Journal of Network and Computer Applications 144 (2019) 59–78

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

QoS-aware secure transaction framework for internet of things using blockchain mechanism

Deepsubhra Guha Roy [a], Puja Das [a], Debashis De [a,b,*], Rajkumar Buyya [c,d]

[a] Centre of Mobile Cloud Computing, Department of Computer Science and Engineering, West Bengal University of Technology, Presently, Maulana Abul Kalam Azad University of Technology, WB BF-142, Sector-I, Salt Lake City, Kolkata, 700064, West Bengal, India
[b] Department of Physics, University of Western Australia, 35 Stirling Hwy, Crawley, WA, 6009, Australia
[c] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia
[d] Manjrasoft Pty Ltd, Melbourne, Australia

ARTICLE INFO

Keywords:
Internet of things
Man in the middle attack
Risk assessment
Risk recovery blockchain
Congestions control
QoS

ABSTRACT

The Internet of Things (IoT) paradigm enables an enormous network with millions of connected smart things to control almost all aspect of services. In an IoT scenario, self-configured things are dynamically connected in a universal network. The small things are broadly distributed in a real world model with some essential processing capacity and finite storage. Security, privacy and reliability are the most important pillars of the IoT infrastructure. It is challenging to share personal data, highly confidential professional data over a centralized system. The infrastructure itself is capable to provide the guarantee to transmit data safely. However, a special attention for the data along with data consistency is required. Cyber attackers and hackers mostly target the centralized systems. To triumph over these types of situations we propose a security based blockchain service along with some encryption approaches for a secure communication. The encryption and decryption is performed by a hash based secret key. Risk modelling and relative risk reduction also done to prevent attacks under a virtual private network in IoT domain. Before and after attack users also wish to know quality of service (QoS). We have developed a mobile application framework which provides desire output of a user from different aspect of QoS for better understanding. The MQTT broker performs as a negotiator to convey data between cloud consumer and cloud provider. In QoS monitoring section user get chance to reconsider availability, reliability and responsibility like QoS factors while recovering resources and actual resources before attacking system based on CPU utilization, cache status and storage. As per experiment result we able to recover 100% RAM, 97.64% CPU utilization and 78.7% storage using proposed encryption and algorithm on blockchain infrastructure.

1. Introduction

Internet of Things (IoT) is a worldwide interrelated network of objects and human beings through which individual addressing structures (Andersen et al., 2017) are capable to connect from one to all. Using digital and IoT based facilities create the huge amount of data which is increasing rapidly. The principal purpose of IoT is to broadcast information of objects, which reflects the fabrication, consumption, transportation along with each and every detail of public life. Use of IoT interface and implementation make the surrounding environment much better perceptive (Pavithra and Balakrishnan, 2015). Till this date, centralized systems are handling and responsible to take of a huge amount of personal and sensitive information for both public and private domain. However as an individual person has no control over the data stored about them self and not even they have any clue of it's used. In these years, media covered and publish controversial cases related to privacy and security issues and loss of confidential data over public and private data repository (Yang et al., 2018; Zyskind and Nathan, 2015). In today's world IoT is a tool which is possibly applied on every non-living object; that is why it is also known as “Internet of Everything”. IoT is used as a tool or service for betterment of our daily life; while cybercriminals also get opportunity to engage in cybercrime (Roman et al., 2013). For example, 51% attackers or hackers try to take control of these centralized system mechanism, using the same technology base. Various type of attacks like man in the middle, data tempering, backdoor attack are applied and get intact control of the entire system. Because of this security issues

* Corresponding author. West Bengal University of Technology, Computer Science and Engineering, B.F.-142, Sector-I, Salt Lake, Kolkata, 700064, West Bengal, India.
E-mail address: dr.debashis.de@gmail.com (D. De).

https://doi.org/10.1016/j.jnca.2019.06.014
Received 4 September 2018; Received in revised form 31 May 2019; Accepted 27 June 2019
Available online 5 July 2019

IoT unable to get its actual popularity. But to overcome all this third party immersion issues blockchain has introduced (Roman et al., 2013) (Arunkumar et al., 2017).

Blockchain is approximately gets revolutionary outcome for prioritized insecurity of decentralized, distributed, authentic, cryptography, immutability like provenances. It uses in finance, governance, intelligence and health along with for content transfer networks, smart grid and smart agriculture systems (Aslam et al., 2017) (Veres and Boda, 2000). Blockchain has the capability to improve every insecure digital system. Blockchain successfully avoid man in the middle attack, backdoor attack with data reliability and digital personalities to permit IoT data transparency. Indeed, blockchain also play a superior role for privacy, availability and reliability, to improve flexibility and inspecting. For it's features it can solve traditional database synchronization problem. The blockchain technology comes with six key elements (Ferrag et al., 2018).

- **Decentralized:** It is the main feature of blockchain, which define, record, store and update each data in distributed way with peer to peer advantage; therefore no centralized node anymore.
- **Transparent:** Recording data in blockchain system is transparent for each node, it also maintain the transparency on update the data; that is why blockchain is trusted.
- **Autonomy:** Using consensus, each and every single node present on the blockchain be able to handover or modify data securely. The notion is establish a trust-based system form one to all. Therefore the system is secure.
- **Immutable:** In blockchain every published data will be kept for endlessly, until and unless someone can take in charge or control more than 51% node in the unchanged time.
- **Open Source:** Maximum case blockchain scheme is open to everyone, record can be validated publicly even if someone want to create any application as per requirement; they can do it effortlessly.
- **Anonymity:** Blockchain technology resolves the trust problem between nodes, so data handover or even contract can be nameless, only person's blockchain address is needed.

Mostly in the block, it holds main data, hash of present block, and hash of preceding block, timestamp and data information. Fig. 1 shows the organisation of blocks into a continuously growing blockchain.

- **Main data:** Define what service request is made by the blockchain applicant, like business deal records, bank payment records, agreement records or IoT statistics record.

In our framework based case study some IoT related real time data have been in consideration.

- **Hash:** When a transaction is implemented, it had been hashed to a programme before broadcast to every other node. Because, a block can contain thousands of transaction records, to generate a final hash value blockchain use merkle tree function that is root of merkle tree. This last hash contain block header with hash of current block. Using merkle tree function, data diffusion and computing resources can be extremely minimized.
- **Timestamp:** When block or node is generated, blockchain counts those time spans as timestamp.
- **Supplementary Information:** It may be signature of the block, value, or some data defined by the user.

2. Related work

Risk identification, analysis and modelling of risks are mainly focused in IoT services (Gai et al., 2018). The risk modelling is the primary objective of the work, and for that, some notable works on the risk identification and relative risk reduction are mentioned. In (Gramoli, 2017; Lazarenko and Avdoshin, 2018) some IoT challenges like denial of service (DoS), eavesdropping, and physical damages related issues have been raised by the researchers. It also shows some interesting properties, strength and weakness of the IoT platform but no solution is provided for those challenges so far. However blockchain can be the adaptive solution of IoT relies machine-to-machine communication; it can provide connectivity for anyone at any time in secure manner. The generic architecture of IoT, distinguishing features, challenges associated with the development of IoT and applications of the IoT are briefly described in (Mingxiao et al., 2017; Peterson et al., 2016). In future, IoT will create massive networks of more than a trillions of "Things", communicating with each other. Communication faces developmental, hardware, technical, privacy and security interrelated challenges (Lee and Lee, 2017). All of these difficulties are described in brief in section 4. The scope of the IoT in the modern world followed by brief description of related issues is present in (Kshetri, 2017). IoT characteristics, some of the technical and application oriented challenges which can be addressed by blockchain solution are also identified in this paper. IoT applications focus on sensing the environment, interaction and providing the information to the users (Zhang and Wen, 2017; Choi et al., 2016; Lee et al., 2016). An overview of key challenges and opportunities presented by this new technology are illustrated in (Ali et al., 2015) (Roy et al., 2016). Table 1 highlights various challenges regarding IoT data transfer and computation with possible solution using blockchain.

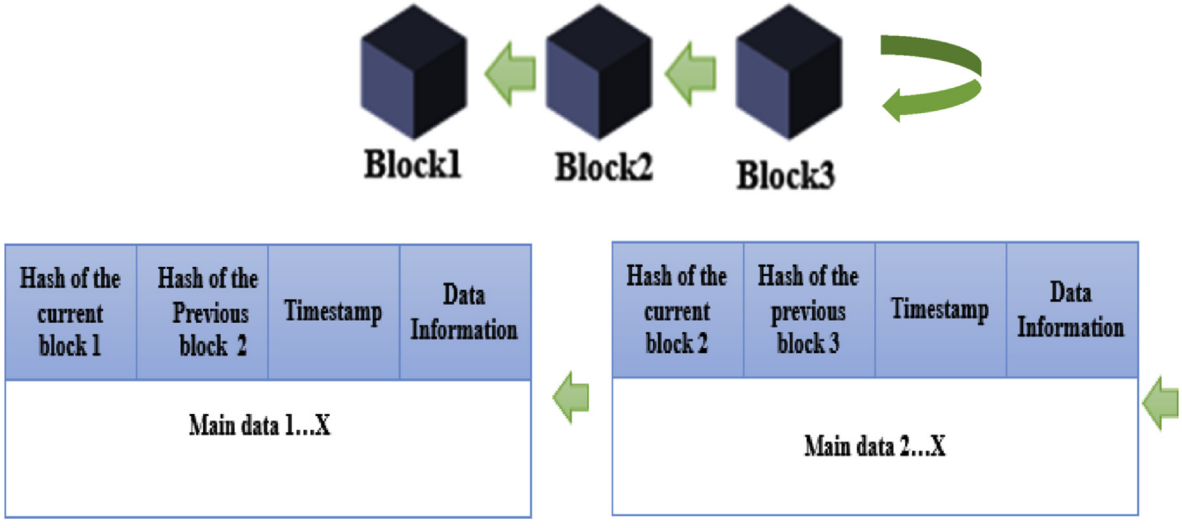


Fig. 1. Traditional blockchain structure.

Table 1
Challenges related to IoT service model with blockchain oriented solution.

Dispute	Justification	Prospective Blockchain Solution
Robust Connectivity	Challenging issue to connect exponentially increasing number of IoT nodes. Each node is vulnerable in nature due to various attacks like denial-of-service, remote hijacking, data stole, hacking.	Smart device-to-device connectivity. Identify each device with verification to guarantee that the originator only can send and definite destined device only can listen to the message.
Efficient Architecture	Faces bottle neck problem as each IoT block consists of some point of failure, makes the intact network disrupt.	No more centralized management; Smart contract update formulae to ensure automatic secure execution.
Time Constraint and Availability	Cloud server down due to recovery from cyber-attack, unnecessary increased of response time while managing data security.	Remove single point failure. Distributed data holding and computation.
Security and Data Protection	Information hacking and manipulation of data can responsible for entire IoT operation.	Distributed data store to handle chain reaction of harmful malicious attack. Interlock provision: System rejection if single appliance's blockchain update is breached.

Security and data privacy remain the major issues for the IoT service (Lin and Liao, 2017). Extensive researches have been achieved towards the development of IoT service. Some IoT communication protocols and cross-layer mechanisms has been proposed in (Clifton et al., 2004). IoT security challenges and their solution have been briefly discussed and eight different categories are identified (Abie and Balasingham, 2012). They raised some open issues for the future research in this domain. It is showing that conventional cryptography is not completely apposite for the IoT systems, because of limited storage and computational space. An encrypted query processing system for IoT also has proposed with inbuilt efficient database query processor in (Roy et al., 2018). In a new approach in query processing to reduce latency for IoT services has been proposed. A light weight encryption algorithm for home automation has been discussed in (Walker et al., 2005), based on identity-based encryption (IBE) technique. It is the combination of Diffie-Hellman Algorithm and IBE. This technique is known as Phong, Mtsuka and Ogata's stateful IBE scheme. Four distinct type of IoT devices attacks: software, physical, network and encryption has been identified in (Cai et al., 2014). A unique IoT heterogeneous identity-based authentication technique for IoT devices has been illustrated in (Kim et al., 2017). A dynamic risk assessment technique for the IoT has been proposed in (Makhdoom et al., 2019), which is inspired by the artificial immune system. A conceptual risk management framework in the domain smart working environment has been illustrated in (Huh et al., 2017). An illustrated risk assessment framework for cloud service eco-system has been proposed in (Zhang et al., 2018). Four risk categories, namely general, technical, policy and legal risks and risk assessment model are the main contribution of the article (Makhdoom et al., 2019). Some empirical risk analyses for home automation system are illustrated in (Lin et al., 2018). Some product-based and scenario-based approaches are briefly described in (Kim et al., 2017; Huh et al., 2017). The use of blockchain in fog and edge computing is proposed in (Tuli et al., 2019). The risk modelling, risk analysis of IoT services and the associated risks to a particular layer of IoT are figured out in this article. In Table 2, a comparative study is shown with the existing approaches. The article therefore proposes a risk analysis based secure IoT solution with the help of blockchain.

3. Contribution of proposed work

We are living in a digital world and this digital interfaces support to make smart life style (Gai et al., 2018; Gramoli, 2017; Lazarenko and Avdoshin, 2018). IoT plays a significant role to make this kind of digital world. In IoT platform every objects are connected with each other's and form a huge communicated network (Mingxiao et al., 2017). In this integrated network data can be availed publicly after published to reach to the destination node. These published data become attractive for hacker or intruder although this data are quite personal and sensitive. To overcome this mishap we develop a secure solution using blockchain. The contributions are of the proposed methodology are as follows:

i. Proposed algorithm offers end to end safety of transaction using blockchain.
ii. Slow transmission rate issue of blockchain is reduced in this proposed framework.
iii. For secure transaction, trust based calculation is done after choosing a network path.
iv. We provide different case studies along with corresponding problem and it's solutions.
v. Various attacks are fabricated through emulation using Kali Linux tools for experimental purpose.
vi. We also provide android based QoS aware monitoring interface to control the network.

4. Risk observation in iot and case wise solution platform

IoT have different functionalities and some dependability as mentioned earlier. In this section, some risks of IoT environment have illustrated with case studies and therefore we can easily point out blockchain benefit.

4.1. Case 1: man-in-the-middle attack

Problem 1: Till date HTTPS and TLS are used for secure encrypted communications for public key infrastructure (PKI) and certificate authorities (CA). Every member has an asymmetric key pair having private/public and the private key is kept by them where public key is publicly available. When a user desires to launch a secure connection, they take off their public key from the broker's source and encode data before sending it, knowing, it can only be decrypted by the receiver because receiver have the private key. So the entire secure connection and system integrity depends upon the principal broker and owner of public keys. For a condition if broker goes down then communications will be unsuccessful. If the system will compromised for some reason then communication system will disrupt. Then attacker is able to accomplish

Table 2
Risk identification according to various layers.

Layer	Natural disaster	Malware	MIMA	BACKDOORA	Bugs	Poor CPU utilization	Bottleneck in data transfer	Loos of backup	Virus
Perception Layer	✓	✓	✓	X	✓	X	✓	X	✓
Network Layer	X	X	✓	X	X	X	✓	X	X
Middleware Layer	X	X	✓	X	X	X	✓	X	X
Application Layer	X	X	✓	X	X	X	✓	X	X
Business Layer	✓	✓	✓	✓	✓	✓	X	✓	✓

MIMA- Man in the Middle Attack.
BACKDOORA-Distributed Denial of Service Attack.

man-in-the-middle attack by providing users a fake key pair and therefore they can able to decrypt sensitive data (Arunkumar et al., 2017).

Solution 1: On the other hand, blockchain is an approach that makes Man in the Middle attacks almost impossible. At the time user announce the public key on the blockchain platform, the data will be circulated over thousands and billions of nodes or blocks, and sender store the link to previous blocks. It will be difficult for hackers to avoid the cryptography that creates the blockchain immutable. So, not a single node will strictly compromise their keys. Fig. 2 is showing possible man in the middle attack during data transmission from sender side to receiver side and Fig. 3 is showing blockchain based recovery from man-in-the middle attack.

4.2. Case 2: backdoor attack

Problem 2: Backdoor attack is one of the tricky attacks to identify. It is an effortless process to inject malware or virus into the network. Sometime this virus affected module is behaved as backdoor; that means this initial module is exploit as a platform for downloading and design the actual attack. For example, when software is developed, some loop hole is created for testing and checking purpose; but this is also a way of backdoor attack if this loop holes are not permanently removed from that specific software. This scenario is depicts in Fig. 4.

Solution 2: Backdoors can be very difficult to detect, and the detection method also varies considerably depending on the computer's operating system. Here we apply block chain that is able to improve system security and trust this is shown in Fig. 5.

Backdoor attack scenario is basically to make a backdoor channel to enter into a target machine without knowing it's owner. The attacker side sends some shell script programming as payload which are able to open particular port of target machine which can be used by the attacker in future, on demand. In our proposed framework the connected IoT sensor nodes are using trendy MQTT protocol to transfer data from sender node to receiver. MQTT protocol only accept some meticulous message to response in order to data transaction. The blockchain framework checks each request carefully and if it is found that the coming request is any kind of script or malware payload then the lightning network mechanism immediately ignore to mining that coming block. Only message and it's hash blocks are allowed to mine the block into network for IoT data

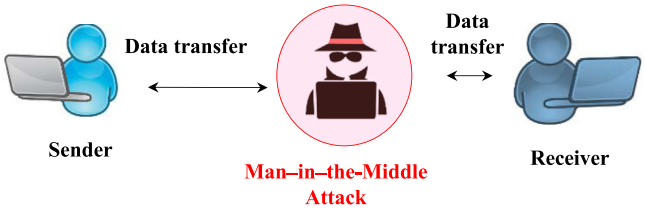


Fig. 2. Man in the middle attack during transmitting data from sender side to receiver side.

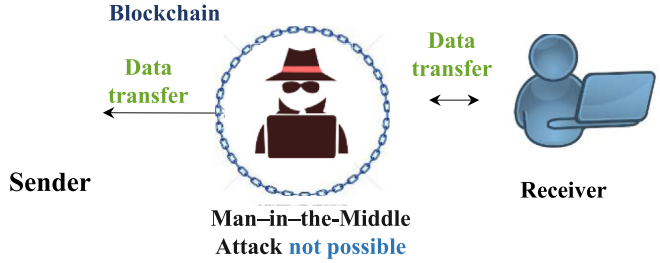


Fig. 3. Man in the middle attack recovery using blockchain while transmitting data.

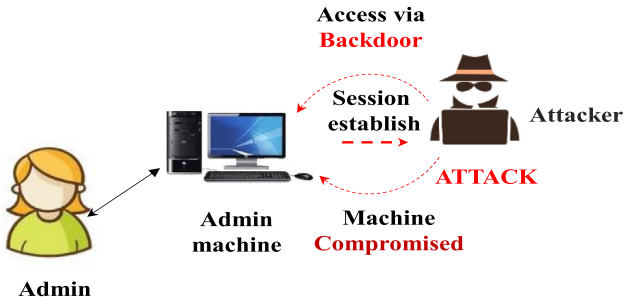


Fig. 4. Backdoor attack during transmitting data without blockchain based security.

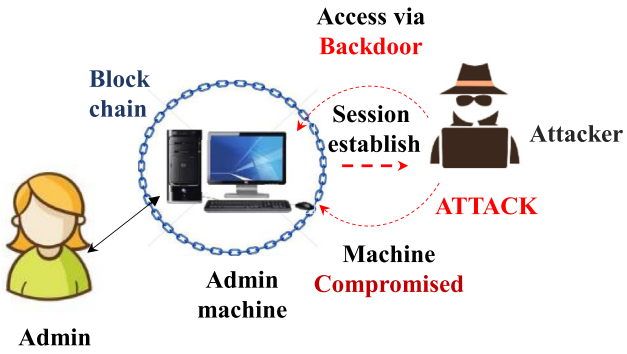


Fig. 5. Recovery of backdoor attack and transmitting data.

transaction. Thus the backdoor attack is prevented using blockchain lightning network into our proposed framework.

Table 2 identifying common risks according to various transaction layers. IoT platform is completely a request/reply progression depending upon the data collected from the real time environment (Roy et al., 2018; Walker et al., 2005; Cai et al., 2014). Thus unavailability of certain data can stop continuous IoT service execution, which can curtail the speed of the computing service and make the service unavailable.

Therefore, in this article particularly man-in-the-middle attack and backdoor attack have focused; both of which causes data loss at the end of the transaction (Zhang and Wen, 2017; Makhdoom et al., 2019). Natural disaster can happen anywhere around the globe that makes the service unavailable. There are several pseudo cloud provider, or malicious insider are present within the cloud platform. Cloud malware is injected into the SQL codes as well as into the transmitted data, which result is service failure. Man in the middle attack is one of the causes that lead to IoT service failure (Zyskind and Nathan, 2015). Distributed denial of service (DoS) attack makes all the resources unavailable for the users. CPU utilization and I/O processing are two important system performance guidelines. The downward performance of any parameter can hamper desired IoT services. Meanwhile, loss of backup from any data centre is alarming and top of that, the IoT service is fell into a dangerous position. Some service providers have false licenses and intend to harm users' security and information should be recognized.

5. Proposed approach

Blockchain has numerous advantages: cryptography, immutability, provenance, decentralized, computing infrastructure responsive, decentralized transaction-processing platform, decentralized database, shared and distributed accounting ledger, software development platform, cloud computing, peer-to-peer network etc. over centralized system. This article considers blockchain to develop a complete secure parallel network for IoT network. Proposed architecture is divided into three major UC sections called public Network, cloud Network, enterprise or parallel network.

5.1. Public network

In a public network domain, different types of users like administrator, developer, operator, and auditor and lastly business users exist. User data of public network are directly transferred to the edge device or servers. This edge servers include service capabilities, need to deliver function and content to the users via the internet and responsible for domain name system (DNS), content delivery network (CDN), firewall and load balancing.

5.2. Cloud network

Proposed cloud network is enriched of blockchain and its application. Here we apply blockchain application which is able to handle all other consecutive applications by taking required information form API manger along with web application and runs into the end user device. Then it directly connects to the API manager and later to the edge server. API manager is responsible to manage blockchain platform. Blockchain platform is built with some high level components like consensus, ledger, membership function, traction, event distribution, communication protocol, crypto currency, secure runtime environment etc. These services are designed with the concern for developing secure and robust communication system. These fundamental services therefore applied in governance procedure. Each policy includes different aspects of security mentoring and intelligence for system monitoring, log analytics for threat detection and avoidance provides a secure smart IoT based service. All these applications are managed by system manager which is also connected to the network level, able to design well maintained service platform. Collected data is transferred to the last level via bridge 2 that is transformation and connectivity level. Transformation and connectivity component enables secure connectivity to the enterprise or other parallel systems. Fig. 6 represents the proposed blockchain based architecture with QoS level security solution. The system is trained to detect attacks, classified the attack and does possible recovery of the duped data after attack, that is shown in Fig. 7.

5.3. Enterprise or parallel network

The enterprise application could be legacy application whom the blockchain application interacts with. Enterprise data include

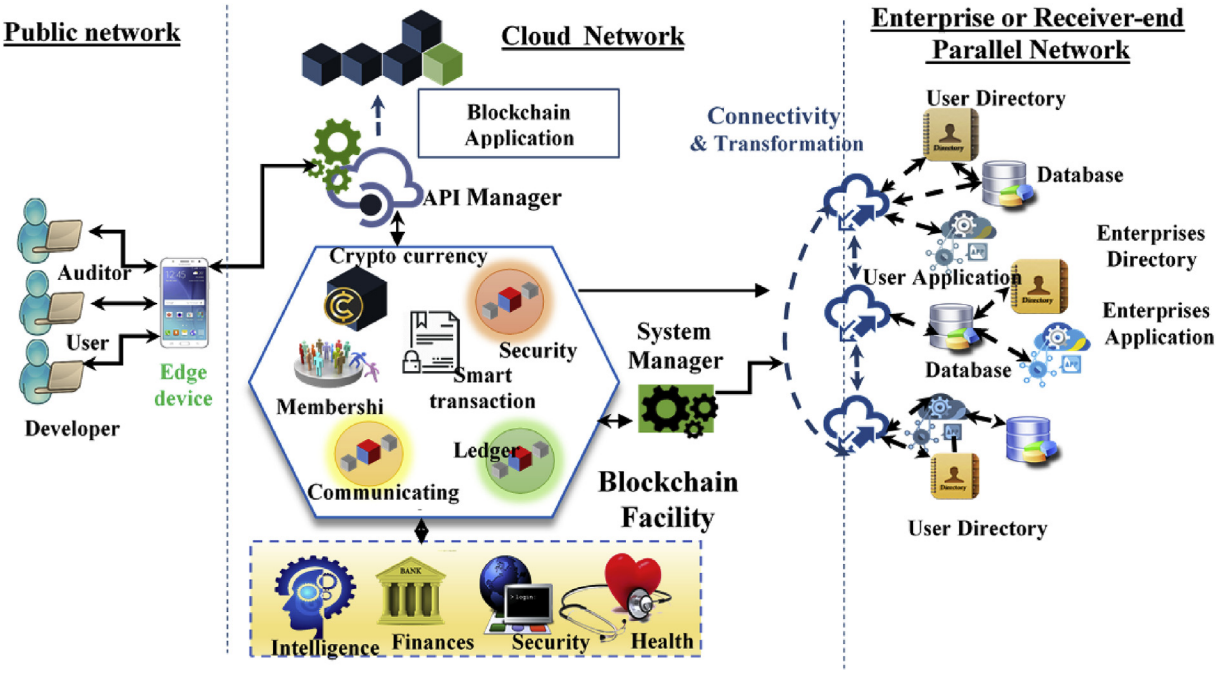


Fig. 6. Proposed architecture-for providing a secure, optimized, QoS analysis based solution.

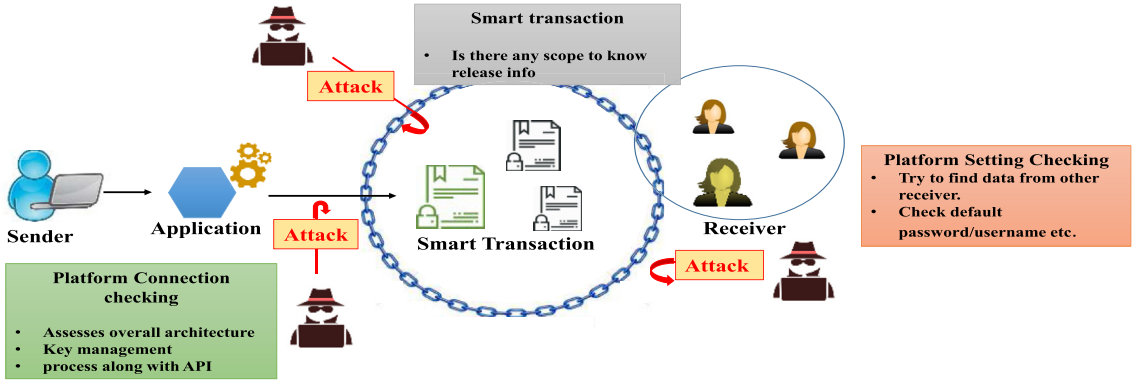


Fig. 7. A secure platform for fast and no involvement of third party.

transactional, application or log data enterprise directory are there into the wireless network to support secure access of the enterprise. Among all others limitations of blockchain, slow data transfer rate is the biggest problem due to slow decision making process by the network manager. To solve this difficulty, here we introduce parallel network for enterprise or data sorting based organizations. This network will select the smallest path with nominal network concession to speed up the overall process.

6. Working procedure

Blockchain is the best tool to make the network secure. Many developers, entrepreneurs and start-ups who are acting as a connector between the administrator and users are able to build platform that can easily disrupt the web's current centralization architecture successfully. Here is just a way to develop application that removes the need of third party monopolies. This proposed work is divides into three major sections; encryption using blockchain, QoS analysis and decryption. Our proposed framework is useful for upcoming industry called industry 4.0 (Yang et al., 2018).

6.1. Encryption

Proposed algorithm prevents the man in the middle attack of IoT service through this encryption phase. The Algorithm 1 is used to encrypt the sensor data which is published by the sensor during the communication period.

Algorithm 1
Encryption Algorithm.

```
Start
{
    Generate a random number ( R_i n) on the Cloud Broker
    Store that random number in cloud
    Send to MQTT broker and devices for data encryption
    Find the active sensor devices (N) present in the  VPN and store it on the cloud.
    {
     For (1 : N rotation)
        { Find the length of the sensor data
            For (1 : R_i rotation)
        {
                For (each character present in  sensor data)
                {
                RightShift(char[i], R_i )
         }
                End For
         }
            End For
        Send the Cipher sensor data to MQTT broker
     }
    End For
    Send all sensor device data to Cloud broker
}
End
```

RightShift (char, no) is the function in which any given character is converted to the ASCII value then right shift it by some numbers “no”. For example, if we perform (MICRO, 3) then it will give us PLFUR. After taking the character, it will convert to the ASCII value of that character and then shift. In the example, 3-bit right shifting is performed.

6.2. Blockchain to ensure security during transaction

After encryption is done this encrypted application is transferred to the blockchain. It follows distributed and decentralized data structure which creates a digital ledger of transactions (DLT) and shares it between disseminated networks.

These blocks of encrypted data are connected with each other's to make an extremely secured distributed database, which is noticeable to the administrator as well as users, except hackers. The database created by numerous people altogether but cannot be altered or updated by any distinct user. This is remove all the probabilities of fraud or third-party participation like man in the middle attack, backdoor attack etc.

With more precisely when a digital transaction is demanded by sender, it is kept in a cryptographically secure block and sent to distributed network, which is also known as the peer-to-peer (P2P) network. This P2P system runs various algorithms to estimate, authenticate and validate the suggested transaction.

If a bulk of nodes agree with a particular transaction and then define transaction as valid — that is, identifying info competitions the block chain's history — then new operation will be permitted and a new block will be added to the chain else the block is rejected. After the transaction is recorded in the block, it is visible to every viewers though cannot be modified. Each individual block, belongs to a chain is connected to the previous and next block which creates it difficult to track a separate record. In order to intrude the security system, the hacker would need to hack each single block contains record that is interconnected altogether,

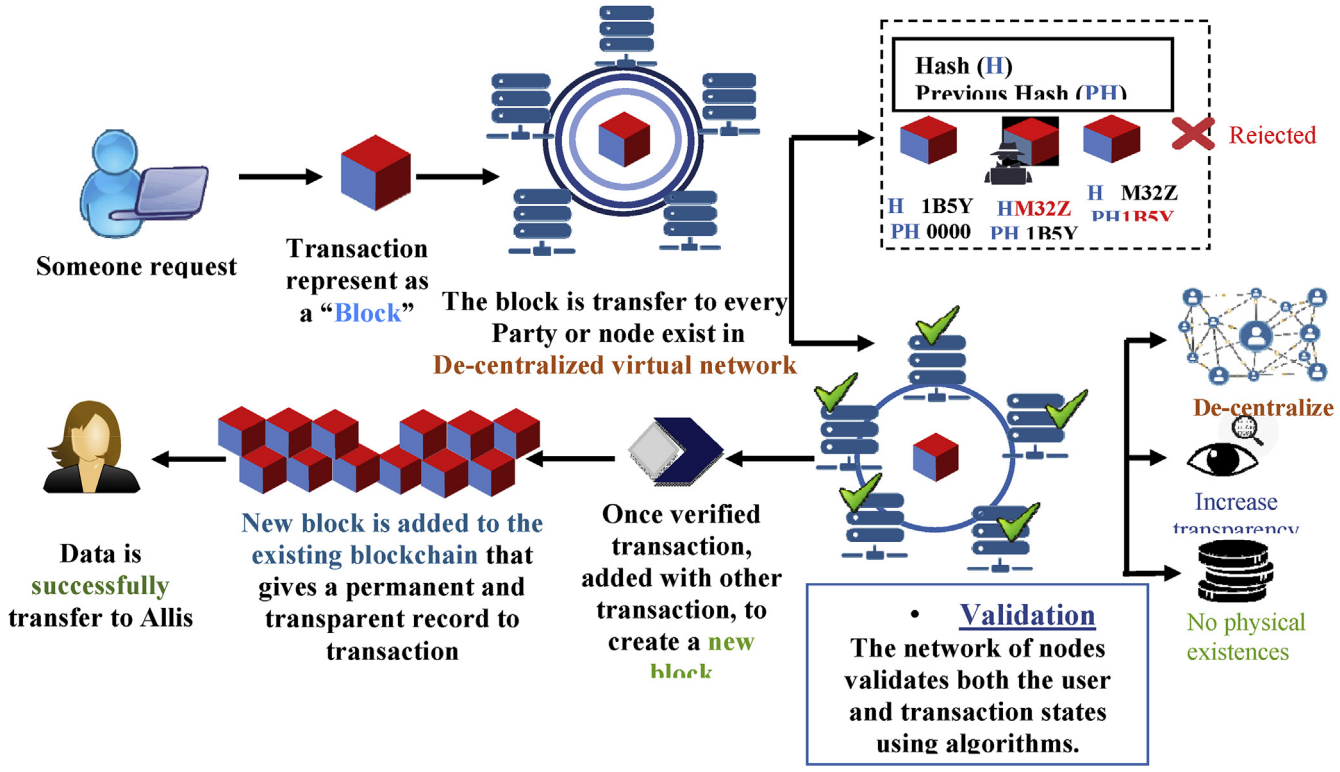


Fig. 8. A secure and step by step details description along with Advantage of blockchain environment.

which is nearly impossible and transaction occurs successfully. Fig. 8 depicts the detailed scenario of secure transaction using blockchain.

6.2.1. Adding data to the blockchain

Basic occurrences of block adding into the blockchain are shown into Fig. 8. In the proposed framework, similarly as bitcoin, a block is joined to the blockchain after a time interval. With the help of *PoW* function, the difficulty regarding time interval for bitcoin has resolved. Inspiring from bitcoin, our proposed network also defines a steady interval to add a new block, named as block period. The network goes through total four different stages of action in this block period. Initially, the transaction phase of circulation starts at timeT_v, transactions are directed according to that called miner_block. This first phase lasts untilT_γ; then may the miner node starts rejecting new connections to form the block. In (Lazarenko and Avdoshin, 2018) the block verification request (BVR) phase review is transferred after collecting the new block from quarry. After verification it returns to the signed block return phase (SBR) (Choi et al., 2016; Lee et al., 2016). After completing the previous phase, the blocks start to add miner_block to its neighbour blockchain, until it comes to the last phase, new blockchain distribution (NBD). The below mentioned Algorithm 2 describes this procedure in supplementary aspect. First start, then elect miner_block with empty set of pending transitions *E*. Transaction is started after that. Although it is noted that the miner cannot mine by itself contacts define in next line, where it is accepted from the set of nodes *X*. The minute of signing the block, to sign at least one transaction is required for a block. This calculation is publicised in next line. In last part of the first section, the concept of blockchain is broadcast to the entire network.

6.2.2. Block mining

Mining means add a node or block to the already existing blockchain. The main aim of the mining is to escape disadvantage of proof of work [PoW] model. In PoW (Mingxiao et al., 2017) computational energy of network is exhausted without making any essential value. Rather, our goal is to meet at system consensus by imposing nodes to deliver proof that the records of a transaction can be done with profoundly understood. However every demanding node has to authenticate these authority proofs. This method is defined below to ensure blockchain stability, but also motivated inter-operability between the blocks in blockchain.

6.2.2.1. Proof of interoperability. Proof of interoperability (PoI) is used to elude some of the shortcomings of proof of work (PoW) model. Specially, it is considered as a controller for the effort necessary to do some essential works like matching the network consensus (Gramoli, 2017): to check the reasonable incoming communications with respect to a known set of semantic and structured constraints. Here, the interoperability parameter is the uniform resource locator (URL) account. As defined earlier, conformation of semantic and structured, permitting organisational constraints on attributes like coordinate and categorised along with semantic constraints uses change of values. Second part of the algorithm 1 describes proposed method in more detail. For transactions, it started to check whether account profile is matched with previously known sets of permissible accounts or not, using Check Account Conformance function account is recognized and get conformance. This action will use to make an authorisation call to the server. The outcome of this method is a response from executed operation that is also checked for conformance using Conforms function.

PoI require network consensus in the set of allowed account, inclusive of the content along with value sets. This consensus reached via programming. Network agreement is an intelligent human-based method, where the system members exchange and co-operate in term of terminology.

6.2.2.2. Block Miner Election. In a PoW consequence, miners help to select right block which will next added to the blockchain. We employ the system using multi chain making procedure. Multi chain has numerous advantages. It is able to point out begin of the block period. Who will be the next miner also is known to the previous node, make the transactions directly. Next, the evenly scattered system need to be kept steady for conform transaction. Lastly, by removing the competition of proof of work, we exclude wasted computational work. This complete method is explained in algorithm 2. In first part of algorithm we have tried to add some blocks in the blockchain for contributing nodes to sign. For signing procedure every node needs to submit a random amount that will use for miner election. According to the algorithm, first all random numbers are collected along with its hashed function to generate new block. Then according to closest match with public key next mining is done. This process prevents a node from producing a non-random figure individually. The miner election procedure is defined in last section of Algorithm 2.

Algorithm 2
Applied Working Principle of Blockchain and Block Miner Election

Input:
X = Set of nodes in network
E = Representing a sequence of (e_0 … e_x) where $e_x$is the present blocks of the blockchain.
T_γ = The completion of the Message Circulation phase
F_r = A valid set of network agreed URIs.
β_a = the current block being assembled.
β_x = The present (last) block on the blockchain,
V_t = A set of valid transactions.
φ = A randomly elected miner block
Start
{
Application transfer as encryption algorithm
{
/*Creating and add a block to the blockchain*/
$\varphi \leftarrow ElectMinerBlock(\beta_x, X)$;
$E \leftarrow \{\}$;
While$T_i() < T\gamma$**do**
  For$x \in X - \{\varphi\}$**do**
    $E \leftarrow E \cup GetTransactionFromBlock(x)$;
$\beta_a \leftarrow Assambleblock(E)$;
/* X' **is all block with** $\geq$ **1 transaction**/
$X' \leftarrow \{x \in X | (\exists t_c)[\in E \wedge IsOriginator(x, t_c)]\}$;
For$x \in X'$**do**
  $SingleBlock(\beta_a, x)$;
$B^c \leftarrow AddBlock(B, \beta_a)$
For $x \in X$**do**
  $DistributeBlockchain(B^c, x)$;
/*Proof of Interoperability*/
$V_t \leftarrow \{\}$;
  For every$t_c \in E$**do**

(continued on next column)

Algorithm 2 (*continued*)

    $U \leftarrow ObtainURL(t_c)$;
    $A \leftarrow ObtainCorrectAccount(t_c)$;
    If$A \in F_r$**then**
/***using Correct URL "Validation" done***/
      $R \leftarrow CheckAccountConformation(U, A)$;
      If*Conformation*(R)**then**
        $V_t \leftarrow V_t \cup \{t_c\}$;
/*Miner Election*/
$S_R \leftarrow ObtainRandomRoot(\beta_x)$;
$H \leftarrow ObtainBlockHash(\beta_x)$;
For every$s_r \in S_R$**do**
  $h \leftarrow Hash(h + s_r)$;
$\varphi \leftarrow where|ObtainPublicKey(x) - h|$**is nominal for ever block x;**
End
}
/***Parallel Optimized and Secure Network***/
After verification is completed it go for find optimized path according to algorithum3
}
After verified application transfer for decrypted by decrypt algorithm
}

6.3. Parallel network connectivity

After verification procedure is done by blockchain it moves to find best secure path through parallel network for completing the transaction.

6.3.1. Check network congestion

For doing the network congestion checking, here K –nearest neighbour is applied first to calculate Euclidian distances between source node (block)$S_i = 1$, to all other nodes; here we define as$F = \{F_1, F_2, F_3 \ldots F_{j1}\}$. After that, top five ($K$) nodes make a group along with all calculated distances. From this top five node distances, the smallest distance is selected for next procedure but here we need to check that either the selected path is congestion free or not, if it gets congestion then according to Algorithm 3, it changes the smallest path and select next minimum value from that top five (Veres and Boda, 2000).

6.3.2. Check trust value

In blockchain, it is accepted that every block or nodes are similarly doubtful and their nature toward decision-making procedure is exclusively depend on the proof-of- interoperability. For more detail, every node$X$, $Trust \infty R_{esources(X)}$selects the node's weightiness in votes. Some of the attacks oblige the system unnecessary energy drainage and system with high-latency. In proof-of- interoperability is used to find which node is coming with decant and minimum resources in the organisation; those are taken less likely to fraud. So here also we state a parameter to calculate trust based on node behaviour. Some predefine values are set to define a good behaviour of a node. Evenly, as it is for a binary random variable, the predictable rate is just the probability. Approximate probability is done by calculating the quantity of good and bad activities taken by the nodes, then by the sigmoid function gets a probability.

$$Trust_F^{(S_i)} = \frac{1}{1 + e^{-\vartheta(\#good - \#bad)}} \tag{1}$$

Algorithm 3
Parallel Optimized and Secure Network.

P_s = Sending packet
ϑ = simply the step size
K =Group of top rank value
$F = \{F_1, F_2, F_3 ... F_{j1}\}$ = Final block
S_i = Sources node or block
P_t = Transmit packet

Start
All = $\{ S_i, F = \{F_1, F_2, F_3 ... F_{j1}\}, K, \vartheta, P_s, P_t \}$
For i= 1 to (F-1) **do**

/Calculate Euclidian distances from i to j=(1…F_j)/
$d(S_i, F_j) = \sqrt{(S_i - F_1)^2}$
End for
Sort sample path in ascending order depending upon the distances from i
Make group of top K=5 ranked sample from sorted neighbour I choose minimum value

/*Check Congestion*/

Initially transmits rate =0;
For (1=1: numberofnode**)**
{
If ($P_s > P_t$ **)**
Then
State = Change path
Else
Go for this path
State= true
}
End for

/*Calculate Trust*/
$Trust_F^{(S_i)} = \frac{1}{1 + e^{-\vartheta(\#good - \#bad)}}$

End

After this calculation, the node gain more weight in network as trusted one and calculate blocks more professionally. Using this method an IoT based network successfully can resist from various backdoor and man-in-the-middle attacks. This might be helpful while selecting different nodes, according by their trust value, to elect on every block and then it will judge with equal majority. This system also prevents a particular node to spare too much influence.

6.4. QoS-parameter aware for cross validation

For validated resource allocation, transmission ratio different type of QoS parameters used have been for cross validation those are availability, reliability, elasticity and responsiveness (Roy et al., 2016).

6.4.1. Availability (A)

Availability of a system is being active or accessible at a particular time. Resources and services are needed for parallel network computing. If the resources are available, we can easily check system access condition before and after attacks. Here we will measure system condition in term of resources availability. Suppose a user system has been assaulted by a hacker through man in the middle attack and after that system has lose maximum memory, CPU and other resources. After using the proposed framework the system can recover. Now we want to show each and every condition of system using QoS monitor application (Ali et al., 2015). It will help to understand the specific system owner in a better manner. The possibility that system is available is specified by Formulae 1 and amount of resources wanted to deliver over desire availability is specified by Formulae 2.

$$A_v = 1 - (1 - \Re)^{\rho}$$
$$\rho = \frac{\ln(1 - A_v)}{\ln(1 - \Re)} \quad (2)$$

Availability lies in the midst of$0 \leq A \leq 1$, whereas$\Re$and$\rho$are signifying the available resource and desire resources needed to satisfy the overall system.

6.4.2. Reliability(R_a)

Reliability defines itself by the system capability to perform a work for specified circumstances within a particular time period. The system contains of memory, CPU cores and storages. The reliability of a machine desires the consistency of these hardware modules. If a single module is miscarried, then a machine will fail to survive. So, here reliability measures depend on QoS basis of hardware modules of servers used for a

machine or pc as specified in Formula 3.

$$R_a = \prod_{m=1}^{p} \prod_{n=1}^{q} 1 - (1 - R_{\beta_i})^n \tag{3}$$

Reliability lies along with$0 \leq R_a \leq 1$, whereas$R_{\beta_i}$is reliability of $m = 1$component from $p$types and $n = 1$to $q$epochs certain module used.

6.4.3. Scalability (S)

The scalability can be defined at the time when work load crosses the system capacity or threshold. Here we used AWS services for calculating this parameter. It can change scale according to assignment. Imagine, an operator fixed his threshold at 85% and if after attack resource crosses its threshold, then a fresh virtual resources is added with the present system as specified in Eq. (4).

After using proposed framework if system backs to its normal condition then extra resources are back to the resource pool.

$$R_{threshold} < \frac{R_{consumed}}{R_{allocated}} \tag{4}$$

Now$R_{consumed}$is the rate of used resource by customer,$R_{allocated}$ is the entire assigned resource and$R_{thresold}$is the threshold rate of exact stock.

6.4.4. Responsiveness (η)

Responsiveness is expressed via how rapid an organisation response after defer to an application. Responsiveness mostly based on the amount of CPU, processing ability along with bandwidth allotted to users. Responsiveness is measured by Eq. (5)

$$\eta = \frac{1 - \int_{i=1}^{n} T_i}{T_{\max}} \tag{5}$$

Now$0 \leq \eta \leq 1$specify responsiveness, $T_i$signifies time of accomplishment also submission of i^{th}application, $T_{\max}$and signifies maximum conventional time to submit an appeal n, is amount of application allotted to process and for the used process function is f.

6.5. Decryption

Decryption algorithm is similar to the encryption algorithm. In that algorithm, the left shift operation is done in the cloud. The Algorithm 4 is used to decrypt the sensor data which are received. **LeftShift(char, no)** is the function in which any given character, is converted to the ASCII value then left shift it by some numbers "no". For example, if we perform **(PLFUR, 3)** then it will give us **MICRO**. After taking the character, it will convert to the ASCII value of that character and then shift. In the example, 3-bit left shifting is performed.

Algorithm 4
Decryption Algorithm.

```
Start
    Retrieve the length of the random number ( R_i ) which was previously generated by the
    Cloud Broker
        For (1 : N Rotation)
                Find the length of received sensor data
            For (1 : R_i rotation)
                For (each character present in sensor data)
                    LeftShift(char[i], R_i )
                End for
            End for
                Decrypt and store the sensor data
        End For
End
```

7. Risk modelling

7.1. Risk analysis

Various constraint are obligatory for developing the model for risk reduction, are represent in Table 3 with their abbreviations.

There are numerous existing restoration mechanisms related to every system failure. The parameter table associated with this risk modelling is shown in Table 2. It is considered that a service hampers due to any cause or an interrupt namely attack if that cause has appeared during providing IoT services to the customers.

Definition 1. Risk occurrence in IoT

$R_i$is a random variable which is define in Eq. (1). The value of the random variable may be 1 when any cause in list occurred with having some probability otherwise $R_i$would be *0*.

$$\begin{aligned} R_i &= 1(C_{ith} cause\_Occoured) && /* \text{ with probability } P_i */ \\ &= 0(C_{ith} causedidnot\_Occured) && /* \text{ with probability } (1 - P_i) */ \end{aligned} \tag{6}$$

Where $0 < P_i < 1$with$i = i_1 \ldots i_2$value$P(R_i = 1) = P_i, \forall P_i \in P$; It is found that the foundation of the R_i follows the **Bernoulli's Distribution** with P_{iI} which signifies the probability of occurrence of the C_i^{th} cause in the list. It is assumed that, all the causes are occurred independently thenR_i is independent distributed. If it occurs then it leads to the IoT service breakdown.

Definition 2. Recovery from the risks in IoT

Lots of recovery mechanisms present to prevent attacks (Makhdoom et al., 2019; Lin et al., 2018). Most of the IoT service failures occur due to the offensive performance of recovery function. We consider$p_r$as recovery probability from the C_i^{th} cause in the list which is mathematically defined in Eq. (2).

Table 3
Some parameter with the expressions.

Parameter name	Abbreviation
R_i	Random variable used to define any cause has occurred or not
p_r	Recover probability from the ith cause
C_i	ith cause in the list occurred
η_i	The cause of the occurs
κ_i	The service is not recovered from the cause of the list
ω	Sum of the all listed failures
P_{ar}	Possibility of IoT service failure after using this approach
P_{br}	Probability of IoT service failure without using this approach

$$P\left(\text{Recover\_from}C_i^{th}\right) = P_r$$
$$P\left(\text{Notrecover\_from}C_i^{th}\right) = (1 - P_r) \tag{7}$$

Definition 3. IoT service failure for risk

The probability of the IoT service failure from that C_i^{th}cause $= P(\eta_i - \kappa_i)$ where $\eta_i$is the cause of the C_i^{th}occurs and$\kappa_i$is that the service is not recovered from the C_i^{th}cause of the list. Eq. (3) represents that the reasons of the C_i^{th}occur, but the service is not recovered from the C_i^{th}cause of the list. Now from the theory of the conditional probability we get,

$$P(\eta_i - \kappa_i) = P(\kappa_i) \times P(\eta_i|\kappa_i)$$
$$= (1 - P_r) \times P_i, \forall P_i \in P \tag{8}$$

Analysis 1. IoT service failure for the identified risks

Here we consider a random variable ξ_i as the Bernoulli distribution variable which indicates the values of $\xi_i$will be *1* if the service is a failure from the C_i^{th} cause in Eq. (4).

$$\xi_i = 1 \quad /* \text{with having probability} (1 - P_r) \times P_i */$$
$$= 0 \quad /* \text{with having probability} [1 - \{(1 - P_r) \times P_i\}], \forall P_i \in P \tag{9}$$

Now we calculate the sum of the all listed failures which are consequences of the listed threats before using this approach, which is denoted by the variableω_{before}. $\omega_{before}$is calculated by adding all the $\xi_i$value which is nothing but the sum of all the failures.

$$\omega_{before} = \sum_{i=1}^{12} \xi_i \tag{10}$$

The value of ω_{before} can vary between 0*to*12. We know the value of $\omega_{before}$from Eq. (5) and use it to find service failure from any cause before using this approach. The probability of the IoT service failure from any listed causes is calculated by using Eq. (6).

$$P(\omega_{before} > 0) = 1 - P(\omega_{before} = 0)$$
$$= \prod_{i=1}^{12} (1 - P_r) \times P_i; \forall P_i \in P \tag{11}$$

The probabilities of P_i and P_r are to be estimated, in which P_i can be easily found out by using the Poisson distribution model to the obtained data of the failures and every cause is fitted as the Poisson variable. Variable $P_r$can be calculated by using the normal approximation of the failures.

Analysis 2. IoT service failure after using this approach

After using this approach, we try to mitigate the man in the middle attack from the IoT services. 11 different causes have remained for which IoT service may fail. In Eq. (7), we calculate the sum of the all listed failures after using this approach by the variable $\omega_{before}$with the help of Eq. (4).

$$\omega_{before} = \sum_{i=1}^{11} \xi_i \tag{12}$$

Now the value of ω_{before}can take values as $0, 1, ..., 11$. The probability of the IoT service failure from any cause after using this approach is calculated by using Eq. (6). The probabilities of $P_i^{'}$ and $P_r^{'}$ are to be estimated, where $P_i^{'}$ can be easily found out by using the Poisson distribution model and$P_r^{'}$can be calculated by using the normal approximation. The probability of the IoT service failure from after using our approach is shown in Eq. (8).

$$P(\omega_{after} > 0) = 1 - P(\omega_{after} = 0)$$
$$= \prod_{i=1}^{11} (1 - P_r^{'}) \times P_i^{'}; \forall P_i^{'} \in P^{'} \tag{13}$$

7.2. Risk reduction

Proposition 1. Risk reduction after using the proposed approach

Relative risk measures how much risk is reduced by using our experimental approach compared to the typical existing approach. $R_i$denotes the probability of IoT service failure after using proposed approach and R_i indicates the probability of IoT service failure without using proposed method. $R_i$is calculated by using Eq. (8) and R_i is calculated by using Eq. (6). The relative risk reduction is calculated by using these two equations. The formula of relative risk reduction is shown in

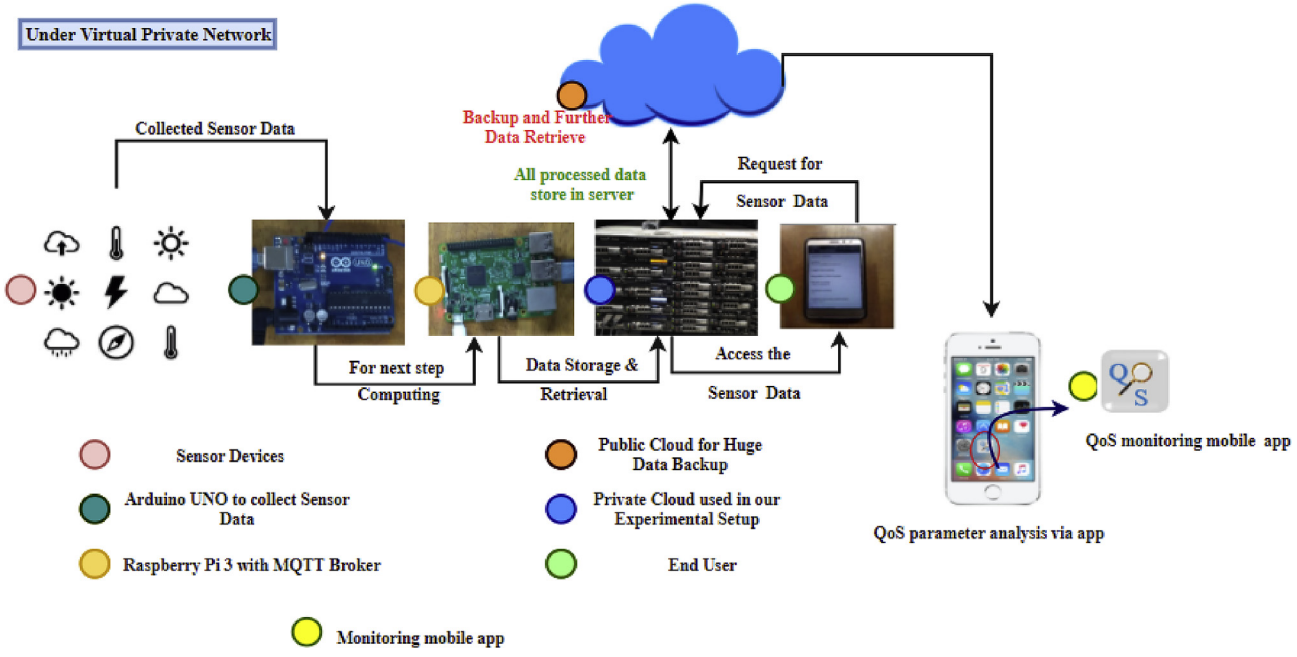


Fig. 9. Experiment setup for collecting sensor data, upload and monitoring via QoS monitoring application.

Eq. (9). Relative risk reduced by our approach is formulated by the following:

$$\text{Related Risk Reduction} = \frac{P_{br} - P_{ar}}{P_{br}} \tag{14}$$

Now we calculate the relative risk reduction by subtracting Eq. (8) and Eq. (6). And divide the result by Eq. (6). Relative risk reduction is formulated by Eq. (10).

$$\begin{aligned} \textbf{Related Risk Reduction} &= \frac{P(\omega>0) - P(\omega_{after} > 0)}{P(\omega>0)} \\ &= 1 - \frac{P(\omega_{after} > 0)}{P(\omega>0)} \\ &= 1 - \frac{\prod_{i=1}^{11}(1 - P_r^{'}) \times P^{'}}{\prod_{i=1}^{12}(1 - P_r) \times P_i} \end{aligned} \tag{15}$$

Here we assumed that all the causes' occurrences are not dependent on each other. Some further study can be done on the interdependence of those causes. This analysis has to be done by the MQTT broker. Every service has different$P(\omega > 0)$ values which are considered as vulnerability quotient of that service.

8. Experimental setup

Phase 1. Sensor data fetch and upload

It is revealed that among all attacks, the man in the middle attack is the maximum damaging in the IoT layers while exchanging data between the users and providers (Lee and Lee, 2017). In Fig. 9, the block diagram of the experimental setup has been proposed for demonstration our encryption and decryption algorithm. Some basic sensors like heat sensor, temperature sensor, humidity and light sensor have been used to take data from the environment, and after that, the sensed data are sent to the Arduino UNO. After processing those data, Arduino UNO sends those data to Raspberry Pi 3 for storing them within a file. Next, the data file has been transferred to the cloud server for further processing by using the MQTT protocol with QoS level 2. A virtual private network (VPN) has been utilised for this whole experimental setup. The public cloud servers have been used for back up.

To accomplish the experiment a four pins temperature sensor with having accuracy level plus/minus 2 °C and three pins heat sensor are used to sense and collect data from lab environment. Both the sensor has the working voltage of 5 V. For the experiment, the Arduino Uno is used, which is nothing but a microcontroller board based on the ATmega 328, having 6 Analog input pins, 14 digital I/O pins, a power jack, a USB connection, a reset button and a 16 MHz resonator. Sensors are placed according to the block diagram of the experiment with proper connectivity and power. Raspberry Pi 3 is a powerful credit-card sized single board computer can be used for several applications and have the ability to run multiple programs at a time. It has USB connection port, storage space, and an operating system. Comparing to Arduino Uno, it has more storage and processing capabilities. Amazon Web Services and OpenStack acted as a public and private cloud respectively. All the data are stored in the database of the AWS cloud provider. The AWS Cloud provides us with a set of infrastructure services, such as storage options, databases, computing power, and networking options. OpenStack is a cloud that controls large pools of storage, compute, and networking resources throughout a data centre. All tasks are managed through a dashboard with the help of a web interface that gives administrators to manage those resources. To publish the collected real time IoT data here MQTT protocol has used.

Phase 2. Blockchain Implementation using MQTT

In this section, a basic demo blockchain server and a blockchain client using Python are implemented. This proposed blockchain framework is ensuring these features: transactions of real time encrypted IoT data using RSA encryption technique, simple conflict resolution between connected active IoT nodes from the lightning network, opportunity of adding numerous IoT nodes to the blockchain, proof of work (PoW). Blockchain client ensuring some features are: wallets generation for IoT data handling by Public/Private key encryption, making transactions between active IoT blocks with encryption. We will also perform two dashboards: one for mining the blocks using created hash named as blockchain frontend for miners. Other one is for the users named as blockchain client. Using this dashboard customers are able to produce wallets for IoT data storing purpose and they can make their secure

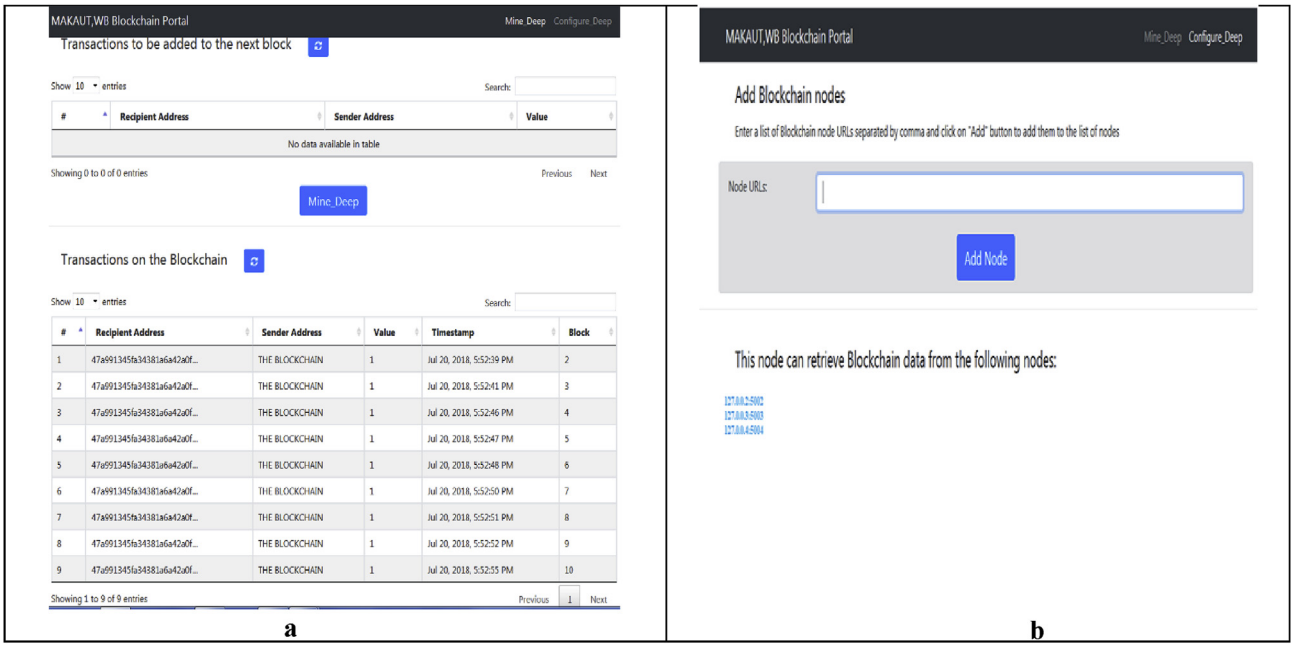
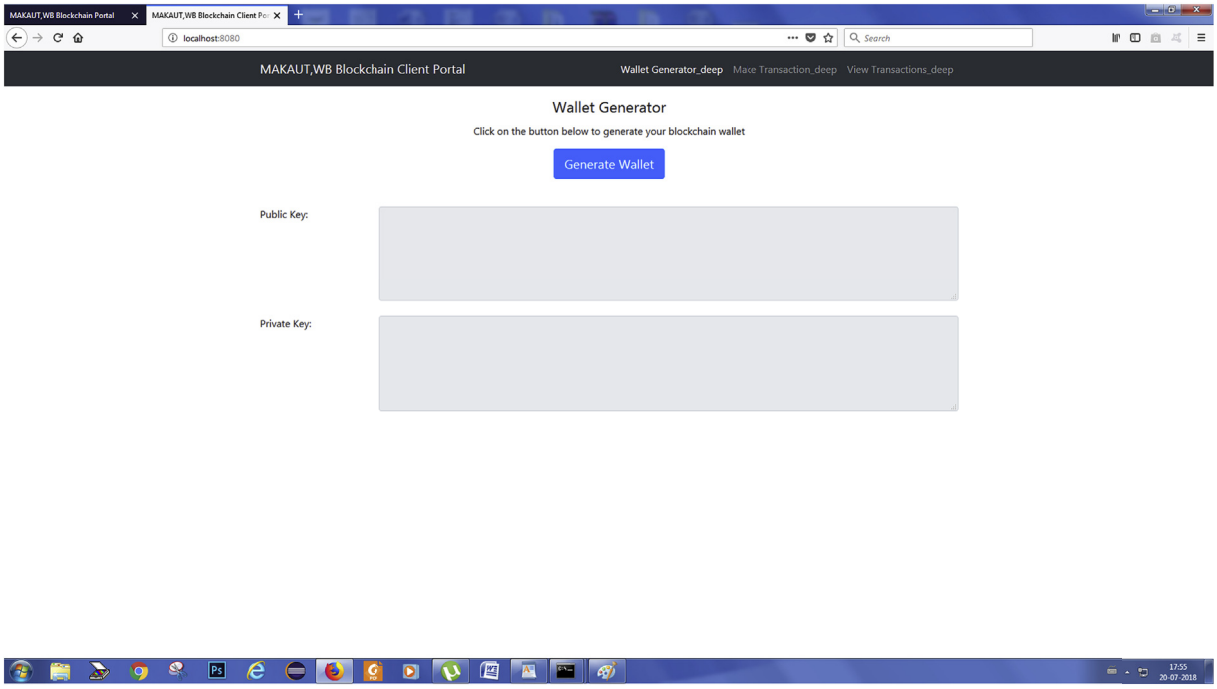
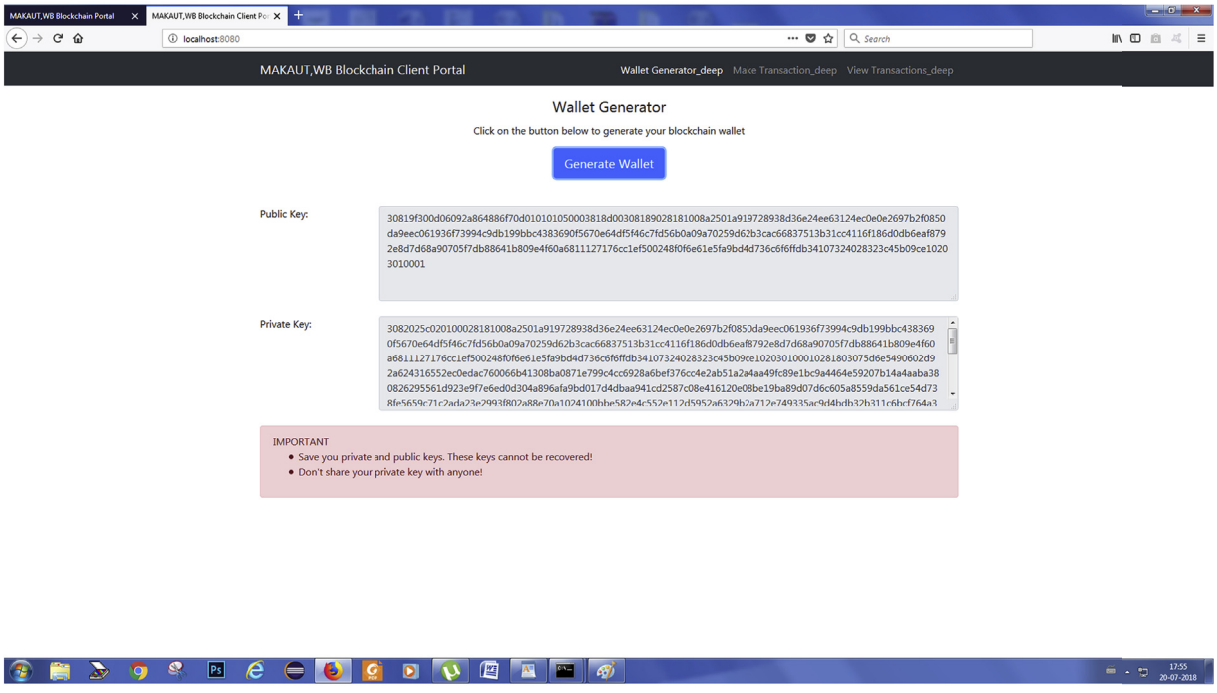


Fig. 10. (a)View mining after and before transaction. (b)View configuration of mining block.



a



b

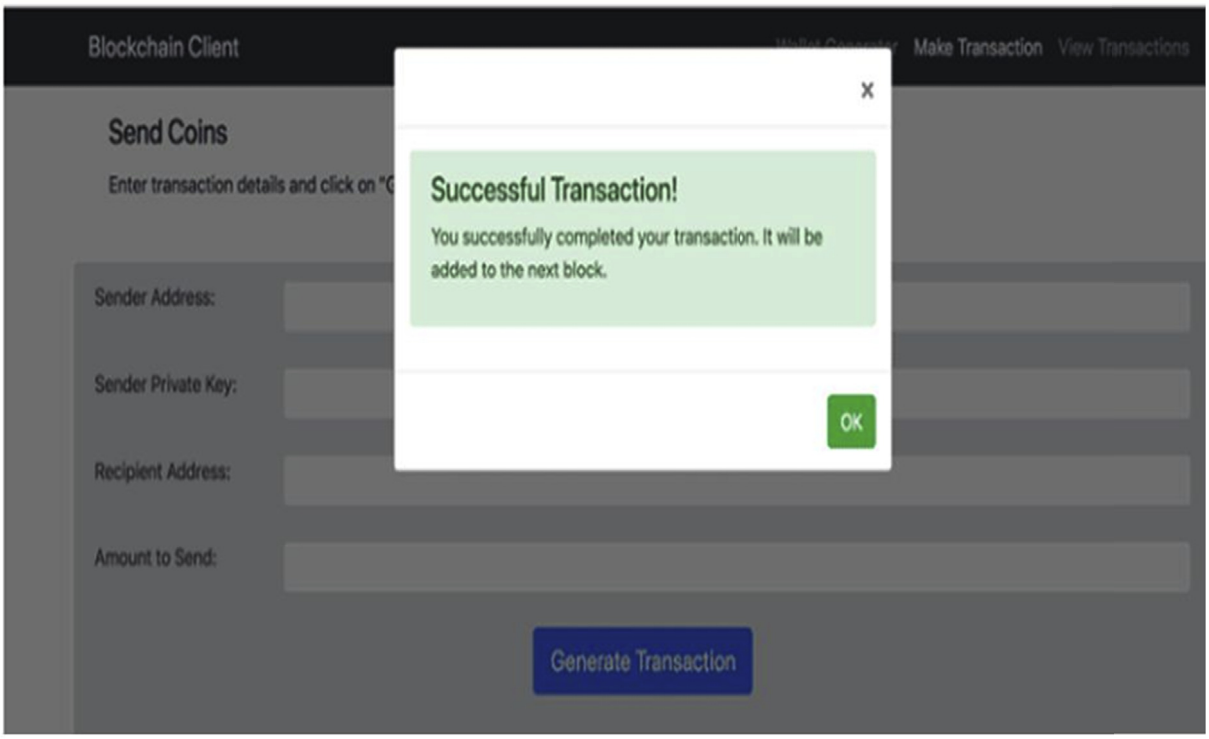
Fig. 11. a before Key pair generation in client portal. Fig. 11b. After Key pair generation in client portal. Fig. 11c. Make transaction using generated Key pair. Fig. 11d. After transaction to view.

transaction. Jupyter notebook is used for blockchain based IoT data wallet generation and to make encrypted transaction. To design the dashboards we have used HTML, CSS.

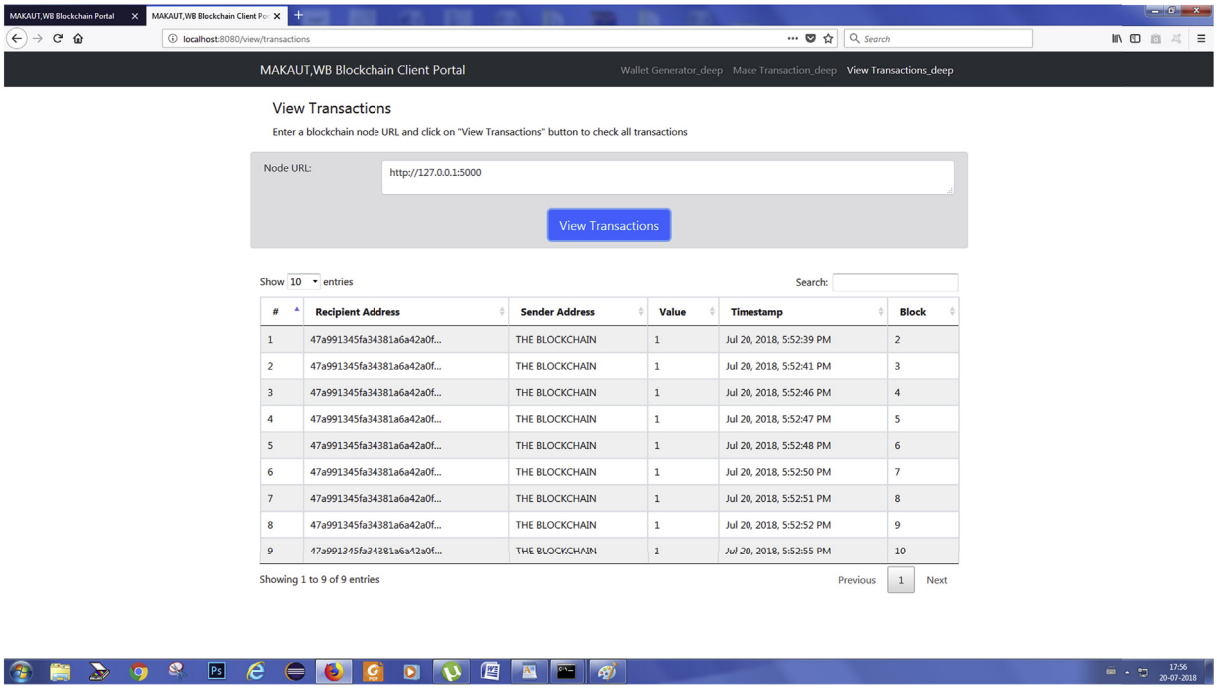
8.1. Blockchain Implementation for miner

From terminal, to create an active blockchain IoT node first have to go the blockchain folder, and compile blockchain_client or along with -p < PORT NUMBER>, the default to port 5000. In browser, set http://localhost:<PORT NUMBER > to see the blockchain frontend dashboard. The dashboard contain two tabs in the navigation bar:

- Mine: To view IoT data transactions and data along with to mine new IoT blocks.
- Configure: To establish connections among the different blockchain nodes.



c



d

Fig. 11. (*continued*).

The blockchain class having some attributes are described in Fig. 10a and Fig. 10b. The attributes are: transactions means the list of active IoT data transactions which will added with the next block; Chain is an array of blocks present in proposed IoT blockchain; node defies a set holding node of URL to access the particular IoT block. Here blockchain is used for these nodes to retrieve data from other nodes and updates if they're not in sync lastly having the node_id, and the hash. To recognise the IoT blockchain nodes a random string is introduced.

8.2. *Blockchain client implementation*

From terminal, to start the proposed IoT blockchain client dashboard,

first have to go to the blockchain_client folder, and compile the blockchain_client Python script. In browser have to write http://localhost:8080 and dashboard will run like Fig. 11. The dashboard contain three tabs in the navigation bar which are shown in Fig. 11a and Fig. 11b, c, Fig. 11d.

- **Wallet Generator**: To generate wallets for secure IoT data transaction, a pair of keys are being generated as Public and Private key using encryption algorithm
- **Make Transaction**: To produce IoT data transactions and send it to an active block of the network.
- **View Transactions**: For checking the committed transactions that are made on.

In order to make or view transactions, at least one IoT block node must be running. A python class has defined for blockchain based transaction attributes. They are sender_address, sender_private_key, recipient_address, and value. These information are obligatory to a sender to generate the IoT data transaction.

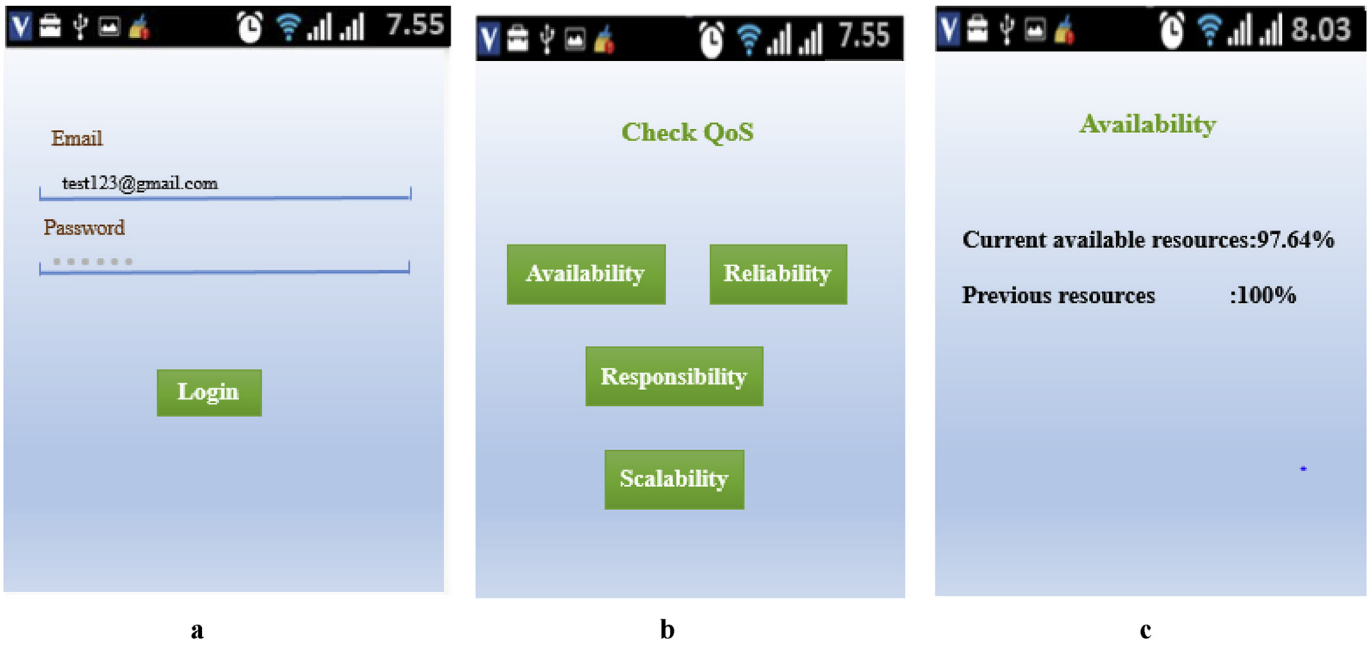


Fig. 12. a. Login page of QoS application. Fig. 12b. QoS available options. Fig. 12c. Availability checking of resource before and after attack.

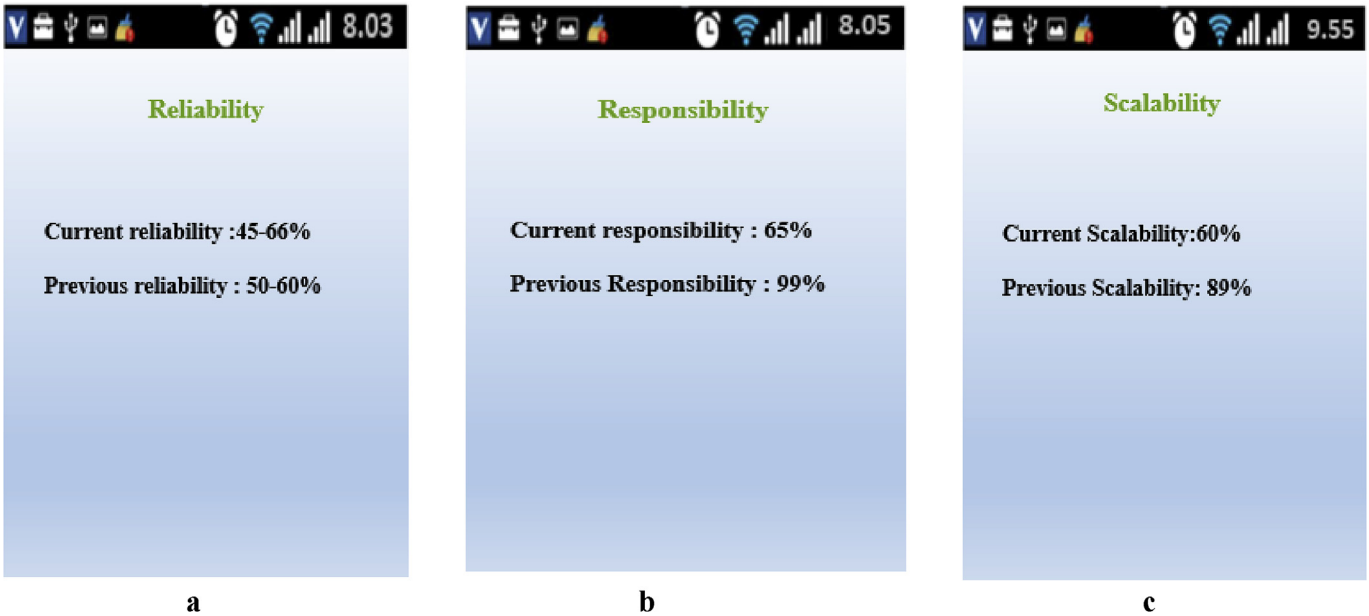


Fig. 13. a. Reliability checking of resource before and after attack. Fig. 13b. Responsibility checking of resource before and after attack. Fig. 13c. Scalability checking of resource before and after attack.

Phase 3. Quality of Services Checking Using Mobile Application.

For quality of service checking phase, we develop a small android application namely QoS monitor for better desired service. According the objective to develop a convinced system, a small experiment is done with some attack like man in the middle attack, backdoor attack are injected using python through kali tools. After terminate the attack, operating system has used 90% of CPU utilization, cache memory was full and some resources also nearly damaged. Where Fig. 12a and Fig. 12b show the login page and all the options are provided by the prototype mobile application. Options are namely availability, reliability, responsibility and scalability. Figs. 12c–13c depict the availability, reliability, responsibility and scalability respectively.

Therefore a time period is nessesary by the QoS assurance application, to check avilibility of the resources, system responsibility, relibility and scalability. After chacking the strength of desired QoS parameters, we repeate this attacking procedure once again on our encryption-decryption algorithm to find the changes onthe system. After and before using same attack on our method gain the availability, responsibility, scalability and reliability is recorded with the help of developed mobile application. Those scanpshoots are given for better understanding.

9. Result and discussion

According to the proposed experiment scenario which has already introduced in section 6 with its set up, is finally explored with the generated graphs. Arduino Uno collects sensor data from the environment and sent those data to Raspberry Pi 3 by using MQTT broker to store and analysis those data. More than one Arduino Uno can send the environmental data to a single Raspberry Pi at the same time. After that, Raspberry Pi 3 sends those data to cloud server with the help of cloud broker. With the installed mobile application we can monitor those sensor data by our smart phone.

Sample temperature, humidity and heat measurements are indexed in Table 4. The MQTT broker sends those sensor data to Raspberry Pi 3 which is shown in Fig. 14. In Fig. 15 the heat index graph is shown with respect to time. Further, these collected sensor data has used in the discussion part of the experiment. Comparison between the proposed method with the existing brokering models are presented in Table 5.

In Fig. 16 and Fig. 17, the light sensor and tampered light sensor data are shown. The deviation of the curve is happened due to the man in the middle attack is done by fabricated emulation intentionally, using Kali Linux and its tools.

Here we discuss case wise solution according to our experiment of different type of attacks. For this list Man in the Middle attack comes first. Problem and its solutions using our proposed algorithm are discussed.

Table 4
Showing resultant sensor data after experiment is done.

Humidity	Temperature	Heat index
56.00	26.00	26.78
55.00	26.00	26.08
54.00	26.00	26.07
53.00	26.00	26.04
53.00	26.00	26.01
52.00	26.00	26.00
51.00	26.00	25.98
50.00	26.00	25.98
50.00	26.00	25.88
49.00	25.00	24.09
49.00	25.00	24.08
49.00	25.00	24.06
49.00	25.00	24.09
49.00	25.00	24.02
49.00	25.00	24.00

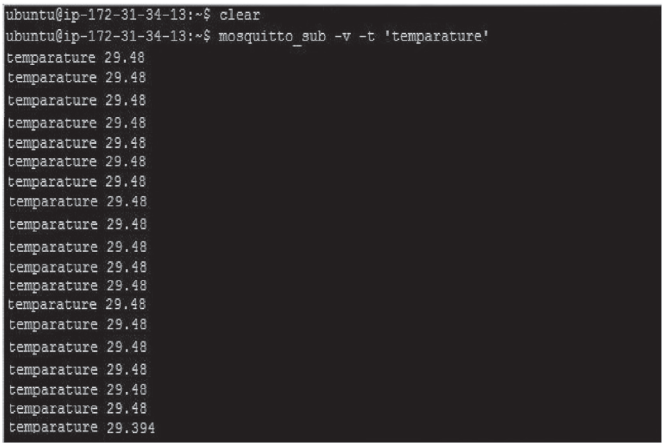


Fig. 14. Experiment result as temperature data is fetch.

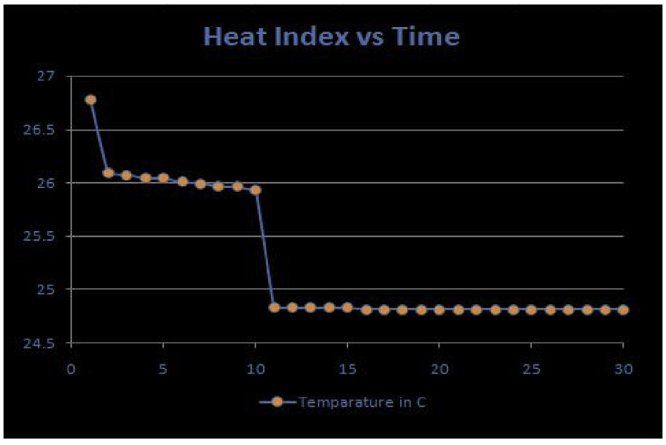


Fig. 15. Graphical relation between heat index and time.

9.1. Case 1: man in the middle attack using Kali Linux and its tools

For demonstration purpose, man in the middle attack is done by our side intentionally, using Kali Linux and its tools. Initially temperature and light sensor data are shown as normal in Fig. 18 but after attack has done some deviation of the curve has happened. In Fig. 19, the temperature sensor data is processed by using our proposed algorithm and no deviation of the curve is visible; from which we can say that no man in the middle attack takes place. Equation (14) shows that some risk has been reduced by our approach.

9.2. Case 2: backdoor attack

For demonstration purpose, we consider application layer based http flood attack (backdoor attack) by our side intentionally, using Python. First time 20 packets per second were sent to the targeted IP address using http protocol for 12 min (with an interval of 1 min) and noted the end-to-end delay. Next we sent 50, 110 and 150 packets and noted the end-to-end delay respectively, but for the 150 packet set the targeted system was crashed (system turned-off) because of memory exhaust. Initially temperature and light sensor data are shown as normal but after attack had done deviation of the curve is happened are shown in Figs. 18 and 19. In Fig. 20 and Fig. 21, the temperature sensor data is processed by using our proposed algorithm and no deviation of the curve is visible, from that we can say that no man in the middle attack took place. Equation (14) shows that some risk has been reduced by our approach. We also recorded the CPU utilization of the targeted system before and after attack.

Table 5
Comparison of proposed method with the existing brokering models.

Properties	A QoS broker for hybrid cloud	BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0	Our strategies
Contribution	Focus on security consideration for IoT from the perspective of cloud computing paradigm.	Develop a theoretical blockchain-based system namely BSeIn, for distant mutual verification for fine-grained, suitable for Industry 4.0 placements.	This article forecasts the security issues solve using blockchain associated with trust based congestion control QoS monitoring system to get secure system.
Applied Hash function for Security	X	✓	✓
Recover system after attack	X	X	✓
Trust calculation	X	X	✓
QoS parameter calculation	✓	X	✓
Risk Modelling	X	X	✓
Use MAC,CL-MRE for protection	✓	✓	X
Risk Identification& Reduction	X	X	✓
Case Study Of Attacks	X	X	✓

The intension is to apply attack on system is, first to check either this encryption decryption algorithm is sustain or able to work after get attacked. Second, using QoS parameter check the network condition; either after attack availability, reliability, responsibility, scalability and security is functioning properly or not. If condition is true then proposed algorithm and procedure is correct for entire system. As a demo, blockchain was implemented, from that we made transaction and according with that a graphical representation is obtained. Fig. 22 shows total number of transitions with respect of time and Fig. 23 shows confirm transactions per day made by using developed blockchain security.

We used python and Kali as an emulating interface. We implemented the man-in-the-middle attack so many times to affect QoS parameters (specifically availability and reliability) to show the results. In

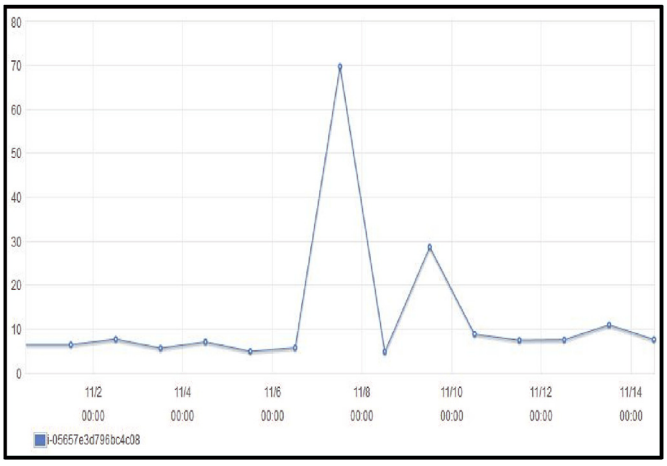


Fig. 16. Experiment result as light sensor data.

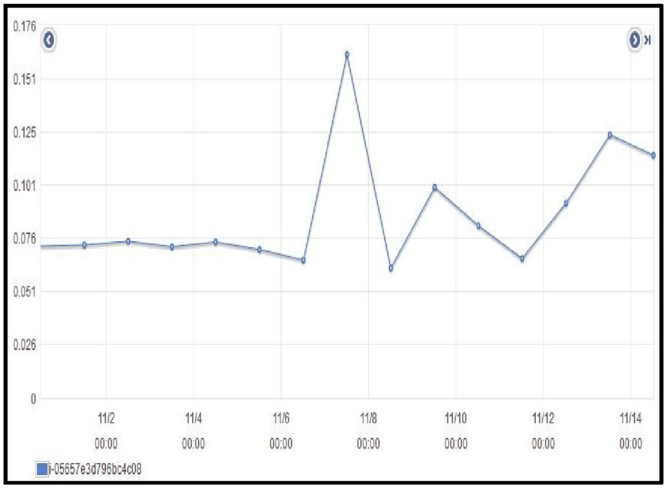


Fig. 17. Experiment result as tampered sensor data.

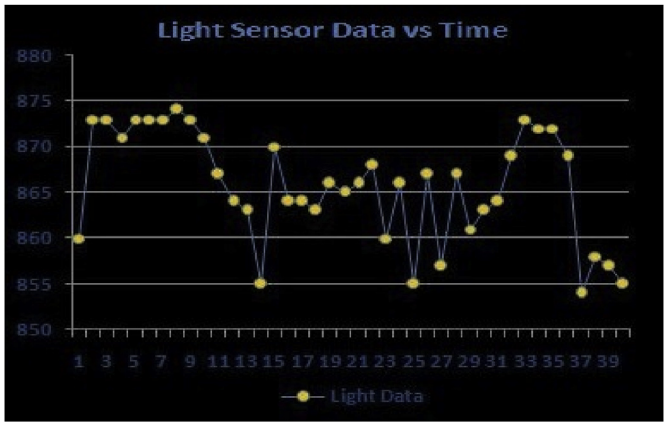


Fig. 18. Light Sensor Data before Man in the Middle attack has done.

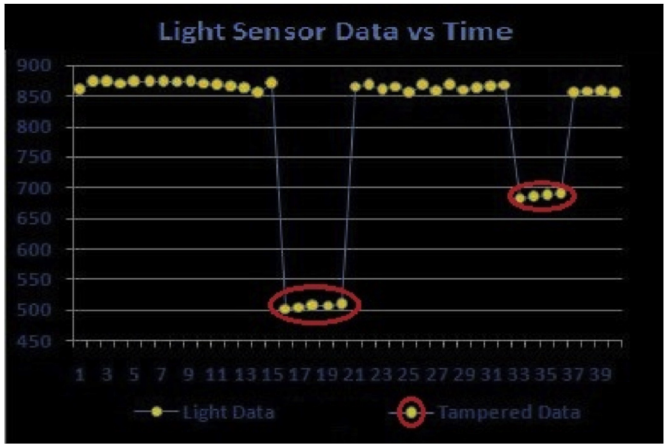


Fig. 19. Light Sensor Data after Man in the Middle Attack has done.

availability we will check CPU usage after attack and attack after apply proposed method. In Fig. 24 and Fig. 25, the CPU usage and available resources in graphical way are shown.

We can see that three scenario more CPU has utilised after attack and respectively less usage and more resource is available after applying proposed algorithm. In Fig. 27, availability and reliability (R) are working vies-versa. After calculating availability of resource it is proved that QoS parameter is compulsory and we can recover a system after attack through analysis of some QoS parameters.

Fig. 26 shows comparison of end-to-end delay during normal transmission applying MITM and BACKDOOR attacks. Reliability, responsibility and scalability are done using AWS instants and resources earlier (Huh et al., 2017; Lin et al., 2018), however we have done with different reliability of system constituents are; CPU, RAM and storage.

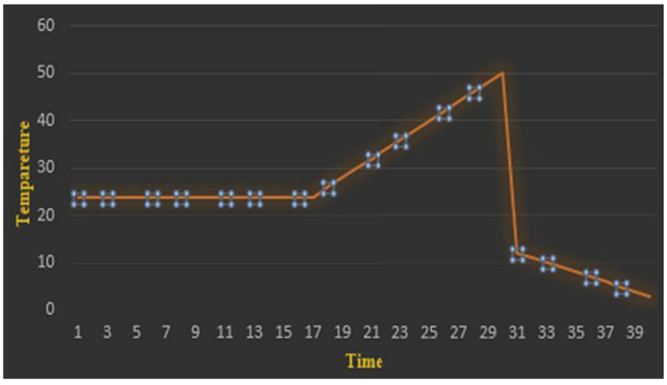


Fig. 20. Temperature data before attack.

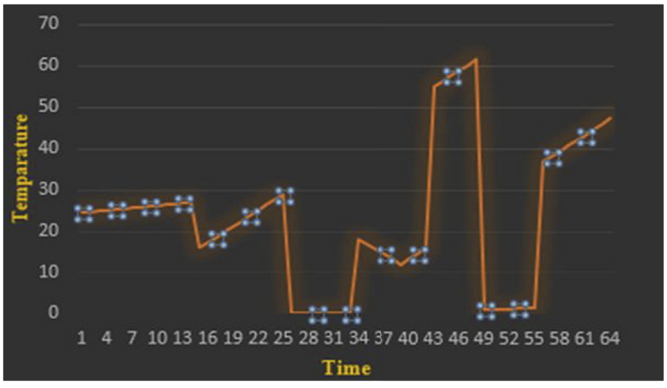


Fig. 21. Temperature data after attack.

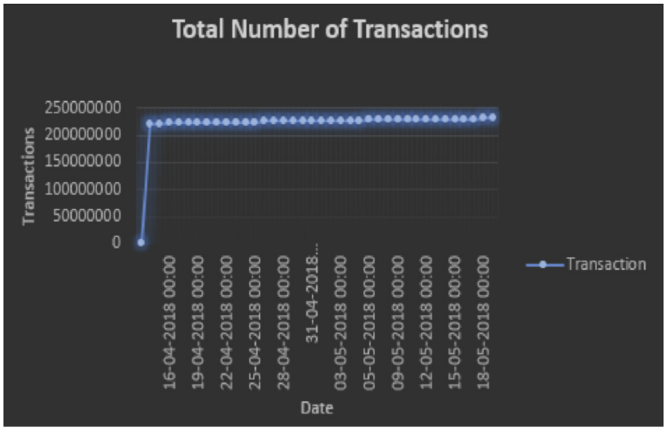


Fig. 22. Total number of transaction done in demo blockchain.

We can notice that proposed blockchain based QoS assessment acquires up to 97.64% system utilization; 100% on RAM and 78.7% on storage recovery.

10. Conclusion

Risk analysis and risk management are the foremost concerns in the growth of internet of things paradigm. As variety of vital information

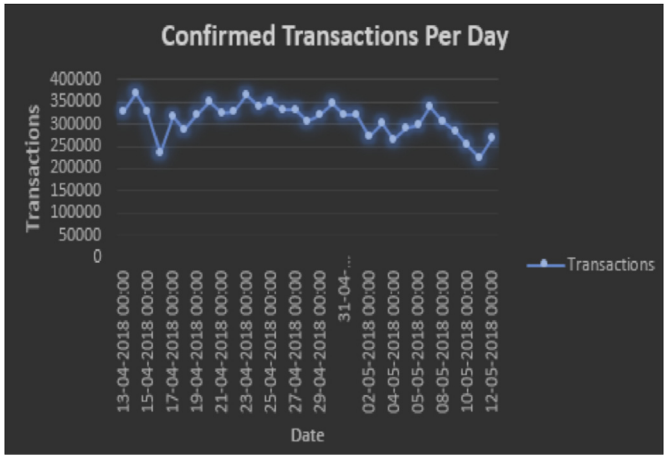


Fig. 23. Calculate number of valid transaction done in demo blockchain.

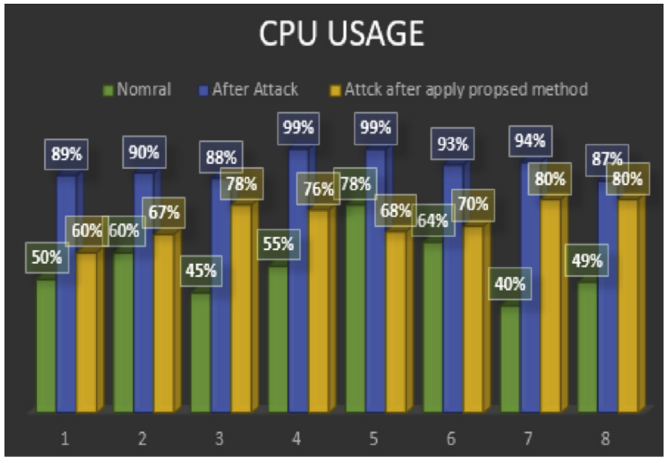


Fig. 24. CPU utilization after attack.

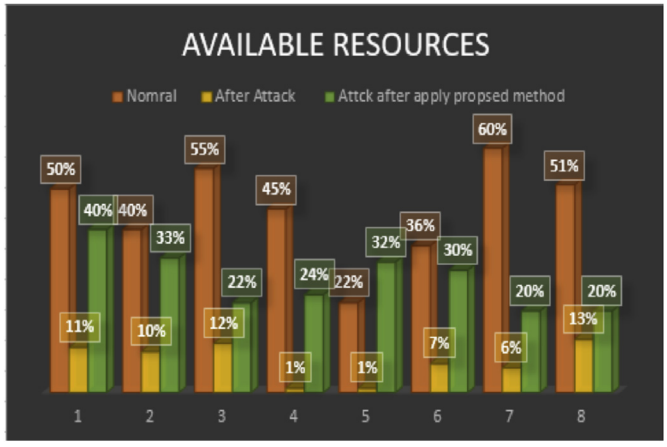


Fig. 25. Available resources after attack.

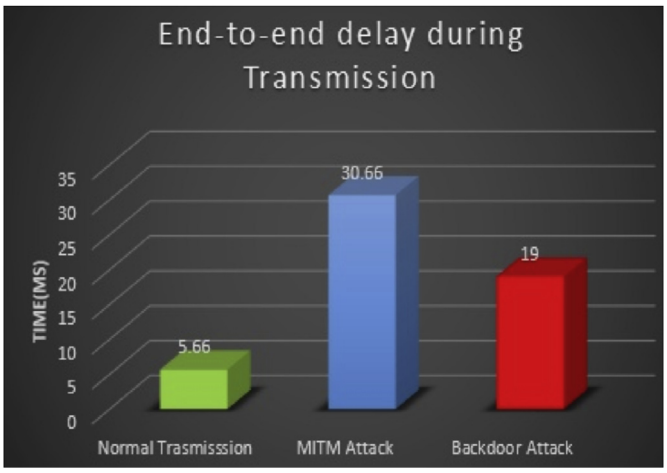


Fig. 26. Compare end-to-end transmission delay.

Fig. 27. Overall reliability calculated after recover system based on CPU utilization, cache status and storage.

from the environment are collected through billons of sensor facilitated small devices. The main goal of this paper is to provide a complete solution over IoT related attacks. Risk identification along with risks prevention using encryption and decryption algorithm and mainly use of blockchain is done here for IoT services. In this article we have created synchronization between risks and resources. Identification of the risks and solution are discussed here. We have performed risk analysis and risk modelling in IoT layer also demonstrated the mechanism of preventing man in the middle attack, backdoor attack in IoT using blockchain. An experimental analysis of the proposed approach has been performed. By using MQTT protocol and virtual private network, the man in the middle attack is prevented successfully. The proposed approach reduces the relative risk along with monitoring section is done by QoS mobile application of 97.64% system utilization, 100% on RAM and 78.7% on storage recovery. This application provides the system condition before and after attack which is really need to recover a system of the IoT prototype. The field of IoT requires massive research effort to tackle the security challenges, but it can provide economic, professional and personal benefits in the future. Thus we can conclude that our approach makes the Internet of things paradigm more secure by reducing the risks of attack as well as by recovering the system even after the attack.

References

Abie, H., Balasingham, I., 2012, February. Risk-based adaptive security for smart IoT in eHealth. In: Proceedings of the 7th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 269–275.

Ali, M., Khan, S.U., Vasilakos, A.V., 2015. Security in cloud computing: opportunities and challenges. Inf. Sci. 305, 357–383.

Andersen, M.P., Fierro, G., Culler, D.E., 2017. Enabling synergy in iot: platform to service and beyond. J. Netw. Comput. Appl. 81, 96–110.

Arunkumar, S., Soyluoglu, B., Sensoy, M., Srivatsa, M., Rajarajan, M., 2017. Location attestation and access control for mobile devices using GeoXACML. J. Netw. Comput. Appl. 80, 181–188.

Aslam, S., ul Islam, S., Khan, A., Ahmed, M., Akhundzada, A., Khan, M.K., 2017. Information collection centric techniques for cloud resource management: taxonomy, analysis and challenges. J. Netw. Comput. Appl. 100, 80–94.

Cai, H., Da Xu, L., Xu, B., Xie, C., Qin, S., Jiang, L., 2014. IoT-based configurable information service platform for product lifecycle management. IEEE Transactions on Industrial Informatics 10 (2), 1558–1567.

Choi, B.C., Lee, S.H., Na, J.C., Lee, J.H., 2016. Secure firmware validation and update for consumer devices in home networking. IEEE Trans. Consum. Electron. 62 (1), 39–44.

Clifton, Chris, Kantarcio_glu, Murat, Doan, AnHai, Gunther, Schadow, Vaidya, Jaideep, Ahmed, Elmagarmid, Dan, Suciu, 2004. Privacy-preserving data integration and sharing. In: Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. ACM, pp. 19–26.

Ferrag, M.A., Maglaras, L., Argyriou, A., Kosmanos, D., Janicke, H., 2018. Security for 4G and 5G cellular networks: a survey of existing authentication and privacy-preserving schemes. J. Netw. Comput. Appl. 101, 55–82.

Gai, K., Qiu, M., Sun, X., 2018. A survey on FinTech. J. Netw. Comput. Appl. 103, 262–273.

Gramoli, V., 2017. From blockchain consensus back to byzantine consensus. Future Gener. Comput. Syst.

Huh, S., Cho, S., Kim, S., 2017, February. Managing IoT devices using blockchain platform. In: 2017 19th International Conference on Advanced Communication Technology (ICACT). IEEE, pp. 464–467.

Kim, E., Chung, K., Jeong, T., 2017, October. Self-certifying ID based trustworthy networking system for IoT smart service domain. In: 2017 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, pp. 1299–1301.

Kshetri, N., 2017. Can blockchain strengthen the internet of things? IT professional 19 (4), 68–72.

Lazarenko, Aleksandr, Avdoshin, Sergey, 2018. Financial risks of the blockchain industry: a survey of cyberattacks. In: Proceedings of the Future Technologies Conference. Springer, Cham.

Lee, B., Lee, J.H., 2017. Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. J. Supercomput. 73 (3), 1152–1167.

Lee, B., Malik, S., Wi, S., Lee, J.H., 2016, July. Firmware verification of embedded devices based on a blockchain. In: International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. Springer, Cham, pp. 52–61.

Lin, I.C., Liao, T.C., 2017. A survey of blockchain security issues and challenges. IJ Network Security 19 (5), 653–659.

Lin, C., He, D., Huang, X., Choo, K.K.R., Vasilakos, A.V., 2018. BSeIn: a blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. J. Netw. Comput. Appl. 116, 42–52.

Makhdoom, I., Abolhasan, M., Abbas, H., Ni, W., 1 January 2019. Blockchain's adoption in IoT: the challenges, and a way forward. J. Netw. Comput. Appl. 125, 251–279.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., Qijun, C., 2017, October. A review on consensus algorithm of blockchain. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp. 2567–2572.

Pavithra, D., Balakrishnan, R., 2015, April. IoT based monitoring and control system for home automation. In: 2015 Global Conference on Communication Technologies (GCCT). IEEE, pp. 169–173.

Peterson, K., Deeduvanu, R., Kanjamala, P., Boles, K., 2016. A blockchain-based approach to health information exchange networks. In Proc. NIST Workshop Blockchain Healthcare 1, 1–10.

Roman, R., Zhou, J., Lopez, J., 2013. On the features and challenges of security and privacy in distributed internet of things. Comput. Network. 57 (10), 2266–2279.

Roy, D.G., De, D., Alam, M.M., Chattopadhyay, S., 2016, March. Multi-cloud scenario based QoS enhancing virtual resource brokering. In: 2016 3rd International Conference on Recent Advances in Information Technology (RAIT). IEEE, pp. 576–581.

Roy, D.G., Mahato, B., De, D., Buyya, R., 2018. Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT)—MQTT-SN protocols. Future Gener. Comput. Syst. 89, 300–316.

Tuli, S., Mahmud, R., Tuli, S., Buyya, R., August 2019. FogBus: a blockchain-based lightweight framework for edge and fog computing. J. Syst. Softw. 154, 22–36.

Veres, A., Boda, M., 2000, March. The chaotic nature of TCP congestion control. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), vol. 3. IEEE, pp. 1715–1723.

Walker, J., Pan, E., Johnston, D., Adler-Milstein, J., Bates, D.W., Middleton, B., 2005. The Value of Health Care Information Exchange and Interoperability: there is a business case to be made for spending money on a fully standardized nationwide system. Health Aff. 24 (Suppl. 1), W5–W10.

Yang, C., Chen, X., andXiang, Y., 2018. Blockchain-based publicly verifiable data deletion scheme for cloud storage. J. Netw. Comput. Appl. 103, 185–193.

Zhang, Y., Wen, J., 2017. The IoT electric business model: using blockchain technology for the internet of things. Peer-to-Peer Networking and Applications 10 (4), 983–994.
Zhang, Y., Deng, R.H., Han, G., Zheng, D., 2018. Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things. J. Netw. Comput. Appl. 123, 89–100.
Zyskind, G., Nathan, O., 2015, May. Decentralizing privacy: using blockchain to protect personal data. In: 2015 IEEE Security and Privacy Workshops. IEEE, pp. 180–184.

Deepsubhra Guha Roy is currently pursuing his Ph.D in the field of Internet of Things. He is the founder developer of Centre of Mobile Cloud Computing. His research area is QoS improvement in mobile cloud computing towards IoT. His email id: roysubhraguha@gmail.com

Puja Das is currently pursuing her Ph.D in the field of Internet of Things. Her research area is QoS provisioning in Intelligence and Secure healthcare system for IoT. Her email id: puj adas.wbut@gmail.com

Dr. Debashis De (M′13-SM′15) is a Professor of the Department of Computer Science and Engineering of Maulana Abul Kalam Azad University of Technology, India and Adjunct Research Fellow of University of Western Australia, Australia. His research area includes energy and latency optimization in mobile cloud computing. He has received Young Scientist award both in 2005 at New Delhi and in 2011 at Istanbul from International Union of Radio Science, H. Q., Belgium. His email id: dr.debashis.de@ieee.org.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He has authored over 625 publications and seven text books including "Mastering Cloud Computing" published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. Dr. Buyya is recognized as a "Web of Science Highly Cited Researcher" in 2016 and 2017 by Thomson Reuters, a Fellow of IEEE, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud computing. For further information on Dr.Buyya, please visit his cyberhome: www.buyya.com.