Gountia (2019) "A Blockchain Based Decentralized Authentication Framework for Resource Constrained IOT Devices", in Proc. of the IEEE International

Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7.

[7] Sicari, Sabrina, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini (2015) "Security, Privacy and Trust in Internet of Things: The Road Ahead." Computer Networks, vol. 76, pp. 146–164.

[8] Zhang, PeiYun, MengChu Zhou, and Giancarlo Fortino (2018) "Security and Trust issues in Fog Computing: A survey." Future Generation Computer Systems, vol. 88, pp. 16–27.

[9] Kang, Kai, Zhibo Pang, Li Da Xu, Liya Ma, and Cong Wang. "An interactive trust model for application market of the internet of things." IEEE Transactions on Industrial Informatics 10, no. 2 (2014): 1516–1526.

[10] Jeong, Seohyeon, Woongsoo Na, Joongheon Kim, and Sungrae Cho. "Internet of things for smart manufacturing system: Trust issues in resource allocation." IEEE Internet of Things Journal 5, no. 6 (2018): 4418–4427.

[11] Roman, Rodrigo, Jianying Zhou, and Javier Lopez. "On the features and challenges of security and privacy in distributed internet of things." Computer Networks 57, no. 10 (2013): 2266–2279.

[12] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 858–880, 2018.

[13] T. Neudecker and H. Hartenstein, "Network layer aspects of permissionless blockchains," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 838–857, 2018.

[14] J. Chauhan, S. Seneviratne, Y. Hu, A. Misra, A. Seneviratne, and Y. Lee, "Breathing-based authentication on resource-constrained iot devices using recurrent neural networks," Computer, vol. 51, no. 5, pp. 60–67, 2018.

[15] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," IEEE Internet of Things Journal, vol. 6, no. 1, pp. 580–589, 2019.

[16] F. Sun, C. Mao, X. Fan, and Y. Li, "Accelerometer-based speed-adaptive gait authentication method for wearable iot devices," IEEE Internet of Things Journal, vol. 6, no. 1, pp. 820–830, 2019.

[17] Z. Wang, "A privacy-preserving and accountable authentication protocol for IoT end-devices with weaker identity," Future Generation Computer Systems, vol. 82, pp. 342–348, 2018.

[18] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[19] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," Computer Networks, vol. 57, no. 10, pp. 2266–2279, 2013.

[20] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," Future Generation Computer Systems, vol. 82, pp. 395–411, 2018.

[21] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in iot: The challenges, and a way forward," Journal of Network and Computer Applications, 2018.