A Comprehensive Study of Attacks on the IoT and its Counter Measures Using Blockchain

Pardeep Kaur and Shalli Rani

1 Introduction

In 1999, the term “Internet-of-things” (IoT) was first used by Kevin Ashton. Such devices are able to assemble information more accurately and efficiently than a person, and this assembled data has possibly changed the human life-style [1]. It combines the ‘Internet’ with devices such as sensors, mobiles, actuators, localization systems or Radio Frequency Identification (RFID) tags called “things” having a unique address and provides a network of different devices. By 2025, the US National Intelligence Council expects that chips will resides in everything, including paper documents, furniture, and food packaging [2]. The integral elements of the IoT are Wireless Sensor Networks (WSN) and Machine-to-Machine (M2M) systems. With the arrival of smart homes, cities, and smart vehicles, the Internet of things (IoT) has developed as a territory of unbelievable effect, capability, and development, and according to the prediction of Cisco Inc., by 2020 there will be 50 billion interconnected devices [3]. Most of these devices are easily hackable because of limited computation, repository, and network capacity; therefore, they are unsafe from endpoint devices such as computers and smartphones. To protect all these devices from the different kinds of attacks, blockchain technology has come into use. Blockchain technology was used by a group of researchers to time stamp digital documents in 1991 [4]. In 2008, this technology was reinvented by Satoshi Nakamoto for bitcoin (cryptocurrency) [5]. Blockchain has been imple-

P. Kaur · S. Rani (✉)
Chitkara University Institute of Engineering and Technology, Chitkara University,
Rajpura, Punjab, India
e-mail: pardeep.kaur@chitkara.edu.in; shalli.rani@chitkara.edu.in

S. Rani et al. (eds.), *Integration of WSN and IoT for Smart Cities*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-38516-3_2

mented over a wide range of industries, including real estate, finance, healthcare, the government sector [6], and the IoT. This explosion occurred because applications that could only run through a central system can now run without any trusted third party through peer-to-peer connectivity with the same functionality and the same amount of certainty. Due to the fast development of smart devices, such as smart television, cars, and smart phones, with high-speed networking capacity, the IoT has achieved vast popularity and acknowledgment. It depicts a network where "things" are embedded devices, and they have sensors/chips. These sensors/chips are able to communicate through a public or private network [1], and these devices can remotely control and provide ideal functionality. The information sharing from one node to another node takes place over a network that deploys the standard protocols of communications. These smart connected devices range from small devices to broad machines, and each of these devices contains sensors or chips. For instance, according to the American College of Cardiology, Apple Watch's effectiveness at detecting heart conditions such as AFib is promising; this test is the largest ever conducted. Apple watch users who received notifications from the watch about an irregular heart beat were given an electrocardiogram device to wear. Using the ECG, the scientists were able to confirm that a third of those who received a warning from the watch actually had AFib. About 84% of notifications from the watch were confirmed to be AFib episodes since the condition can be intermittent [7]. Similarly, home devices, such as television, lights, refrigerators, and washing machines, can also be controlled remotely through the IoT. Similarly, smart cars contain many features such as an auto gear system and auto parking system. There are manifold other applications, such as industrial controlling and monitoring, location determination at hazard sites, smart badges and tag, monitoring tire pressure, monitoring soil for moisture, pesticide, herbicide, and pH levels, as well as peripheral devices, such as joystick, wireless mouse, and games. Home automation involves heating, air conditioners, security, lighting, ventilation, automatic curtains, windows, doors, locks, etc.

The IoT is not only available for personal use; it also serves in national needs. Different smart devices perform various activities, such as traceability of objects through chips so that the current location of an object can be known, interconnectivity in automobiles, placing the chips on birds helps to uncover areas for maps, monitoring surgery in hospitals, identifying climate conditions, and distinguishing proof of creatures using biochips. The information gathered through these devices is used in real time to enhance the performance of the entire framework. Distributed computing gives the virtual framework to utility processing, which incorporates examine devices, repository devices, visualization platforms, analytics tools, and customer shipments [8]. On demand customers can access the data from anywhere at any time. Smart connectivity with existing networks using the network's resources is an essential element of the IoT.

2 Security Requirement for the IoT

2.1 Privacy, Integrity, and Confidentiality of Data

In IoT networks, data travels over various devices; therefore, to ensure the confidentiality of data, cryptography techniques such as the encryption decryption mechanism of data are required. For pernicious goals, attackers may attack the data stored in IoT devices by modifying that data to violate its integrity.

2.2 Authorization, Verification, and Accounting

Authorization mechanisms ensure that information access is only provided to authorized users. Authentication is essential in IoT devices to secure the communication between two parties. The devices in the network must be verified so that privileged access can be given to specific devices. Because IoT architecture contains heterogeneous devices, the verification mechanisms must be diverse. The combination of verification and authorization provides a proper secure and trustworthy environment for interaction. Accounting represents the source utilization, and a solid mechanism for protecting the network framework is given by evaluating and maintaining records.

2.3 Services Availability

Denial-of-service attacks might block the availability of services given by IoT devices. To degrade the services provided by IoT devices to their users, at various levels attackers use different attack approaches, such as Spoofing attacks, Sinkhole attacks, Blackhole attack, Buffer reservation attack, jamming adversaries or replay attacks.

2.4 Energy Efficient

With low power consumption and short storage memory, IoT devices provide services to their users at very low cost.

By producing unnecessary artificial service requests, attackers cause network overflow to exhaust the IoT devices, resulting in an increase in power utilization [3].

3 Failure of Single Point

A regular development of heterogeneous systems on the IoT-based framework may disclose countless single-points of failure. It is necessary to develop a protected environment for a large network of IoT devices.

4 Attack at Access Level

Attackers can access IoT devices in two ways:

Passive attack: In this type, the attacker can only read the data that is sent from sender node to receiver node.

Active attack: In this type, the attacker can read as well as modify the data over the communication medium.

5 Categories of Security Issues with Their Solutions

The IoT has a wide range of heterogeneous devices from small chips to large-end users. Therefore, security and privacy issues take place at different levels [1]. Detecting attacks is essential, as it is the primary step toward constructing a harmless and trustworthy wireless network. In recent years, the IoT has been a focal point of research, and since everything is connected to the Internet, security and protection of broadcasted and stored data are the main concerns in IoT applications. As security threats, various vulnerabilities exist at different levels.

5.1 Security Issue at Low Level

Jamming Attack (Denial-of-Service)

The jamming style attacks occur due to a shared medium in wireless networks. In this type of attack, a jammer repeatedly ejects radio frequency signals to block the traffic on the network [9, 10]. A jammer remains silent when there is no action taking place over the channel but when it identifies activity on the channel it starts interference over the network and stops following MAC protocols. Before transmitting it does not wait for channels to become ideal. Sometimes, instead of injecting bits, it alternates the network between sleeping and jamming. As a result, the attacks target the reception of messages. There are four categories of jamming attack models: (a) Constant (b) Deceptive (c) Random, and (d) Reactive jammer [11].

For wireless sensor networks, a jammer is an element that is intentionally attempting to hamper communication to corrupt packages over transmission. The presence of a jamming attack can be determined by different techniques such as signal strength (ambient energy) and carrier sensing time. Both of these techniques are able to detect a constant jammer and deceptive jammer but are not able to detect random and reactive jammers effectively. Xu et al. [11] proposed a technique to identify jamming attacks through the procedure of successful Packet Delivery Ratio (PDR). The PDR is either calculated by the sending node or by the receiving node in communication. The PDR on the sender side is evaluated by keeping track of acknowledgements received from the receiver. On the other hand, the PDR is evaluated by the ratio of packages that pass the Cyclic Redundancy Check (CRC) to the number of packages that reach the receiver side. There are two detection algorithms: (a) Consistency check signal strength can be achieved by evaluating PDR between each node with its neighboring nodes and then checking whether this PDR value is consistent with signal strength. (b) Consistency check location information uses the PDR value to indicate the link quality and provide location information of the WSN. The jamming status of a node can be concluded by checking whether the node's calculated PDR value is consistent with the given location of its neighboring nodes. A High PDR value means neighboring nodes are close to that particular node, and if all neighboring nodes have low PDR values, it means the node is affected by a jamming attack.

Another anti-jamming technique is dependent upon the integration of error-correction codes with cryptographically strong interleaves (i.e., attackers are not able to estimate the interleaving function). Existing anti-jamming systems depend upon a broad utilization of spread-spectrum techniques utilized at the bit level to protect data packages. This kind of technique independently assures bits from jammers and are satisfactory for sound communication in which the jammer blocks the communication channel so that packages would not reach their destination in the appropriate time. The interconnecting nodes use a high-gain spread sequence to reduce the energy of jammers. In a "non-error-correction" encoded data package, a single bit error causes deprivation to the loss of the full package generating a CRC error [12].

Insecure Initialization

For appropriate functionality of an integrated system without breach security and privacy of network services, a secure method is required to initialize and configure the IoT framework at the physical level. Due to the broadcast behavior in wireless networks at the medium access control (MAC) layer or network layer, communication over the channel is generally vulnerable to intrusion by unauthorized receivers. That is why security becomes an essential issue as well as a challenge. Cryptographic technologies are the center attraction of security issues; however, they also have some limitations including computational complexity because it is tough to perform resource constrained wireless networks and proper arrangement of secret keys [13].

The existing physical layer security mechanism was introduced by Pecorella et al. [14], and these mechanisms are categorized into four parts on the basis of physical characteristics such as secrecy transmission capability, channel fingerprint, spectrum spreading of signal power, and cooperative. The physical layer technique is used to upgrade secrecy for WSN in which an unauthorized user is unfamiliar with the communication channel or the communication channel of an authorized user is less noisy than for an unauthorized user. The secrecy transmission rate is the rate at which secrecy data is transferred from sender to receiver, and both of them are assured that there is no eavesdropper within the specific area. The maximum secrecy rate is called its secrecy capacity.

Using the physical layer security technique makes it more challenging for attackers to analyze the broadcasted data [13]. Signal processing methods play an essential role in developing physical-layer secrecy in multiantenna wireless systems. In the data transmission phase, artificial noise and secrecy precoding transmission mechanisms are appropriate ways to expand the signal quality difference at the intruder [15].

Low Level Sybil Attack

Sybil attacks are popular in peer-to-peer networks where the communicating medium is broadcasting and are generated because of malicious Sybil nodes, which operate a huge number of forged identities. By using random medium access control (MAC) protocols a Sybil node might throw a channel demand message of high rate to an access point to mimic a huge number of clients so that the IoT application's performance can be degraded. As a result, appropriate nodes in the same network are refused access once the Sybil node has dominated an access point's channel schedule [16]. One approach to execute a Sybil attack is to have the Sybil node directly interact with appropriate nodes, and when this appropriate node broadcasts a message to the Sybil node, the existing malicious node reads the information. A second approach is that no appropriate node directly interacts with Sybil nodes. Messages broadcasted to a Sybil node are transmitted through one of these malicious nodes, and that intermediator malicious node pretends to pass on the information to a Sybil node [17].

In Sybil attacks, a malicious node illegally claims a huge number of forged identities and exhausts resources over the network. In a Rich-scattering environment, to detect Sybil attacks, Hong et al. [15] introduced a channel-based authentication method that uses the uniqueness of channel responses in indoor and urban environments. The Sybil indicator includes a test measurement that is selected on the basis of attack procedure, number of claimed identities, and synchronous access points. The test examines various parameters such as number of channel estimates, total users, Sybil users, access point, bandwidth, and signal power. Demirbas and Song [18] find Received Signal Strength Indicator (RSSI) to be a reliable and time-varying solution to Sybil attacks because it does not burden WSN with keys pair. When the destination node receives a message from the sending node, the receiver joins the sender identity with RSSI of the message. When other messages using the same RSSI associate with various sender identities, the receiver will be able to detect the Sybil attack.

Spoofing Attacks

Spoofing attacks such as IP address spoofing attack and address routing protocol are genuine risks and occur when a malicious party mimics another user in a network to eject attacks so that it will be able to read stored or transmitted data from the sender to the intended receiver. It is necessary to identify the existence of spoofing attacks and remove them from the network. Using cryptographic verification is the conventional way to deal with such attacks [19].

Full-scale cryptography authentication is not enough for identity verification; it needs a proper key-management system, further infrastructure and computational load. K-mean cluster analysis is the mechanism for identifying spoofing attacks along with locating the address of the performers of the attacks. The position of the attacker can be identified by using localization algorithms, which are based on area or point. The functionality of the K-means spoofing identifier is calculated in terms of detection rates and receiver operating characteristic curves. Normally, the gap among the original node and spoofing node can be predicted with a median error of 10 feet [19]. Another approach, proposed by Xiao et al. [20], is a rich scattering environment, the physical layer technique for upgrading authentication. This technique uses hypothesis test and channel frequency response measurements to distinguish between unauthorized and authorized users. Another approach proposed by Li et al. [21] is the idea of a forge-resistant link related to transmitted packages and when forge-resistant consistency verifies, which allows other network items to identify abnormal functionality.

Sleep Deprivation Attack

Sleep deprivation attack is a destructive attack and especially appears in a link layer, where a pernicious node insists for appropriate nodes to waste their power by restricting the sensor nodes from going into low power sleep mode. The main aim of this attack is to enlarge the power consumption of the target node. In the 6LoWPAN environment, when a huge number of activities are performed at a time it causes reduction of battery life.

Rivest et al. [22] introduced a system of cluster-based layered technique to design a lightweight Insomnia Mitigating Intrusion Detection System (IMIDS) that can mitigate this kind of attack without utilizing MAC protocols, such as G-MAC, S-MAC, B-MAC, and T-MAC. The goal of this system is to increase the lifetime of the WSN, and this system framework can productively relieve sleep deprivation assault in WSN. Computer Simulation conclusions in MATLAB display the viability of the introduced model in finding sleep deprivation attacks. Bhattasali and

Chaki [23] solve the problem of WSN by utilizing multiagent and semantic techniques. Network security consists of two categories: prevention-based techniques and detection-based techniques. At the point when an interruption happens, prevention-based techniques are regularly the main line of defense against assaults. After the failure of prevention-based techniques, then the detection-based technique is used to identify and exclude the attacker. The detection-based technique is further categorized: misuse detection and anomaly detection. After attack, misuse detection is used to define a changed pattern from the original pattern. Anomaly detection uses a set of typical profiles and distinguishes uncommon deviations from the ordinary conduct as anomalies.

5.2 Security Issues at the Intermediate Level

At the IoT network layer and transport layer, intermediate-level security issues take place that are primarily concerned with session management, routing, and interaction between nodes.

Fragmentation Caused Replay or Duplication Attacks

The fragmentation procedure is a breakdown of an IP datagram into small packages so that they can be easily transferred over different networks from a sender and assembled on the receiver side. A datagram receipt consists of the sender's address, receiver's address, size of datagram, and tag of datagram. Several styles exist where attackers can use fragmentation to percolate, causing denial of service or replay attack to networks. Only a sender and receiver is required for fragmenting and reassembling of packages. Thus, in-between routers and switches should not include the fragmentation process. If the IP package is modified in between sender and receiver, a security hole exists. An attacker can block the successful transmission of packages at the destination node. Additionally, an attacker can manipulate the datagram receipt based on which sensor nodes will suffer from re-booting, performance can be stopped or shutdown, because of re-sequencing of packages re-assembling buffer overflowed at receiver side, exhausting processing sources, etc. Some IP package fragmentation attacks include ping of death, jolt, fragrouter, new teardrop, and tiny fragmentation. [24].

Kim [24] introduced a replay attack protection technique. This technique adds a timestamp for unidirectional fragmented packages and a nonce for bidirectional fields to the datagram receipt. Without any serial number, timestamp and nonce provide security against replay attacks. The main aim of the timestamp filed is to fit a datagram within a single frame by dividing it into small packages. All packages, except the last one, are a multiple of eight bytes. The nonce field consists of an arbitrary number of at least 6 bytes that is chosen by the sender of the requesting message. The main aim of nonce is to ensure that packaging is a fresh reaction to a request sent previously by the node.

Hummen et al. [25] proposed two other approaches: content chaining scheme and split buffer. In the content chaining approach, a node is able to cryptographically check that accepted fragments at the receiver end belong to the same package on a pre-fragment basis. In the split buffer approach, the appropriate sender directly competes with an attacker to assemble buffer sources and splits the assembled buffer into fragment-size buffer slots. The buffer slot will be filled when a complete package has been delivered or overload condition is reached. If overload condition is reached, then the node has the capability to take the decision of which package to discard, and this decision depends upon per-package scores. Packages with lower score will be discarded, and those with highest score will be accepted.

Buffer Reservation Attack

The buffer reservation attack reserves the memory space of the receiving node for a time period of package reassembly timeout. If all fragments are delivered successfully, the attacker benefits from the recipient of a fragmented package not being able to determine that most of the space has been misused by the attacker. The receiving node in the network is the buffer space used to store the fragmented packages for assembly, and the attacker can misuse this space by sending inadequate packages to the target node. The space is occupied by these inadequate packages, causing the overflow condition. Now the targeted node has to discard some packages.

The attacker can block this space at very low cost. Hummen et al. [25] proposed a scheme that increases this cost for the attacker so that the attacker has to send inadequate fragmented packages continuously in short bursts to keep the network busy and reserve space at the target node and the appropriate node would not be able to reserve space for package assembly at the receiver end, causing the buffer reservation attack to suffer from a flooding attack. Thus, the attacker gains no advantage by sending unfragmented packages to the target node.

Another approach sets the reassemble timeout of fragmented packages to 60 seconds to handle fragment loss during communication. The goal of this timeout is to decrease the space occupied by incomplete reassemble fragments. When this time runs out, the receiving node will drop all the incomplete fragments so that memory can be free for new fragmented packages.

Insecure Neighbor Discovery

In IoT development architecture, each device has to be especially distinguished over the network. It is necessary to secure the particular channel through which information is to transfer to a specific destination in the end-to-end message communication. The sensors in WSN have less power, are less expensive, of small size, and are capable of doing small computations. Thus, these sensors are penetrable to different attacks, such as man in middle of communication of two nodes, tracing of router, and duplicate address identification, and these are neighbor discovery attacks. If the neighbor discovery signals penetrate toward wormholes (discussed later), the appropriate nodes will get the wrong routing information about their neighbors. This might prompt the decision of a non-existence route [26].

To secure neighbor discovery packages directly, a security framework with modules is explained by Riaz et al. [27], and it can be achieved by using an IP security authentication header with a pair of symmetric keys. This pair of keys is only known to participation nodes of the network. However, it stills faces some security issues, and to overcome this problem it uses a public key signature for verification of neighbor discovery packages. When one node finds its neighbor node, before set up communication keys, first it will verify that discovered node by applying some verifications methods.

Sinkhole Attack and Blackhole Attack

A sinkhole is a kind of denial of service (DoS) attack engaged by an inner attacker to disturb the functionality of a wireless sensor network. Basically, a sinkhole attack happens when a bargained node executes two pernicious acts. The first act is, by advertising a favorable path, the sinkhole attracts valid traffic from its surrounding nodes by modifying the rank of messages in the destination information object (DIO) message. The rank field describes the positivity of the node to its neighbors. The bargained node creates false routing rumors that it has power and spreads it to neighboring nodes. When the bargained node broadcasts its low rank then all traffic moves toward this node, causing its neighbor to choose it as a parent, and it becomes the cluster header of each round. The second act is, when all nodes start transferring packages to this malicious cluster header, it starts dropping only selected packages. A sinkhole attack slows down the process of message sending from the sender node to the receiver node to reduce the end-to-end delivery functionality. In a blackhole attack, the procedure is the same, except that the malicious cluster header drops all the packages.

Weekly and Pister [28] found two methods to overcome the sinkhole problem. The first is rank verification and the second is parent failover. These methods allow complete packages to be sent to their destination.

The rank verification method necessitates that bargained nodes lower their rank by 1 and all the edges in the graph should be of equal weight. Moreover, together these techniques are more effective than if they are applied individually. The method is based on a one-way hash function such as SHA1 [29] and a hash chain. This technique can help to acquire high quality end-to-end performance by upgrading the density of routers in the network. The root node begins with any random number and calculates its hash. When this hash is broadcasted, the DIO rank is accordingly increased or replaced. In the network, before forwarding, the appropriate nodes execute further hashing, but on the other hand the bargained node starts communicating with the hash value. After some time, to assure that the routing tree has converged, for verification by individual nodes, the root node executes a broadcast of the random value selected at first to all the nodes in the network. If a discrepancy occurs at a node, it means an inauthentic parent rank value.

In the end-to-end acknowledgement method or parent failover method, when the root node first transmits the DIO message, it inserts an unheard noise set (UNS) attribute to this message. This UNS is approved by the root node so that no one can change the transmitted data. The UNS is used to identify the bargained node by a sinkhole in the network. After an interval of 10 seconds, non-root nodes transfer data. If the root receives this data less than 30%, this node will be added by the root to the UNS. Carefully choose the threshold value. When a node receives the DIO message with itself involved in the UNS, the node blacklists its parent for consequent intercommunication, so that no node would ever select it as parent.

Firoz et al. [30] introduced two approaches for blackhole attack: local decision and global verification. Every node in the network notices the communication behavior of its neighbor by eavesdrop data packages transferred by its neighbors, and a node identifies a mistrustful node based on the data collection of the communication behavior of its neighbor nodes. After finding a mistrustful node, the checkout procedure starts, and, here, a validating node validates a mistrustful node. Verification is started by a query from the root node to find out if it received packages.

Wormhole Attacks and Rushing Attack

In networks, sensors use radio channels to transfer data/information from a sender to an intended destination. Malicious nodes intercept the packages and pass them through to some other location in the same network and retransfer them, and this penetrating process makes a wormhole in the sensor network [26]. If a fast transmission path lies in-between the two ends of a wormhole, the penetrated packages can propagate faster than the normal multi-jump route, causing a rushing attack [31].

The previous approaches require that the node should be prepared with special hardware, such as a synchronized clock or antenna. Wang and Bhargava [26] presented a method to protect a network from such wormhole attacks, MDS–VOW (Multi-Dimensional Scaling – Visualization Of Wormhole). In this method, special hardware does not need to be attached to sensors, and it selects and combines mechanisms from sociology, PC illustrations, and logical representation. By using multiple dimension scaling, MDS–VOW retraces the network to find the positions of fake identities.

RPL Routing Attack

Due to the increasing demand for wireless sensors with short memory and low power, IPv6 Routing protocol was deployed by IETF [32] ROLL Working Group to provide routing solutions. The IPv6 Routing Protocol for Low-power and Lossy

Networks (RPL), constructs and maintains directed acyclic graphs (DAG) routed to at least one base station. The base station compiles the information evaluated by other participating nodes in the network and controls these nodes. Thus, the destination node in the network is the base station. An attacker enters into the base station or attacks the nodes near the base station and can modify, intercept, forge, replay, and create messages so that they can interfere with the function of the whole network by divert routing toward a large path so that the node batteries are exhausted. If the attack impact is major then it will lead to reconstructing of the entire DAG and may exhaust the node's batteries. An attacker can achieve this by modifying the version number of the Destination Oriented Directed Acyclic Graph (DODAG) or by modifying the DODAG node's rank value.

The term rank value defines the node's level in the graph. If the rank is low, the node is close to the base station, or if its rank is high, the node is far from the base station. Sibling nodes of a particular node also have the same rank value, as a node will forward more messages to the closest one. If a node sets one node as a base station, a loop can occur in the network. If the network find such loops, and sometimes it is difficult to solve loops, then the whole graph might reconstruct, and this construction starts when the base station transfers DIO messages with an upgraded version number.

To block loop generation in the network, the RPL follows the rank rules that in the network a child node should always have greater rank than its parent. The DIO message has the version number as a component, which is correlated to the network. At each time, the version number is augmented monotonically by the network root, and to revalidate the integrity and enable worldwide occurrence, the root of DODAG chooses to form a new version of the DODAG. The version number is channeled unchanged down the DODAG as nodes join the new DODAG. For verification version numbers and ranks, the proposed security system is known as Version Number and Rank Authentication (VeRA) [33] and uses the hash function such as SHA [29] and MAC function such as digital signature RSA [33] and HMAC [34]. According to RPL protocol, the rank value of a child node is greater than the selected DODAG parent node to stop generation of loops, and the rank value of sibling should be equal to that node. RPL provides cryptography methods for securing control messages, but the network is still able to be attacked because sensor objects are not manipulation resistant. In RPL, through control messages, the child gets the parent's information messages, causing it to follow a poor quality route because it is not able to check the service provided by the parent node, and it may be possible that there is a malicious node.

Sybil Attacks at the Intermediate Layer

When attackers generate fake identities, IoT devices are penetrable to Sybil attack so that system efficiency will be compromised. Due to Sybil attack, the IoT system may generate wrong reports, which is like cheating a user, the privacy of the system can be lost, and the effectiveness of the network reduced. Users might receive spam messages or mails. These Sybil identities attract other users by sending spam advertisements and mails to other users to steal other client's private data. Sybil identities disperse malware and phishing sites. In a dispersed vehicular correspondence framework [35] and portable social frameworks [36], Sybil identities produce one-sided choices with "legible" accounts. Without a successful Sybil detection system, the aggregate outcomes can be effectively controlled by the attackers. Since most Sybil identities behave like ordinary clients, it is difficult to distinguish them [37]. Usually, the Sybil identities exist in sensing and social domains, such as Online voting systems [38] or mobile sensing systems [39].

Yu et al. [40] presented SybilGuard as a novel decentralized protocol against Sybil attackers by limiting the size and number of the Sybil group. This protocol depends upon the "social network", where a connection among two users defines a human-established trust relationship like a friend relationship. Sybil users can create many accounts in one network but there are few true relationships. The attack edges connect the honest region (which contains all honest users) to the Sybil region (which contains all malicious users). For 99.8% of the honest users, SybilGuard assures that the size and count of Sybil groups are bounded, and then only these honest nodes will be accepted in the same network by 99.8% of all other honest nodes. These attack edges effect both distributed and peer-to-peer systems including the IoT. The protocol assures that the count of Sybil identities are independent from attack edges, but it depends upon trust edges among Sybil nodes and honest nodes.

Du et al. [41] analyzed various Sybil attacks with their solutions in the IoT. Sybil attacks are divide into three categories based on Sybil attacker's capabilities: Sybil Attack-1, Sybil Attack-2, and Sybil Attack-3. The Sybil attack's solutions are: (a) social graph-based Sybil detection (SGSD), (b) behavior classification-based Sybil detection (BCSD), and (c) mobile Sybil detection (MSD). In Sybil Attack-1, Sybil identities strongly connect with Sybil identities and there are limited relations between Sybil identities and honest identities. In Sybil Attack-2, attackers exist in a social domain. Sybil Attack-2 builds social connections with Sybil identities and true identities. The aim of Sybil Attack-2 is to steal the user's personal information to violate the user's privacy by modifying their personal details. In Sybil Attack-3, Sybil attackers exist in a mobile domain. Basically, Sybil Attack-3 has the same aim as Sybil Attack-2. Due to the dynamic nature of a mobile network, Sybil identities cannot connect with others for a long time period. The aim of SGSD is to label nodes "Sybil" or "honest". Consequently, there are basically two types of SGSD: (a) social network-based Sybil detection (SNSD) [40] and (b) social community-based detection (SCSD) [42]. Analyses of the Orbit Showtime Network (OSN) in [43] compares true and Sybil client's behavior by their clicking habit and browser history and distinguishes them [43]. According to the client's behavior categorization and learning, the BCSD [44] can identify Sybil Attack-2. The aim of MSD [45] is to either identify Sybil Attack-3 or to limit Sybil attacker's behaviors by three ways: (a) Friend Relationship-Based Sybil Detection (FRSD) (b) Cryptography-Based Mobile Sybil Detection, and (c) Feature-Based Mobile Sybil Detection.

Verification and Secure Communication

For secure communication in various IoT devices, verification and access control are essential activities. Due to weak physical security, mobility and dynamic network topology of low power and small storage devices in IoT networks are penetrable to several attacks [46]. Any ambiguity in security at the network layer might expose the network to a large number of dangers [47, 48]. Moreover, because of the constrained resources nature of the smart object, the overhead of Datagram Transport Level Security (DTLS) needs to be reduced and standard protocols established, but direct solutions cannot be used in 6LoWPAN/CoAP networks [49].

To achieve security in wireless networks, the IoT devices and clients are authenticated through key management systems [41]. With the origination of IPv6, a unique ID is assigned to each and every device in the entire world. It is easy to detect what data is sent or received by which device at what time, etc. To ensure data exchanges, IPv6 protocol stacks use IP security [50]. Today, in the evolution of the Internet, the IPv6 protocol and the LoWPAN adaption layer play an important role. Myriad sensing applications have come into existence on the basis of end-to-end communication and secure group communication among the internet administrator and sensing devices. These types of communications are only executable if proper security processes are adhered to in LowPAN [49]. For proper security in the network layer, Granjal et al. [50] proposed an authentication header [51] and encapsulating security payload (ESP) [52] on wireless sensor networks that provides security for IPv6 communications. Security headers with the cryptographic algorithms are particularly used with the IP security architecture while providing fundamental security guarantees independently of the applications running on sensor nodes. For the network layer, security schemes are mapped to adapt to the requirements of various wireless sensor network applications.

Transport Level End-to-End Security

For end-to-end communications between sensor objects and other internet objects, a wireless sensor network application requires these devices to be interconnected with internet hosts. By using a pre-shared key distributed to participating nodes in advance and datagram transport layer security (DTLS) (advanced version of transport layer security (TLS)) [53], devices that depend upon constrained application protocols (CoAP) [49] can protect their communications [54]. At transport-layer, TLS provides end-to-end security and plays an important role. With the aim of minimum communication errors, the constraints such as microprocessor, energy, and memory evaluate how much energy is required for low-energy wireless communications and provide small packages with low communication speeds. The aim of transport level end-to-end security is to send data from a sender to a desired receiver; therefore, it requires authentication mechanisms so that data can be transmitted over a network in a secure manner without violating privacy [55].

There are three key security management and verification approaches for end-to-end communication with other devices: (a) In a Pre-shared key approach, devices save preconfigured keys. (b) In a Raw Public key approach, identification devices have at least one public key. (c) In certification mode, certification authority devices get public keys [55].

For end-to-end secure communication among IoT devices, Kothmayr et al. [56] proposed a scheme using two-way authentication based on RSA [57] that used a public key cryptography algorithm dependent upon existing Internet standard protocols such as Datagram Transport Layer Security (DTLS) protocol. The verification can be executed either by fully authenticated DTLS handshakes pre-shared keys or by a trusted platform module (TPM) [58] using RSA. The RSA certificates are transmitted in X.509 format with the help of RSA. The proposed architecture is an appropriate solution for the IoT because it provides confidentiality, integrity, verification with low energy, end-to-end potential, and memory overhead.

When Hyper Text Transfer Protocol (HTTP) clients approach the Constrained Application Protocol (CoAP) server at the back end, the proxy is required to translate packages. At the application layer, a mapping is required between HTTP and COAP so that no malicious code will be added. One solution is to use a Transport Layer Security pre-shared key (TLS-PSK) [59] for transport between routers, in which DTLS packages are encoded into TLS packages and TLS packages are encoded into DTLS packages. Each stream of data has its own TCP connection and protected TCP header.

Another approach is integrated transport layer security (ITLS) [60]. A package sent by a sender is encrypted with a pair of keys. The proxy uses the primary key to decrypt a package and then forward that package to the intended destination. DTLS does not support multicast; therefore, two participating nodes using DTLS first discuss a session key for communication. The key is evaluated by a pre-shared key and a pair of nonce generated by the server and client.

The requirements to fulfill the multicast for CoAP are defined in [61]. Another framework, BlinkToSCoAP [63], implements lightweight versions of DTLS, CoAP, and 6LoWPAN protocols over a tiny operating system. BlinkToSCoAP messages interchange among two Zolertia Zl devices, permit evaluation of energy consumption, memory footprint, and package overhead and potential. The obtained outcome demonstrates that securing CoAP with DTLS in the IoT is absolutely achievable without bringing about much overhead.

Session Establishment and Resumption

At transport layer, a session can be hijacked with a forged message, which can cause replay attacks, denial-of-service (DOS), wiretapped secret-key attacks, man in the middle attacks, etc. [62, 63]. An attacking node plays the role of a victim node to continue the session among two nodes. By changing the serial number, communicating devices have to re-transmission the message [1].

Peretti et al. [62] suggested a mutual authentication based on cryptographic module and session key distribution for a secure session management that is robust to various attacks. In this scheme, first an arbitrary number is selected, then encoding is applied on that number to produce a session key; afterwards, that key is used for encoding of another arbitrary number. This encoded value will be used for verification. Without repetition of parameters, a new session key can be generated for each session. Another verification method is the Edge-Fog-Cloud network architecture; at the Edge of the network Fog user's smart cards/devices are mutually verified with the Fog servers at the Fog layer. This scheme is known as Octopus and needs a roamer user in the network, who holds one long-lived master secret key so that they can communicate with any of the Fog servers in the network with the proper verifiable method [63]. Without any extra overhead and re-enrolment, the fog users can mutually authenticate with this new Fog server. For each user the Fog server stores only one secret key and fog users have no concern with public-key architecture. In this method, the fog user has to perform a few hash functions and symmetric cryptography.

Cloud-Based Privacy Violation

Network-based arrangements inside the IoT depend on cloud organizations. This methodology permits the advantageous and dependable communication of networks [64]. Due to the source constraints of IoT devices, the entire data related to a user is stored in clouds. These clouds can be used at anytime from anywhere. Cloud services are cheap compared to building one's own storage mechanisms. The input, output, and computation function are nearly correlated to the IoT user's privacy, which should not be presented to malignant IoT clients and pernicious cloud servers [65]. IoT attacks can breach a user's privacy and location privacy that is stored in the cloud. A pernicious cloud service could manipulate confidential data that is transferred from a sender to receiver or may modify data stores in clouds.

To overcome the concerns such as a pernicious cloud service provider that manipulates confidential data or an attack on the cloud, Henze et al. [64] proposed D-CAM, which uses the hash chain and digital signatures to tackle the issues of distributed and secure design, authorization and management borders in a cloud-based IoT network. To control messages, the cloud with D-CAM provides highly available and scalable storage.

5.3 Security Issues at High Level

Constrained Application Protocol Security with Internet

The application layer is penetrable to various attacks. The CoAP is a synchronized web transfer protocol constructed by IETF using HTTP for use with constrained networks and constrained sources, such as low power and short memory. CoAP gives a request/response communication system among end user applications by running over User Datagram Protocol (UDP) for both synchronized and unsynchronized acknowledgement. In a client-server system, to provide resource-oriented communication, CoAP uses HTTP commands such as PUT, POST, GET, and DELETE. The main purpose of designing a UDP-based application layer protocol is to overcome TCP overhead and bandwidth needs. Each header of the two bits package describes the Quality of service rank, what type of message the package contains, and the state of the package. The CoAP messages use a particular format described in RFC-7252 [66] for secure communication. The multicast support in CoAP requires verification methods and proper key management [67].

CoAP does not involve any inbuilt security functions; it was designed for machine-to-machine and IoT devices interactions. To protect CoAP-based networks and to assure end-to-end protection among two end users located in different networks, Datagram Transport Layer Security (DTLS) is used. DTLS fulfills all the requirements of an IoT environment, such as key management, cryptographic algorithms, integrity of data, and verification of data. At the application layer, during transition these protocols do not allow access to data at the gateway. One solution to secure LLN from various attacks is to use tunnels such as Virtual Private Network (VPN) [67]. At the network layer, VPN is used to interconnect two independent networks through a tunnel to hide the actual receiver and messages. The tunnel ends at the gateway such as the 6LoWPAN Border Router (6LBR) in another network. Support from the operating system is not important for utilizing the DTLS-DTLS tunnel method; however, it needs changes in the network stack of the back end system. Another approach proposed in [68] provides a protection system promoting internet-integrated wireless sensing applications and LoWPANs.

Middleware Security

In the IoT worldview, owing to the enormous number of technologies that take part in the exchange of data, some type of middleware layer is employed. The Naming Addressing Profile server (NAPS) connects various platforms as a middleware in the IoT environment. At the back end NAPS provides the services of collecting sensor's data, filtering data on the basis of content, and matching [69].

These devices are deployed across various platforms and middleware is used in the IoT for communication between different devices, networking, verification, to handle all the data transfer over a platform, for service support, protection, security, encryption, etc. Conzon et al. [70] proposed VIRTUS as a middleware solution to implement authentication and encryption for distributed application running in the IoT environment by using open standard protocols such as XMPP [71] and OSGi [72] for private networks only. By utilizing the standard security highlights given by the XMPP protocol, the middleware provides a solid and protected communication medium for distributed applications, ensured with both encryption and verification systems. Because of the scalability problem, for Java the OSGi scheme is a productive unit management framework and gives the primary standardized solution that enables applications to be made out of small, recyclable, and collective elements. In order to support the dynamic modules combination to simplify the configuration and the deployment, OSGi gives structuralism to runtime units' management and has been exploited in VIRTUS to deal with automatic supervising of dependencies and updates. The XMPP protocol is open for everyone, it is free, provides long-time security, being decentralized means anyone can individually manage the network by using its XAPP server, and being extensible means anyone can insert their features into the existing core features.

6 Blockchain

A blockchain is a chain made up of blocks, where every block consists of a public ledger or history of all the transactions made in a distributed manner and a copy of this ledger is shared among all the participating nodes in the network. Instead of a central server there is a peer-to-peer connectivity and each and every transaction is confirmed by consensus of a majority of participating nodes. Once information is entered into the public ledger it is almost impossible to delete it. Blockchain [73] is the technology behind the most famous digital currency—bitcoin [5], and it was introduced by Satoshi Nakamoto to solve the problem of double spending [74].

7 Blockchain Concept

Blockchain came into existence to solve the conventional problem of lack of trust in central authority. Blockchain technologies consist of cryptography algorithms, such as Security hash algorithm 256, mathematical puzzle, economic model, networks, and distributed consensus algorithms, for example, Proof-of-stake and Proof-of-work. Some essential elements of blockchain technology are: decentralized, transparent, open source, immutable, and anonymity [3].

7.1 Decentralized

Nodes are connected in a peer-to-peer manner in a blockchain network so that they can share their data directly with others. Distributary data can be stored and updated.

7.2 Anonymity

Trust issues between node to node in a blockchain network are solved by keeping transactions anonymous; only the node's public key address needs to be known.

7.3 Transparent

The recorded, stored, and updated data is transparent to every node in a blockchain network. It helps to understand when which transaction takes place, with whom it takes place, and who knows about the data.

7.4 Open Source

Anyone can join a blockchain network because it is open to each person.

7.5 Immutable

Once data is packed in a blockchain, it is impossible to tamper with it, unless at the same time an attacker or any other node can take control of over 51% of the network.

8 The Need of Blockchain in the IoT

The updated digital world still depends upon a trusted third party to send or receive their data. Most access solutions such as an email service provider that is a certification authority send notifications to us that an email has been delivered successfully or an email address is invalid. It could be any social network such as Facebook, Instagram or it could be a bank sending time-to-time notifications about a person's account balance or transactions and thus they have collected a user's personal data. The conventional truth of the digital universe is it depends on a trusted third party for the security and protection of our digital resources. In some companies such as Fitbit, the data they gather is public by default. Now the problem is hackers can hack these resources; they can modify these resources. That is why blockchain has come into use for security and privacy in every field, including the IoT. Blockchain has the potential to covert the third-party dependent world into independent by empowering a distributive consensus in which at any time in the future every single online transaction involving past and present digital assets can be verified without violating the protection of the involved digital resources and participating parties.

9 Types of Blockchain

9.1 Public Blockchain

In public blockchain each node can verify every transaction and by voting can take part in the consensus process. Examples of public blockchain are bitcoin and Ethereum [75].

9.2 Consortium Blockchain

Partially decentralized, open or private data. In this blockchain the node that has authority can be chosen in advance. An example of a consortium blockchain is hyper ledger [76].

9.3 Private Blockchain

Because accessing data for nodes will be restricted by authority, only selected nodes can participate in this blockchain.

10 Workings

The name ‘blockchain’ indicates that it is a chain of sequenced blocks in a chronological order. Every individual block consists of a unique hash, transactions that occur at the same time, the hash of the previous block that is mainly responsible for the chain, and the size of the block depends upon the transaction size. The first block of the chain is known as the genesis block as it does not contain any previous hash. Instead of a trusted third party, blockchain uses cryptography to execute a transaction between two willing parties. As shown in Fig. 1, each transaction is secured with a digital signature using a private key and shared public key [77]. The node that first receives the transaction confirms the digital signature with the help of the “public key” of the sending owner of the respective transaction.

In a network, Bob (a node) initializes a transaction request that he wants to send money to Alice (another node) and broadcasts this request over the network. The other nodes in the network verify this request by checking the history of whether the node that generated the request has enough balance to send money. If the transaction is valid then other nodes add this transaction into the public ledger. Then, nodes collect transactions that occur at the same time in a block and broadcast this block over the network. A problem exists here because nodes can also collect unconfirmed transactions and insert them into their block and can broadcast that created block over the network as a suggestion for the next block in the network. The order in which transactions are generated is different from the order in which they are received by different nodes in the network. How is it decided which block should be selected next for the chain? A number of blocks are generated by different nodes at the same time. To solve this problem Satoshi Nakamoto introduced a mathematical puzzle [5]. Each and every block is only accepted into the chain if it is able to solve the answer to this special mathematical puzzle.

This whole procedure is called “proof of work”. The miner has to show that he has enough computing resources and is able to solve the mathematical problem for

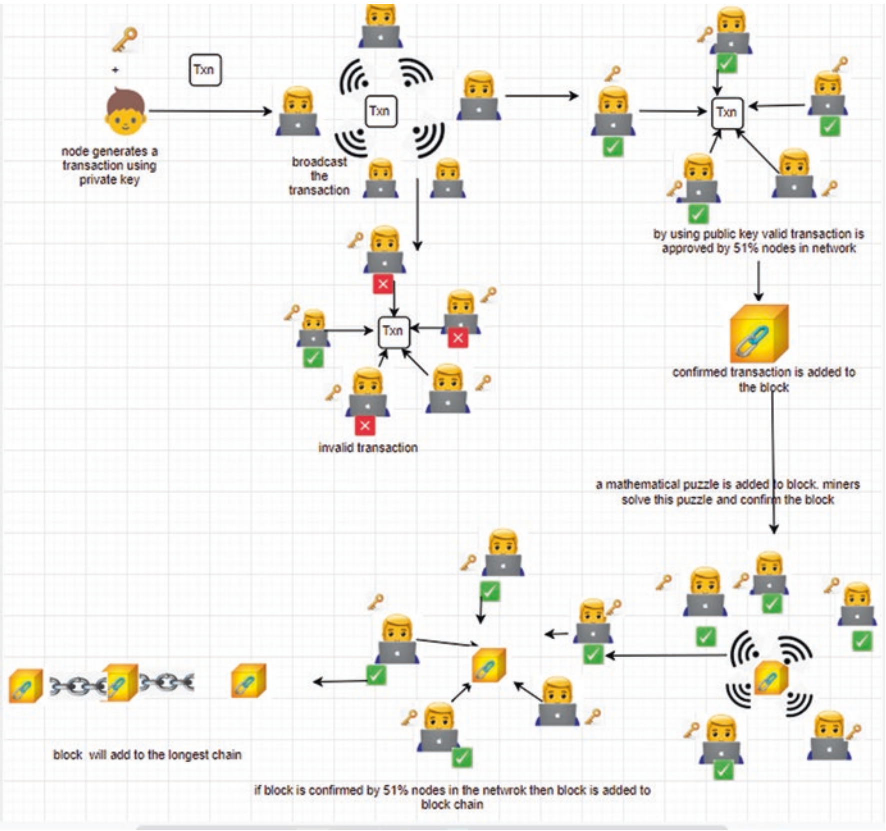
+
Txn
node generates a transaction using private key
Txn
broadcast the transaction
Txn
by using public key valid transaction is approved by 51% nodes in network
Txn
invalid transaction
confirmed transaction is added to the block
a mathematical puzzle is added to block. miners solve this puzzle and confirm the block
block will add to the longest chain
if block is confirmed by 51% nodes in the netwrok then block is added to block chain

Fig. 1 Workings of blockchain

generating a new block. Moreover, a node has to evaluate a "nonce" that is hashed with hashes of the last blocks and both transactions generate a hash with four leading zeros. The standard work needed is exponential in the count of zero bits required and by executing a single hash authentication procedure. The complexity of the mathematical puzzle is adjusted such that on average a node takes 10 minutes in a blockchain network to make a true answer and is able to add a new block in the chain. The first successful user to solve the puzzle will broadcast the block over the network. Sometimes, more than one block will be solved, leading to several branches in the chain. To order blocks in the chain according to agreement, nodes have to donate their computing resources to solve the puzzle and generate blocks. The nodes that donate their resources are called "miners" and they are awarded for their efforts (Fig. 2).

The longest chain is considered the only appropriate chain in the network. A node can generate a block not only by solving a mathematical puzzle but also there is a mathematical race with honest nodes in the network. Therefore, it is very difficult for an attacker to generate a fraudulent transaction. If a miner mines a block

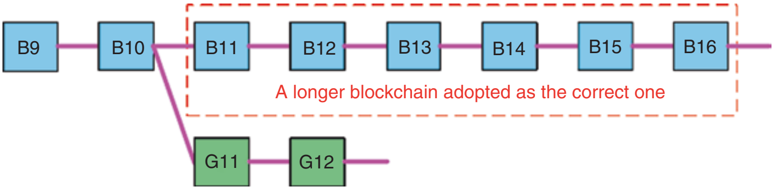

Fig. 2 Select longest chain [78]

with fraudulent transactions then he can lose his computing resources; therefore, every miner mines a block with valid transactions only. As blocks in the blockchain are linked cryptographically together, the whole procedure becomes more challenging.

11 Blockchain Solution for the IoT

As we have discussed, there are several types of attacks in the IoT environment. Blockchain provides a better, more robust and distributed peer-to-peer mechanism to prevent attacks, identify incompatibility and forks, and automatically resolve them without any help from a trusted central server. According to a client's requirement in the case of energy, the integration of the IoT with blockchains takes into consideration a shared market in which machines can buy and sell vitality energy automatically. For example, TransActive Grid [79] is exploring various avenues regarding the idea of a shared market for a sustainable power source in an area of Brooklyn, NY [80]. Solar panels record their overabundance yield on a blockchain, and sell it to neighboring nodes through smart contracts.

11.1 Identity of IoT Devices

IoT devices have relationships with persons, and these persons could be users, manufacturers, administrators or suppliers. Management of this relationship is a difficult task because ownership of IoT devices changes during their lifetime from manufacturer to suppliers, then supplier to retailer, retailer to consumer, thus affecting their identity procedures such as verification and authorization of data. When devices are delivered at a destination through shipping, at the destination the owner sends a signed message to a smart contract to inform everyone that the IoT device has successfully reached the destination. Now this signed transaction plays the role of a verifiable cryptographic receipt of the delivery party's claim that IoT devices were received. On the other end, the receiving party also posts that it is in possession of the IoT devices. When devices get resold, the ownership of devices will be changed and revoked. The whole procedure includes a number of stakeholders and checks. To keep track of IoT devices, each and every stakeholder in this network maintains their own database, and the input from other parties in the same blockchain updates this database. Other challenges are management of IoT device attributes, such as type, location features, make, manufacturer, deployment, serial number, Global Positioning System coordinates, and capabilities, and relationships, such as deployed by, dispatched by, utilized by, updated by device-to-service, device-to-device, and device-to-human [6]. Blockchain has the capability to effectively resolve these kinds of issues. Blockchain provides a faithful, honest, reliable, authentic environment for observation and trace ownership of goods and assets. When IoT devices register in a blockchain network, then it provides an identity, attributes set, and complex relations to connect, and these attributes and relationships can upload, store, and update distributary on the blockchain network. TrustChain [81] is capable of generating such transactions among unknowns without any trusted central control. TrustChain is a permission-less tamper-proof warehouse to store transaction details regarding ownership of IoT devices. TrustChain generates a temporally immutable chain that is parallel to the original chain for each owner to keep their records; thus, every client can create their own genesis block.

11.2 Address Space

Blockchain is more scalable than IPV6 for IoT devices because blockchain supplies 4.3 billion addresses, and this count is extremely greater than IPV6. The address space of blockchain is 160-bit. A blockchain address is 20 bytes or a 160-bit hash of the public key generated by the Elliptic Curve Digital Signature Algorithm (ECDSA) [82]. Blockchain is able to produce and allocate addresses offline for around 1.46×1048 IoT devices. The probability of address collision is approximately 1048, which is considered sufficiently secure to provide a GUID (Global Unique Identifier). GUID requires no registration or verification when assigning and allocating an address to an IoT device [6].

11.3 Storing Data

Constantly, every second, a huge amount of data is collected and analysis results in economic growth. Data can be public or private. On the basis of this collected data, various organizations predict the future and much more. Each block contains a number of transactions, and each transaction contains one hash value. Storing each hash value requires a large amount of space; therefore, blockchain provides a solution to huge data by storing data in a Merkle tree root form. It combines the hashes of two transactions to make a single hash unless and until one single hash of each block is generated.

11.4 Public Ledger

In every sector, including finance or social networking, a user's sensitive and personal data is stored at a central server and users have no control over that stored data and are not able to use them. They do not know who uses their data or where it is used. Moreover, several types of attacks on central authority can modify or destroy data. Instead of storing data at a central server, blockchain stores data in a chain of blocks and a copy of each blockchain is provided to each user in the network, meaning each user maintains their database individually. When any transaction occurs in the network database of each user, they will be updated automatically. An attack on blockchain would only reach 10 or 20 copies because it is nearly impossible to attack a billion copies of blockchain. When an attacker makes some changes in one block then he has to recalculate the PoW for that block and blocks after that block because each block uses the previous hash and with the change of even a single bit the hash changes. Additionally, in blockchain, only the longest chain is considered a valid chain in the network.

12 Conclusion

IoT devices are used throughout the entire world. Each place surrounding us is sensing enabled by wireless sensor networks, and these IoT devices need to communicate with each other in a synchronized manner. On one hand, there are many benefits of these devices in real life, but on the other hand, these devices are vulnerable to various attacks. Managing such devices properly is a big challenge. When the whole procedure depends on a single central authority, an attack on this single center or any technical challenge may cause this central point to fail. In order to prevent such situations, blockchain has come into existence with peer-to-peer connectivity. There is no central server, and each participating node directly communicates with other nodes. Anonymity, transparency, immutability, decentralization, etc., make blockchain perfect to use in the case of IoT devices.

References

1. Ashton, K.: That 'internet of things' thing. RFID J. **22**(7), 97–114 (2009)
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
3. Khan, M.A., Salah, K.: IoT security: review, blockchain solutions, and open challenges. Futur. Gener. Comput. Syst. **82**, 395–411 (2018)
4. Haber, S., Stornetta, W.S.: How to time-stamp a digital document. In: Conference on the Theory and Application of Cryptography, pp. 437–455. Springer, Berlin/Heidelberg (1990)
5. Nakamoto, S.: A peer to peer electronic cash system, Bitcoin Organization (2008) http://bitcoin.org/bitcoin

6. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. Ieee Access. **4**, 2292–2303 (2016)
7. Apple Watches are surprisingly good at detecting heart conditions says recent Stanford study. https://www.techspot.com/news/79227-apple-watches-surprisingly-good-detecting-heart-conditions-recent.html
8. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. Futur. Gener. Comput. Syst. **29**(7), 1645–1660 (2013)
9. Wood, A., Stankovic, J., Son, S.: JAM: a jammed-area mapping service for sensor networks. In: 24th IEEE Real-Time Systems Symposium, pp. 286–297 (2003)
10. Xu. W., Wood, T., Trappe, W., Zhang, Y.: Channel surfing and spatial retreats: defenses against wireless denial of service. In: Proceedings of the 2004 ACM Workshop on Wireless Security, pp. 80–89 (2004)
11. Xu, W., Trappe, W., Zhang, Y., Wood, T.: The feasibility of launching and detecting jamming attacks in wireless networks. In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 46–57. ACM (2005)
12. Noubir, G., Lin, G.: Low-power DoS attacks in data wireless LANs and countermeasures. ACM SIGMOBILE Mob. Comput. Commun. Rev. **7**(3), 29–30 (2003)
13. Chae, S.H., Choi, W., Lee, J.H., Quek, T.Q.: Enhanced secrecy in stochastic wireless networks: artificial noise with secrecy protected zone. IEEE Trans. Inf. Forensics Secur. **9**(10), 1617–1628 (2014)
14. Pecorella, T., Brilli, L., Mucchi, L.: The role of physical layer security in IoT: a novel perspective. Information. **7**(3), 49 (2016)
15. Hong, Y.W.P., Lan, P.C., Kuo, C.C.J.: Enhancing physical-layer secrecy in multiantenna wireless systems: an overview of signal processing approaches. IEEE Signal Process. Mag. **30**(5), 29–40 (2013)
16. Xiao, L., Greenstein, L.J., Mandayam, N.B., Trappe, W.: Channel-based detection of sybil attacks in wireless networks. IEEE Trans. Inf. Forensics Secur. **4**(3), 492–503 (2009)
17. Newsome, J., Shi, E., Song, D., Perrig, A.: The sybil attack in sensor networks: analysis & defenses. In: Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004, pp. 259–268. IEEE (2004)
18. Demirbas, M., Song, Y.: An RSSI-based scheme for sybil attack detection in wireless sensor networks. In: 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), pp. 5. IEEE (2006)
19. Chen, Y., Trappe, W., Martin, R.P.: Detecting and localizing wireless spoofing attacks. In: 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 193–202. IEEE (2007)
20. Xiao, L., Greenstein, L., Mandayam, N., Trappe, W.: Fingerprints in the ether: using the physical layer for wireless authentication. In: 2007 IEEE International Conference on Communications, pp. 4646–4651. IEEE (2007)
21. Li, Q., Trappe, W.: Light-weight detection of spoofing attacks in wireless networks. In: 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, pp. 845–851. IEEE (2006)
22. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM. **21**(2), 120–126 (1978). https://doi.org/10.1145/359340.359342
23. Bhattasali, T., Chaki, R.: A survey of recent intrusion detection systems for wireless sensor network. In: International Conference on Network Security and Applications, pp. 268–280. Springer, Berlin, Heidelberg (2011)
24. Kim, H.: Protection against package fragmentation attacks at 6lowpan adaptation layer. In: 2008 International Conference on Convergence and Hybrid Information Technology, pp. 796–801. IEEE (2008)
25. Hummen, R., Hiller, J., Wirtz, H., Henze, M., Shafagh, H., Wehrle, K.: 6LoWPAN fragmentation attacks and mitigation mechanisms. In: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 55–66. ACM (2013)

26. Wang, W., Bhargava, B.: Visualization of wormholes in sensor networks. In: Proceedings of the 3rd ACM Workshop on Wireless Security, pp. 51–60. ACM (2004)
27. Riaz, R., Kim, K.H., Ahmed, H.F.: Security analysis survey and framework design for ip connected lowpans. In: 2009 International Symposium on Autonomous Decentralized Systems, pp. 1–6. IEEE (2009)
28. Weekly, K., Pister, K.: Evaluating sinkhole defense techniques in RPL networks. In: 2012 20th IEEE International Conference on Network Protocols (ICNP), pp. 1–6. IEEE (2012)
29. Eastlake, D., Jones, P.E: RFC3174-US Secure Hash Algorithm 1(SHA1). (2001). https://tools.ietf.org/html/rfc3174
30. Ahmed, F., Ko, Y.B.: Mitigation of black hole attacks in routing protocol for low power and lossy networks. Secur. Commun. Netw. **9**(18), 5143–5154 (2016)
31. Hu, Y., Perrig, A., Johnson, D.: Rushing attacks and defense in wireless Ad Hoc network routing protocols. In: Proceedings of the ACM Workshop on Wireless Security (WiSe) (2003)
32. Handley, M., Bonaventure, O., de Louvain, U.C.: Internet Engineering Task Force (IETF) A. Ford Request for Comments: 6824 Cisco Category: Experimental C. Raiciu (2013)
33. Somani, U., Lakhani, K., Mundra, M.: Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. In: 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), pp. 211–216. IEEE (2010)
34. Krawczyk, H., Bellare, M., Canetti, R.: HMAC: keyed-hashing for message authentication (1997). https://tools.ietf.org/rfc/rfc2104.txt
35. Lin, X.: LSR: mitigating zero-day Sybil vulnerability in privacypreserving vehicular peer-to-peer networks. IEEE J. Sel. Areas Commun. **31**(9), 237–246 (2013)
36. Liang, X., Lin, X., Shen, X.: Enabling trustworthy service evaluation in service-oriented mobile social networks. IEEE Trans. Parallel Distrib. Syst. **25**(2), 310–320 (2014)
37. Zhang, K., Liang, X., Lu, R., Shen, X.: Sybil attacks and their defenses in the internet of things. IEEE Internet Things J. **1**(5), 372–383 (2014)
38. Tran, D., Min, B., Li, J., Subramanian, L.: Sybil-resilient online content voting. In: Proceedings of USENIX Network Systems Design and Implementation (NSDI), pp. 15–28 (2009)
39. Reddy, Y.: A game theory approach to detect malicious nodes in wireless sensor networks. In: Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 462–468 (2009)
40. Yu, H., Kaminsky, M., Gibbons, P., Flaxman, A.: SybilGuard: defending against Sybil attacks via social networks. IEEE ACM Trans. Netw. **16**(3), 576–589 (2008)
41. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: IEEE INFOCOM 2004, vol. 1. IEEE (2004)
42. Xue, J., et al.: VoteTrust: leveraging friend invitation graph to defend against social network Sybils. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp. 2400–2408 (2013)
43. Wang, G., et al.: You are how you click: clickstream analysis for Sybil detection. In: Proceedings of 22nd USENIX Security Symposium, pp. 241–255 (2013)
44. Yu, H., Shi, C., Kaminsky, M., Gibbons, P., Xiao, F.: DSybil: optimal Sybil-resistance for recommendation systems. In: IEEE Symposium on Security and Privacy, pp. 283–298 (2009)
45. Piro, C., Shields, C., Levine, B.N.: Detecting the sybil attack in mobile ad hoc networks. In: 2006 Securecomm and Workshops, pp. 1–11. IEEE (2006)
46. Mahalle, P.N., Anggorojati, B., Prasad, N.R., Prasad, R.: Identity authentication and capability based access control (iacac) for the internet of things. J. Cyber Secur. Mobil. **1**(4), 309–348 (2013)
47. Granjal, J., Monteiro, E., Silva, J.S.: Network-layer security for the Internet of Things using TinyOS and BLIP. Int. J. Commun. Syst. **27**(10), 1938–1963 (2014). https://doi.org/10.1002/dac.2444
48. Granjal, J., Monteiro, E., Silva, J.S.: Enabling network-layer security on IPv6 wireless sensor networks. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, pp. 1–6 (2010). https://doi.org/10.1109/GLOCOM.2010.5684293

49. Brachmann, M., Garcia-Morchon, O., Kirsche, M.: Security for practical coap applications: issues and solution approaches. In: GI/ITG KuVS Fachgesprch Sensornetze (FGSN). Universitt Stuttgart (2011)
50. Granjal, J., Monteiro, E., Silva, J.S.: Enabling network-layer security on IPv6 wireless sensor networks. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, pp. 1–6. IEEE (2010)
51. Kent, S.: RFC4302-ipauthenticationheader (2005). https://tools.ietf.org/html/rfc4302
52. Kent, S.: RFC4303-IP Encapsulating Security Payload (ESP) (2005). https://tools.ietf.org/html/rfc4303
53. Shelby, Z., Hartke, K., Bormann, C. The constrained application protocol (CoAP) (2014). https://tools.ietf.org/html/rfc7252
54. Dierks, T.: The transport layer security (TLS) protocol version 1.2 (2008). https://tools.ietf.org/html/rfc5246
55. Granjal, J., Monteiro, E., Silva, J.S.: End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication. In: 2013 IFIP Networking Conference, pp. 1–9. IEEE (2013)
56. Kothmayr, T., Schmitt, C., Hu, W., Brnig, M., Carle, G.: {DTLS} based security and two-way authentication for the Internet of Things. Ad Hoc Netw. **11**(8), 2710–2723 (2013). https://doi.org/10.1016/j.adhoc.2013.05.003
57. Young, M., Boutaba, R.: Overcoming adversaries in sensor networks: a survey of theoretical models and algorithmic approaches for tolerating malicious interference. IEEE Commun. Surv. Tutorials. **13**(4), 617–641 (2011). https://doi.org/10.1109/SURV.2011.041311.00156
58. Huang, X., Xiang, Y., Bertino, E., Zhou, J., Xu, L.: Robust multi-factor authentication for fragile communications. IEEE Trans. Dependable Secure Comput. **11**(6), 568–581 (2014)
59. Reardon, J., Goldberg, I.: Improving Tor using a TCP-over-DTLS tunnel. In: Proceedings of the 18th Conference on USENIX Security Symposium, pp. 119–134. USENIX Association (2009)
60. Kwon, E.K., Cho, Y.G., Chae, K.J.: Integrated transport layer security: end-to-end security model between WTLS and TLS. In: Proceedings 15th International Conference on Information Networking, pp. 65–71. IEEE (2001)
61. Rahman, A., Dijk, E.: Group communication for coap. Group (2011)
62. Peretti, G., Lakkundi, V., Zorzi, M.: BlinkToSCoAP: an end-to-end security framework for the Internet of Things. In: 2015 7th International Conference on Communication Systems and Networks (COMSNETS), pp. 1–6. IEEE (2015)
63. Park, N., Kang, N.: Mutual authentication scheme in secure internet of things technology-forcomfortablelifestyle. Sensors. **6**(1), 20–20 (2016)
64. Henze, M., Wolters, B., Matzutt, R., Zimmermann, T., Wehrle, K.: Distributed configuration, authorization and management in the cloud-based internet of things. In: 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 185–192. IEEE (2017)
65. Zhou, J., Cao, Z., Dong, X., Vasilakos, A.V.: Security and privacy for cloud-based IoT: challenges. IEEE Commun. Mag. **55**(1), 26–33 (2017)
66. Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., Alonso-Zarate, J.: A survey on application layer protocols for the internet of things. Trans.. IoT Cloud Comput. **3**(1), 11–17 (2015)
67. Scott, C., Wolfe, P., Erwin, M.: Virtual private networks, ser. Animal Series. O'Reilly, Bejing (1999)
68. Granjal, J., Monteiro, E., Silva, J.S.: Application-layer security for the WoT: extendingCoAPtosupportend-to-endmessagesecurityforinternet-integrated sensing applications. In: International Conference on Wired/Wireless Internet Communication, pp. 140–153. Springer, Berlin/Heidelberg (2013)
69. Liu, C.H., Yang, B., Liu, T.: Efficient naming, addressing and profile services in Internet-of-Things sensory environments. Ad Hoc Netw. **18**, 85–101 (2014)
70. Conzon, D., Bolognesi, T., Brizzi, P., Lotito, A., Tomasi, R., Spirito, M.A.: The virtus middleware: an xmpp based architecture for secure iot communications. In: 2012 21st International Conference on Computer Communications and Networks (ICCCN), pp. 1–6. IEEE (2012)

71. XMPP Standards Foundation. XMPP. [Online]. http://xmpp.org/
72. OSGi Alliance. OSGi main. [Online]. http://www.osgi.org
73. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V.: Blockchain technology: beyond bitcoin. Appl. Innov. **2**(6-10), 71 (2016)
74. Double-Spending—Bitcoin WiKi, Mar. (2016). [Online]. Available: https://en.bitcoin.it/wiki/Double-spending
75. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper. **151**(2014), 1–32 (2014)
76. Cachin, C.: Architecture of the hyperledger blockchain fabric. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers, vol. 310, pp. 4 (2016)
77. Understanding Public Key Cryptography. [Online] (2005). Available: https://technet.microsoft.com/en-us/library/aa998077(v=exchg.65).aspx
78. Zheng, Z., Xie, S., Dai, H.N., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **2016**, 1–25 (2016)
79. TransActive Grid, Mar. (2016). [Online]. Available: http://transactivegrid.net/
80. Rutkin, A.: Blockchain-based microgrid gives power to con-sumers in New York (2016). [Online]. Available: https://www.newscientist.com/article/2079334-blockchain-based-microgrid-gives-power-to-consumers-in-new-york/
81. Otte, P., de Vos, M., Pouwelse, J.: TrustChain: a Sybil-resistant scalable blockchain. Futur. Gener. Comput. Syst. (2017) https://doi.org/10.1016/j.future.2017.08.048
82. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**(1), 36–63 (2001)

Pardeep Kaur is a Research Scholar in CSE at Chitkara University (Baddi (Himachal Pardesh)), India. In 2016, she received her Bachelor degree in CSE from Shaheed Udham Singh College of Engineering & Technology, Mohali. In 2019, she completed a 2 month internship on the project 'E-Voting system using blockchain' at the National Chung Cheng University, (Chiayi (Taiwan)). Her current areas of interest are blockchain and the Internet of Things.

Shalli Rani is Associate Professor in CSE at Chitkara University (Rajpura (Punjab)), India. She has 14+ years teaching experience. She received her MCA degree from Maharishi Dyanand University, Rohtak in 2004, her M.Tech. degree in Computer Science from Janardan Rai Nagar Vidyapeeth University, Udaipur in 2007, and her Ph.D. degree in Computer Applications from Punjab Technical University, Jalandhar in 2017. Her main areas of interest and research are Wireless Sensor Networks, Underwater Sensor networks and the Internet of Things. She has published/accepted/presented more than 35 papers in international journals/conferences. She has worked on Big Data, Underwater Acoustic Sensors, and the IoT to show the importance of WSN in IoT applications. She received a young scientist award in Feb. 2014 from Punjab Science Congress, in the same field.