Future Generation Computer Systems 82 (2018) 395–411

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

IoT security: Review, blockchain solutions, and open challenges

Minhaj Ahmad Khan [a,*], Khaled Salah [b]

[a] Bahauddin Zakariya University Multan, Pakistan
[b] Khalifa University of Science, Technology & Research, Sharjah, United Arab Emirates

HIGHLIGHTS

- IoT is a promising disruptive technology with incredible growth, impact and potential.
- A review of emerging topics related to Internet of Things (IoT) security and Blockchain is presented.
- A mapping of the major security issues for IoT to possible solutions is tabulated.
- Blockchain technology and its robust solutions for challenging and critical IoT security problems are reviewed.
- A parametric analysis of the state-of-the-art IoT security issues and solutions is described.

ARTICLE INFO

Article history:
Received 17 July 2017
Received in revised form 2 November 2017
Accepted 12 November 2017
Available online 26 November 2017

Keywords:
IoT security
Blockchain
IoT protocols
Network security
Data security

ABSTRACT

With the advent of smart homes, smart cities, and smart everything, the Internet of Things (IoT) has emerged as an area of incredible impact, potential, and growth, with Cisco Inc. predicting to have 50 billion connected devices by 2020. However, most of these IoT devices are easy to hack and compromise. Typically, these IoT devices are limited in compute, storage, and network capacity, and therefore they are more vulnerable to attacks than other endpoint devices such as smartphones, tablets, or computers.

In this paper, we present and survey major security issues for IoT. We review and categorize popular security issues with regard to the IoT layered architecture, in addition to protocols used for networking, communication, and management. We outline security requirements for IoT along with the existing attacks, threats, and state-of-the-art solutions. Furthermore, we tabulate and map IoT security problems against existing solutions found in the literature. More importantly, we discuss, how blockchain, which is the underlying technology for bitcoin, can be a key enabler to solve many IoT security problems. The paper also identifies open research problems and challenges for IoT security.

1. Introduction

With the rapid growth of smart devices and high speed networks, the Internet of Things (IoT) has gained wide acceptance and popularity as the main standard for low-power lossy networks (LLNs) having constrained resources. It represents a network where "things" or embedded devices having sensors are interconnected through a private or a public network [1,2]. The devices in IoT can be controlled remotely to perform the desired functionality. The information sharing among the devices then takes place through the network which employs the standard protocols of communication. The smart connected devices or "things" range from simple wearable accessories to large machines, each containing sensor chips. For instance, the Lenovo smart shoes contain chips which provide support of tracking and analyzing fitness data [3]. Similarly, the electrical appliances including washing machines, and refrigerators can be controlled remotely through IoT. The security cameras installed for surveillance of a location can be monitored remotely anywhere in the world.

Apart from the personal use, IoT serves the community needs as well. Various smart devices which perform diverse functionalities such as monitoring surgery in hospitals, detecting weather conditions, providing tracking and connectivity in automobiles, and identification of animals using biochips are already serving the community specific needs [4]. The data collected through these devices may be processed in real-time to improve efficiency of the entire system.

The future significance of IoT is evident due to its application in everyday life. It continues to grow rapidly due to evolution of hardware techniques such as improving bandwidth by incorporating cognitive radio based networks to address underutilization

* Corresponding author.
E-mail addresses: mik@bzu.edu.pk (M.A. Khan), khaled.salah@kustar.ac.ae (K. Salah).

https://doi.org/10.1016/j.future.2017.11.022
0167-739X/

of frequency spectrum [5,6]. In the literature, the Wireless Sensor Networks (WSNs) and Machine-to-Machine (M2M) or Cyber-Physical Systems (CPS) have now evolved as integral components for the broader term IoT. Consequently, the security problems related to WSN, M2M, or CPS continue to arise in the context of IoT with the IP protocol being the main standard for connectivity. The entire deployment architecture therefore needs to be secured from attacks which may hinder the services provided by IoT as well as may pose threat to privacy, integrity or confidentiality of data. Since the IoT paradigm represents a collection of interconnected networks, and heterogeneous devices, it inherits the conventional security issues related to the computer networks. The constrained resources pose further challenges to IoT security since the small devices or things containing sensors have limited power and memory. Consequently, the security solutions need to be adapted to the constrained architectures.

There has been a tremendous effort in recent years to cope with security issues in the IoT paradigm. Some of these approaches target security issues at a specific layer, whereas, other approaches aim at providing end-to-end security for IoT. A recent survey by Alaba et al. [7] categorizes security issues in terms of application, architecture, communication, and data. This proposed taxonomy for IoT security is different from the conventional layered architecture. The threats on IoT are then discussed for hardware, network, and application components. Similarly, another survey by Granjal et al. [8] discusses and analyzes security issues for the protocols defined for IoT. The security analyses presented in [9–11] discuss and compare different key management systems and cyrptographic algorithms. Similarly, the authors in [12–14] target a comparative evaluation of intrusion detection systems. An analysis of security issues for fog computing is presented in [15,16]. A survey by Sicari et al. [17] discusses contributions providing confidentiality, security, access control and privacy for IoT along with the security for middleware. The authors discuss trust management, authentication, privacy issues, data security, network security, and intrusion detection systems. For edge computing based paradigms including mobile cloud computing, mobile edge computing and fog computing, the identity and authentication, access control systems, network security, trust management, fault tolerance and implementation of forensics are surveyed in [18].

A survey of privacy preserving mechanisms for IoT is given in [19]. The author describes the secure multi-party computations to be enforced for preserving privacy of IoT users. The mechanisms of credit checking and attribute based access control are described to be effective solutions for privacy preserving in IoT. Zhou et al. [20] discuss different security threats and their possible countermeasures for cloud-based IoT. The authors describe identity and location privacy, node compromising, layer removing or adding, and key management threats for IoT using clouds. Another survey by Zhang et al. [21] discusses major IoT security issues in terms of unique identification of objects, authentication and authorization, privacy, the need for lightweight cryptographic procedures, malware, and software vulnerabilities. The IOT-a project [22] describes a reference architecture for IoT whose compliance requires implementation for trust, privacy, and security. The trust model is expected to provide data integrity and confidentiality while making end-to-end communication possible through an authentication mechanism. Moreover, to avoid improper usage of data, the privacy model requires defining access policies and mechanisms for encrypting and decrypting data. The security aspect incorporates three layers corresponding to the services, communication, and application. Similarly, the Open Web Application Security Project (OWASP) [23] describes top 10 vulnerabilities for the IoT architecture. These vulnerabilities include insecure interfaces of entities of the IoT architecture, inappropriate security configuration, physical security and insecure software/firmware.

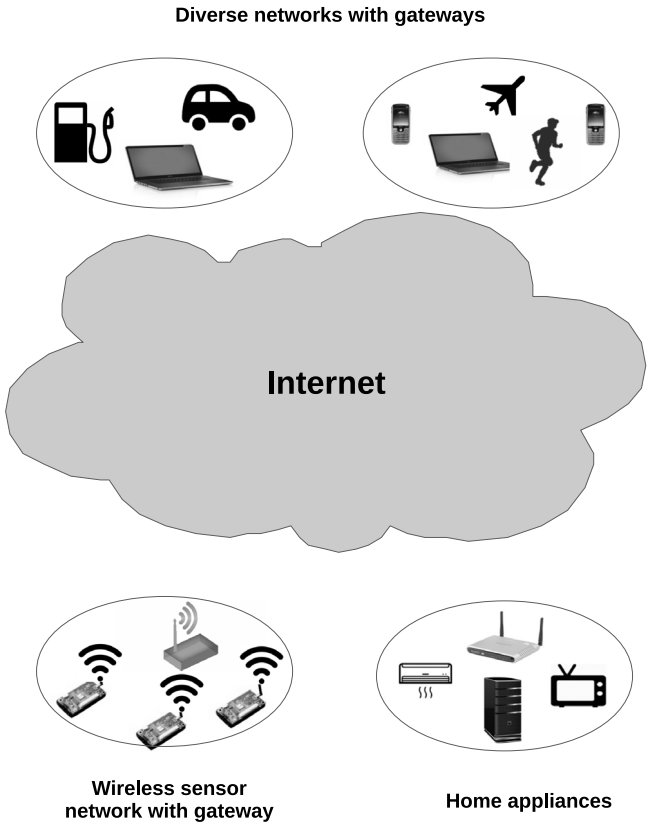

Fig. 1. An overview of IoT elements.

In sharp contrast to the survey articles found in the literature, our main contributions in this article can be summarized as follows:

- A parametric analysis of security threats and their mapping to possible solutions for IoT.
- Taxonomy and categorization of IoT security issues with respect to its layers, and the countermeasures used to address these issues.
- Discussion of basic characteristics of the blockchain based security solutions and analysis of their effectiveness for securing IoT.
- Future directions highlighting possible solutions for open IoT security problems.

The rest of the paper is organized as follows. Section 2 delineates the IoT architecture and the security challenges being faced at each layer of the protocol stack deployed by IoT. Section 3 categorizes the main security issues, whereas, Section 4 analyzes and describes a mapping of the solutions proposed. Various solutions related to blockchain security are discussed and analyzed in Section 5. In Section 6, we discuss the research challenges posing main hindrance to IoT security and their possible solutions before concluding the paper in Section 7.

2. IoT architecture and security challenges

A typical IoT deployment contains heterogeneous devices with embedded sensors interconnected through a network, as shown in Fig. 1. The devices in IoT are uniquely identifiable and are mostly characterized by low power, small memory and limited processing capability. The gateways are deployed to connect IoT devices to the outside world for remote provision of data and services to IoT users.

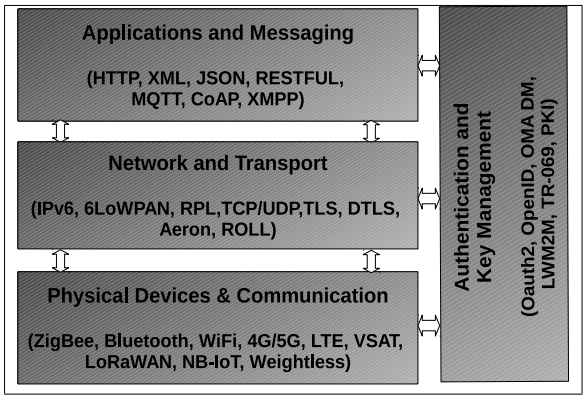

Fig. 2. Common IoT standards and protocols.

2.1. IoT protocols and standards

Fig. 2 depicts a layered architecture with the common IoT protocols used for applications & messaging, routing/forwarding, physical devices and those for key management and authentication. It includes the standards and protocols for the commonly used low rate wireless personal area networks (LR-WPANs) [24], and the recently evolved protocols for the low power wide-area-network (LPWAN) based protocols.

For LR-WPANs, the IEEE standard 802.15.4 describes two low-level layers: Physical Layer and the Medium Access Control (MAC) layer. The physical layer specification is related to communication over wireless channels having diverse frequency bands and data rates. The MAC layer specification is related to mechanisms for channel access as well as for synchronization. Due to a small size of maximum transmission unit (MTU) used by the IEEE 802.15.4 standard, an IPv6 over low-power wireless personal area network (6LoWPAN) adaptation layer is incorporated above the link layer in order to enrich sensor node with IP based communication capabilities. Each device in IoT is uniquely identified by an IPv6 network address. The Routing Protocol for Low-Power and Lossy Networks (RPL) [25] is used to support 6LoWPAN environments. The RPL standard supports point-to-point traffic as well as the communication between multi-points and single point.

Due to a limited payload, the application design in IoT incorporates User Datagram Protocol (UDP) [26] for communication as it is considered to be more efficient and less complex than TCP. Moreover, the UDP header compression may be performed for a better utilization of the limited payload space [27]. For control messages, such as specifying unreachable destination, and neighbor discovery, the Internet Control Message Protocol (ICMP) [28] is used by 6LoWPAN. The Constrained Application Protocol (CoAP) [29] provides a request–response based model for low-power lossy networks existing in constrained environments. The CoAP protocol supports asynchronous message communication and also provides HTTP mapping to access IoT resources through HTTP.

The LPWAN allows for a long range communication of "things" in IoT. In contrast to a wireless WAN which requires more power to work with a high bit-rate, it supports low-power communication with low bit-rate. The LPWAN uses LoRaWAN protocol for communication between gateways and the end devices while supporting varying data rates in a network of battery operated things. Similarly, the narrow-band IoT (NB-IoT) is a 3GPP protocol for communication in LPWANs to provide indoor coverage while using LTE spectrum. The Weightless protocol uses three different standards for communication in LPWAN to support uni-directional, bi-directional and low-power modes, respectively.

2.2. Security requirements for IoT

For a secure IoT deployment, various mechanisms and parameters need to be reckoned with as described below.

2.2.1. Data privacy, confidentiality and integrity

As IoT data travels through multiple hops in a network, a proper encryption mechanism is required to ensure the confidentiality of data. Due to a diverse integration of services, devices and network, the data stored on a device is vulnerable to privacy violation by compromising nodes existing in an IoT network. The IoT devices susceptible to attacks may cause an attacker to impact the data integrity by modifying the stored data for malicious purposes.

2.2.2. Authentication, authorization and accounting

To secure communication in IoT, the authentication is required between two parties communicating with each other. For privileged access to services, the devices must be authenticated. The diversity of authentication mechanisms for IoT exists mainly due to the diverse heterogeneous underlying architectures and environments which support IoT devices. These environments pose a challenge for defining standard global protocol for authentication in IoT. Similarly, the authorization mechanisms ensure that the access to systems or information is provided to the authorized ones. A proper implementation of authorization and authentication results in a trustworthy environment which ensures a secure environment for communication. Moreover, the accounting for resource usage, along with auditing and reporting provide a reliable mechanism for securing network management.

2.2.3. Availability of services

The attacks on IoT devices may hinder the provision of services through the conventional denial-of-service attacks. Various strategies including the sinkhole attacks, jamming adversaries or the replay attacks exploit IoT components at different layers to deteriorate the quality-of-service (QoS) being provided to IoT users.

2.2.4. Energy efficiency

Th IoT devices are typically resource-constrained and are characterized with low power and less storage. The attacks on IoT architectures may result in an increase in energy consumption by flooding the network and exhausting IoT resources through redundant or forged service requests.

2.3. Single points of failure

A continuous growth of heterogeneous networks for the IoT-based infrastructure may expose a large number of single-points-of-failure which may in turn deteriorate the services envisioned through IoT. It necessitates the development of a tamper-proof environment for a large number of IoT devices as well as to provide alternative mechanisms for implementation of a fault-tolerant network.

3. Categorization of security issues

As the IoT paradigm encompasses a wide variety of devices and equipment ranging from small embedded processing chips to large high-end servers, it needs to address security issues at different levels.

A taxonomy of security issues for IoT is given in Fig. 3 along with publication references related to each issue. We categorize the security threats with regard to the IoT deployment architecture as described below.

- Low-level security issues
- Intermediate-level security issues
- High-level security issues

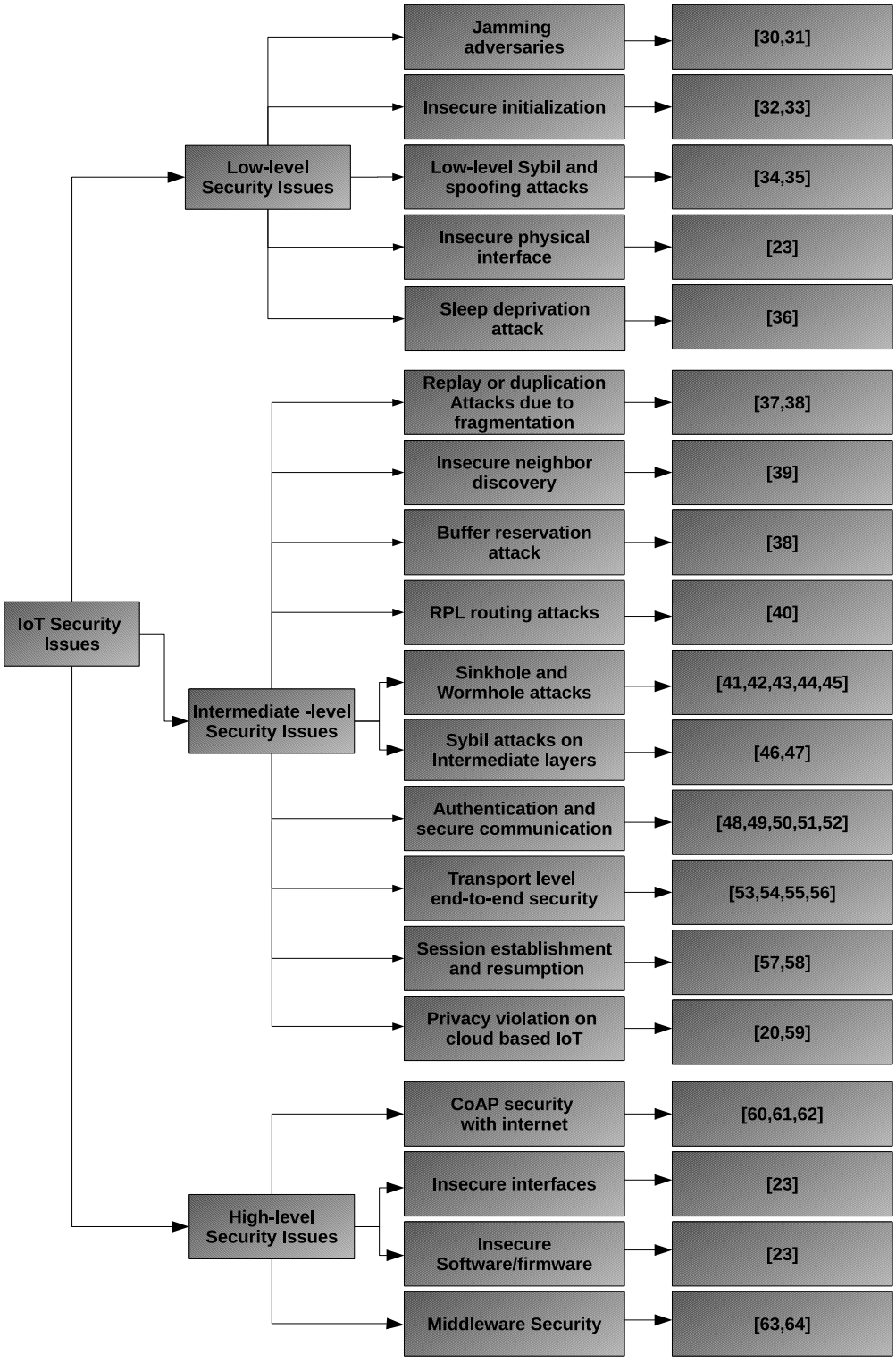

Fig. 3. A taxonomy of security issues and related publications.

3.1. Low-level security issues

The first level of security is concerned with the security issues at the physical and data link layers of communication as well as hardware level, as detailed below.

Jamming adversaries. The jamming attacks on wireless devices in IoT target deterioration of the networks by emitting radio frequency signals without following a specific protocol [30,31]. The radio interference severely impacts the network operations and can affect the sending and receiving of data by legitimate nodes, resulting in malfunctioning or unpredictable behavior of the system.

Insecure initialization. A secure mechanism of initializing and configuring IoT at the physical layer ensures a proper functionality of the entire system without violating privacy and disruption of network services [32,33]. The physical layer communication also

needs to be secured in order to make it inaccessible to unauthorized receivers.

Low-level Sybil and spoofing attacks. The Sybil attacks in a wireless network are caused by malicious Sybil nodes which use fake identities to degrade the IoT functionality. On the physical layer, a Sybil node may use random forged MAC values for masquerading as a different device while aiming at depletion of network resources [34,35]. Consequently, the legitimate nodes may be denied access to resources.

Insecure physical interface. Several physical factors compound serious threats to proper functioning of devices in IoT. The poor physical security, software access through physical interfaces, and tools for testing/debugging may be exploited to compromise nodes in the network [23].

Sleep deprivation attack. The energy constrained devices in IoT are vulnerable to "sleep deprivation" attacks by causing the sensor nodes to stay awake [36]. It results in depletion of battery when a large number of tasks is set to be executed in the 6LoWPAN environment.

3.2. Intermediate-level security issues

The intermediate-level security issues are mainly concerned with the communication, routing and session management taking place at network and transport layers of IoT as described below.

Replay or duplication attacks due to fragmentation. The fragmentation of IPv6 packets is required for devices conforming to the IEEE 802.15.4 standard which is characterized with small frame sizes. A reconstruction of the packet fragment fields at the 6LoWPAN layer may result in depletion of resources, buffer overflows and rebooting of the devices [37]. The duplicate fragments sent by malicious nodes affect the packet re-assembly, thereby hindering the processing of other legitimate packets [38].

Insecure neighbor discovery. The IoT deployment architecture requires every device to be identified uniquely on the network. The message communication taking place for identification must be secure to ensure that the data being transmitted to a device in the end-to-end communication reaches the specified destination. The neighbor discovery phase prior to transmission of data performs different steps including the router discovery and address resolution [39]. The usage of neighbor discovery packets without proper verification may have severe implications along with denial-of-service.

Buffer reservation attack. As a receiving node requires to reserve buffer space for re-assembly of incoming packets, an attacker may exploit it by sending incomplete packets [38]. This attack results in denial-of-service as other fragment packets are discarded due to the space occupied by incomplete packets sent by the attacker.

RPL routing attack. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is vulnerable to several attacks triggered through compromised nodes existing in the network [40]. The attack may result in depletion of resources and eavesdropping.

Sinkhole and wormhole attacks. With the sinkhole attacks, the attacker node responds to the routing requests, thereby making the packets route through the attacker node which can then be used to perform malicious activity on the network [41,42]. The attacks on network may further deteriorate the operations of 6LoWPAN due to wormhole attacks in which a tunnel is created between two nodes so that packets arriving at a node reach other node immediately [43–45]. These attacks have severe implications including eavesdropping, privacy violation and denial-of-service.

Sybil attacks on intermediate layers. Similar to the Sybil attacks on low-level layers, the Sybil nodes can be deployed to degrade the network performance and even violate data privacy. The communication by Sybil nodes using fake identities in a network may result in spamming, disseminating malware or launching phishing attacks [46,47].

Authentication and secure communication. The devices and users in IoT need to be authenticated through key management systems. Any loophole in security at network layer or large overhead of securing communication may expose the network to a large number of vulnerabilities [48–50]. For instance, due to constrained resources, the overhead of Datagram Transport Level Security (DTLS) requires to be minimized, and the cryptographic mechanisms ensuring secure communication of data in IoT must take into account the efficiency as well as the scarcity of other resources [51,52].

Transport level end-to-end security. The transport level end-to-end security aims at providing secure mechanism so that the data from the sender node is received by the desired destination node in a reliable manner [53,54]. It requires comprehensive authentication mechanisms which ensure secure message communication in encrypted form without violating privacy while working with minimum overhead [55,56].

Session establishment and resumption. The session hijacking on transport layer with forged messages can result in denial-of-service [57,58]. An attacking node can impersonate the victim node to continue the session between two nodes. The communicating nodes may even require re-transmission of messages by altering the sequence numbers.

Privacy violation on cloud-based IoT Different attacks which may violate identity and location privacy may be launched on cloud or delay tolerant network based IoT [20,59]. Similarly, a malicious cloud service provider on which IoT deployment is based, can access confidential information being transmitted to a desired destination.

3.3. High-level security issues

The high-level security issues are mainly concerned with the applications executing on IoT as described below.

CoAP security with internet. The high-level layer containing the application layer is also vulnerable to attacks [60–62]. The Constrained Application Protocol (CoAP) being a web transfer protocol for constrained device uses DTLS bindings with various security modes to provide end-to-end security. The CoAP messages follow a specific format defined in RFC-7252 [29], which need to be encrypted for secure communication. Similarly, the multicast support in CoAP requires adequate key management and authentication mechanisms.

Insecure interfaces. For accessing IoT services, the interfaces used through web, mobile, and cloud are vulnerable to different attacks which may severely affect the data privacy [23].

Insecure software/firmware. Various vulnerabilities in IoT include those caused by insecure software/firmware [23]. The code with languages such as JSON, XML, SQLi and XSS needs to be tested carefully. Similarly, the software/firmware updates need to be carried out in a secure manner.

Middleware security. The IoT middleware designed to render communication among heterogeneous entities of the IoT paradigm must be secure enough for provision of services. Different interfaces and environments using middleware need to be incorporated to provide secure communication [63,64].

4. Security solutions for IoT

The security threats in IoT exploit vulnerabilities of various components such as applications/interfaces, network components, software, firmware, and physical devices, existing at different levels. The users in an IoT paradigm interact with these components through protocols which may also be dismantled of their security

Table 1
Mapping of low-level IoT security threats, implications, and solutions.

Sr#	Security issue	Implications	Affected layers	IoT levels	Proposed solutions	References
1	Jamming adversaries	Disruption and denial-of-service	Physical layer	Low-level	Measuring signal strength,computing packet delivery ratio, encoding packets with error correcting codes, and change of frequencies and locations	[30,31,66]
2	Low-level Sybil and spoofing attacks	Network disruption, denial-of-service	Physical layer	Low-level	Signal strength measurements, and channel estimation	[34,35,68–70]
3	Insecure initialization and configuration	Privacy violation and denial-of-service	Physical layer	Low-level	Setting data transmission rates b/w nodes, and introducing artificial noise	[32,33,67]
4	Insecure physical interface	Privacy violation, denial-of-service	Hardware	Low-level	Avoiding software/firmware access to USB, hardware based TPM modules, and avoiding testing/debugging tools	[23]
5	Sleep deprivation attack	Energy consumption	Link layer	Low-level	Multi-layer based intrusion detection system	[36]

Table 2
Mapping of intermediate-level IoT security threats, implications, and solutions below transport layer.

Sr#	Security issue	Implications	Affected layers	IoT levels	Proposed solutions	References
1	Replay or duplication attacks due to fragmentation	Disruption and denial-of-service	6LoWPAN adaptation layer, and network layer	Intermediate-level	Introduction of timestamp and *nonce* options for protecting against replay attacks, and fragment verification through hash chains	[37,38]
2	Insecure neighbor discovery	IP Spoofing	Network layer	Intermediate-level	Authentication using Elliptic Curve Cryptography (ECC) based signatures	[39]
3	Buffer reservation attack	Blocking of reassembly buffer	6LoWPAN adaptation layer, and network layer	Intermediate-level	Split buffer approach requiring complete transmission of fragments	[38]
4	RPL routing attack	Eavesdropping, man-in-the-middle attacks	IPv6 network layer	Intermediate-level	Hashing and Signature based authentication, and monitoring node behavior	[40,75]
5	Sinkhole and wormhole attacks	denial-of-service	Network layer	Intermediate-level	Rank verification through hash chain function, trust level management, nodes/communication behavior analysis, anomaly detection through IDS, cryptographic key management, graph traversals, and measuring signal strength	[41–45,76–84]
6	Sybil attacks	Privacy violation, spamming, Byzantine faults, unreliable broadcast	Network layer	Intermediate-level	Random walk on social graphs, analyzing user behavior, and maintaining lists of trusted/un-trusted users	[46,47,86–89]
7	Authentication and secure communication	Privacy violation	6LoWPAN adaptation layer, transport layer, network layer	Intermediate-level	Compressed AH and ESP, Header compression and software based AES, TPM using RSA, SHA1/AES, hybrid authentication, authentication with fuzzy extractor, encryption of payload dispatch type values with compressed AH, IACAC using the Elliptic Curve Cryptography, distributed logs, and symmetric homomorphic mapping	[48–52,92,93,96,99,101,59,20,100]

measures. The countermeasures for security threats address vulnerabilities of this interaction at different layers to attain a specific security level. The diverse protocols supporting deployment of components add to the complexity of these countermeasures. A review of major security solutions proposed in the literature is given in this section. A comparative analysis of the security threats, and their possible solutions is given for the low-level, intermediate-level below transport layer, intermediate-level involving transport layer, and high-level in Tables 1–4, respectively. The comparative analysis considers the parameters of threats, their implications, affected layers, corresponding levels and the possible solutions proposed in the literature.

4.1. Low-level security solutions

For wireless sensor networks, the jamming attacks relate to interference resulting in message collisions or flooding the channels. An approach for detection of jamming attacks is proposed by Young et al. [65]. The detection of attacks is made possible by measuring the signal strength which is then used for extracting noise-like signals. These statistics are then compared with customized threshold

Table 3
Mapping of intermediate-level IoT security threats, implications, and solutions involving transport layer.

Sr#	Security issue	Implications	Affected layers	IoT levels	Proposed solutions	References
1	Transport level end-to-end security	Privacy violation	Transport layer, and network layer	Intermediate-level	DTLS-PSK with *nonces*, 6LoWPAN Border Router with ECC, DTLS cipher based on AES/SHA algorithms, compressed IPSEC, DTLS header compression, IKEv2 using compressed UDP, and AES/CCM based security with identification and authorization	[53–56,92,93,102–105]
2	Session establishment and resumption	denial-of-service	Transport layer	Intermediate-level	Authentication with long-lived secret key, and symmetric key based encryption	[57,58,106]

Table 4
Mapping of high-level IoT security threats, implications, and solutions.

Sr#	Security issue	Implications	Affected layers	IoT levels	Proposed solutions	References
1	CoAP security with internet	Network bottleneck, denial-of-service	Application layer, and network layer	High-level and intermediate-level	TLS/DTLS and HTTP/CoAP mapping, Mirror Proxy (MP) and Resource Directory, TLS-DTLS tunnel and message filtration using 6LBR	[60–62,108]
2	Insecure interfaces	Privacy violation, denial-of-service, network disruption	Application layer	High-level	Disallowing weak passwords, testing the interface against the vulnerabilities of software tools (SQLi and XSS), and using *https* along with firewalls	[23]
3	Insecure software/firmware	Privacy violation, denial-of-service, network disruption	Application layer, transport layer, and network layer	High-level, intermediate-level, and low-level	Regular secure updates of software/firmware, use of file signatures, and encryption with validation	[23]
4	Middleware security	Privacy violation, denial-of-service, network disruption	Application layer, transport layer, and network layer	High-level, intermediate-level, and low-level	Secure communication using authentication, security policies, key management between devices, gateways & M2M components, service layer M2M security, transparent middleware using authentication/encryption mechanisms	[63,109,64,110,111]

values for attack detection. Xu et al. [30] suggest an approach of detecting jamming attacks through computation of successful packet delivery ratio. The proposed algorithms work by performing consistency checks on signal strength and locations of the nodes. Another anti-jamming mechanism using cryptographic functions and error correcting codes is proposed by Noubir et al. [31]. The approach works by encoding packets through division into blocks and interleaving the encoded packet bits. Similarly, the strategies incorporating channel surfing and spatial retreats are also proposed to cope with jamming attacks [66]. The channel surfing enables the legitimate communicating devices to change channel frequencies, whereas, the spatial retreats causes these devices to change their location while moving to a desired location at some specific distance.

For a secure physical layer communication, a framework aimed at secure initialization of IoT is proposed by Pecorella et al. [67]. A minimum data rate is configured between the sending and receiving nodes to ensure absence of eavesdroppers. Other approaches of introducing artificial noise [32,33] in signals are also used for securing the communication.

A malicious Sybil node may use fake MAC values for masquerading as a different device. It can result in resource exhaustion as well as denial of access to legitimate devices in the network. An approach of detecting Sybil attacks using signal strength measurements is given by Demirbas et al. [68]. Their approach works by deploying detector nodes to compute the sender location during message communication. Another message communication with the same sender location but different sender identity is implied as a Sybil attack. The assumptions of the proposed approach make it applicable to static networks. Other approaches by Chen et al. [35] and Li et al. [69] use signal strength measurements for MAC addresses for detecting spoofing attacks. Another approach by Xiao et al. [34] incorporates channel estimation for detecting Sybil attacks. The approach uses number of identities and other parameters related to channel estimation for detecting Sybil nodes. Similarly, the approach in [70] uses channel response to differentiate between legal users and attackers.

The devices having improper physical security are characterized with having external interfaces providing firmware or software access, and providing vulnerable utility tools such as those for testing and debugging. The Open Web Application Security Project (OWASP) provides recommendations to improve physical security of the devices in IoT [23]. The unnecessary hardware interfaces such as USBs providing access to the device firmware/software must be avoided. The testing and debugging tools must be disabled and hardware based mechanisms such as Trusted Platform Modules (TPMs) should be incorporated to improve physical security.

A framework for mitigating sleep deprivation attacks in wireless sensor networks is described in [36]. The proposed framework incorporates a cluster based approach where each cluster is divided into several sectors. The energy consumption is reduced by avoiding long distance communication. The framework performs intrusion detection with a 5-layers model of the wireless sensor network. A cluster coordinator contains an extended intrusion detection system together with the leader nodes and sink nodes in upper layers of the WSN model. Similarly, the follower nodes existing in lower layers of the WSN model are equipped with simple intrusion detection systems.

4.2. Intermediate-level security solutions

The threats arising from replay attacks due to fragmentation of packets in 6LoWPAN are addressed by adding timestamp and

nonce options to the fragmented packets [37]. These packets are added to the 6LoWPAN adaptation layer corresponding to the fragmented packets. The timestamp option and the *nonce* option work for the unidirectional and bi-directional packets, respectively. The 64-bit timestamp value in the fragment ensures to eliminate the redundant advertisements and redirects in the network. The *nonce* option ensures that the advertisement is only made to respond to a fresh solicitation. Similarly, a content-chaining strategy which ensures an in-order transmission of fragments of IPv6 packets in 6LoWPAN is proposed in [38]. The fragment contents are added to the hash-chain generation in order to verify the fragments.

A security framework with modules for secure neighbor discovery, authentication, key generation and data encryption is proposed by Riaz et al. [39]. For secure neighbor discovery, the Elliptic Curve Cryptography (ECC) [71] is used. The ECC public key signatures are used to identify nodes in the neighbor discovery phase. Both symmetric and asymmetric key management systems are proposed to be deployed depending upon the application requirements. The encrypted data is then communicated to ensure node-to-node security.

Through a buffer reservation attack, the reassembly buffer of a node may be blocked. This attack is mitigated through split buffer approach [38] which increases the cost of launching attack by requiring complete fragmented packets to be transmitted in short bursts. Every node is required to compute the percentage of completion of a packet and record the behavior of sending fragments. Upon overload, the node can discard the packets with low percentage or having large variation in fragment sending pattern.

For mitigating adversary attacks during routing through the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), a security service for authentication of rank and version numbers is proposed by Dvir et al. [40]. The RPL protocol works by constructing the Directed Acyclic Graph (DAG) with root at any of the gateways. The version number is updated when a new version of the destination-oriented DAG is being constructed. RPL uses ranks for representing the quality of route to the final sink node. The rank value of a node may be decreased to connect to the root for eavesdropping. The proposed security mechanism termed Version Number and Rank Authentication (VeRA) uses the hash function (SHA [72]), MAC function (HMAC [73] and digital signature (RSA [74] etc.) for authenticating version numbers and ranks. Similarly, with RPL, the rank value computed on the basis of rank of the preferred parent is broadcast to other nodes. The RPL standard requires the parent node to have lower rank than the children. An attack proposed by Le et al. [75] sets the malicious node to select the worse parent instead of the best parent. The compromised node does not forward the DAO messages, thereby adding traffic delay during transmission through malicious nodes. The attack becomes more severe depending upon the forwarding load of the network area. To mitigate the attack, it is suggested to monitor node behavior for various parameters including the messages delivered and the end-to-end delay etc.

To cope with sinkhole attacks in the low-power lossy networks, a mechanism incorporating failover and authentication techniques is proposed by Weekly et al. [41]. For rank verification corresponding to a Destination Information Object (DIO) message, a one-way hash function is used together with a hash chain function. A hash value generated for a random number selected by the root node is broadcast through DIO message. The valid nodes on the network perform further hashing before forwarding messages, whereas, the compromised nodes communicate the received hash values. After convergence of routing tree, the root performs a broadcast of the initially selected random value for verification by individual nodes. A mismatch at a node implies an unauthentic parent rank value. Similarly, the parent failover technique augments the DIO message with a special field signed by the root node. The special field represents the non-root nodes which are unable to transmit 30% of sensor data at specified intervals. The parents of such nodes are therefore blacklisted for subsequent communication. Another approach by Firoz et al. [42] identifies suspicious nodes by analyzing the behavior of the neighbor nodes. It then requires the suspicious node to be verified as a black hole. Another strategy to countermeasure sinkhole attacks using different trust levels is given by Pirzada et al. [43]. Their approach uses different features of the Dynamic Source Routing (DSR) protocol for detecting and avoiding the sinkhole and wormhole attacks in wireless networks. The approach is based on the accuracy and sincerity levels which are computed by verifying the forwarded packets through some integrity checks. Similarly, an ad hoc routing protocol [76] incorporating the symmetric cryptography based algorithms is designed for securing the nodes from the compromised wireless nodes in a network. Another approach of detecting wormhole attacks in wireless sensor networks works by broadcasting distances estimated between neighbors [44]. The network distortions are then analyzed to detect wormholes and suspicious neighbor connections. Another approach suggested by Wazid et al. [45] aims at detecting sinkhole or wormhole attacks for a hierarchical wireless sensor network. The entire network is distributed into several clusters with each cluster containing a high power sensor node which works for detecting sinkholes for its cluster. Various approaches using an intrusion detection mechanism for detecting and avoiding sinkhole attacks are proposed [77,78]. The proposed strategies incorporate an analysis of network packets, and anomaly detection using pre-defined rules. Other approaches of wormhole detection employ the network graphs [79–81], analyzing signal strength messages [82], or key management systems [83,84].

The Sybil attacks on network layer use pseudo-identities to mimic multiple unique identities termed as Sybil nodes [85]. These attacks pose a serious threat to distributed as well as peer to peer (p2p) systems including IoT. These attacks may also affect defense against Byzantine faults thereby producing hindrance in reliable broadcast in the network. For social networks, a trust relationship is incorporated to limit the creation of Sybil identities. The countermeasures using social graphs make it possible for legitimate nodes to detect Sybil nodes by traversing the graph through random walks or using the community detection algorithms [46,86–88]. Similarly, the users' behavior regarding activities on the network are analyzed, and subsequently, the users with a fixed pattern of activities are assumed to be Sybil users [47,46]. For mobile networks, the lists of trusted and untrusted users may be maintained to detect Sybil nodes [89].

For securing network layer of IPv6 based networks using 6LoWPAN adaptation layer, the compressed formats of Authentication Header (AH) [90] and Encapsulating Security Payload (ESP) [91] are proposed in [50]. The 8-bits of the IPv6 addressing header as per 6LowPAN adaptation layer specification are used to define header dispatch and addressing types. The compressed headers are used in communication in two modes: transport mode and tunnel mode, depending upon the payload encryption. The evaluation of different encryption techniques implemented for the new proposed security format shows the SHA1 [72] algorithm to have less time and energy requirements. Similarly, a compressed format of IPsec is described by Raza et al. [49,92,93] for providing end-to-end security. The authors use the authentication header (AH) and the Encapsulating Security Payload (ESP) for providing security using IPsec. The encodings for AH and ESP headers are performed using NHC encoding which is defined in HC13 compression mechanism [94]. For authentication and encryption, different variants of SHA1 and AES are implemented. The bitwise encodings result in a reduced packet size, however, the proposed approach incurs overhead in terms of the energy consumption and average response time.

Another mechanism for securing network layer of 6LoWPAN by supporting new dispatch type values is suggested by Granjal et al. [48]. The authors propose the usage of reserved values of the payload byte as given in RFC 4944 [95]. The first 3 bits of the dispatch type values describe the security header and the usage mode, whereas, the remaining 3 bits describe the types of 6LoWPAN addressing headers. To extract information from a packet regarding the cryptographic algorithms and the keys to be applied for processing the packet, a 2-byte Security Parameters Index (SPI) is used. In contrast to this approach, Mahalle et al. [51] propose a protocol for securing IoT against denial-of-service (DoS), man-in-the-middle, and replay attacks. The DoS attacks may arise on constrained devices as the attackers may send messages for utilization of resources. Similarly, the secret keys revealed due to eavesdropping may result in identity theft due to main-in-the-middle attacks in a networked environment. Moreover, the identity information or credentials can be replayed by attackers to affect network traffic. The proposed approach called Identity Authentication and Capability based Access Control (IACAC) generates secret keys using the Elliptic Curve Cryptography based Diffie Hellman algorithm. For communication and access, the devices are mutually authenticated through encryption with secret keys. A capability based access control is implemented where the capability represents a structure containing access rights and the device identifier. With the capability based access, the communication to take place between two devices is first verified. Moreover, the capability of the device to perform the desired functionality is checked before the actual operation takes place.

Kothmayr et al. [96,97] describe an approach for end-to-end security using two-way authentication through public key cryptography. A trusted Access Control server is deployed to store access rights of publishers in the network. The certificate of the publisher and the Certificate Authority (CA) must exist on the publisher site. The authentication may be performed via the Trusted Platform Module (TPM) [98] chips using RSA or through the DTLS pre-shared keys. With TPMs, the RSA certificates are transmitted in X.509 format. The end-to-end communication is set to take place only after authentication of subscribers with the Access Control server. The proposed approach is shown to work with low energy and memory requirements. Another authentication and authorization scheme based on multiple factors is proposed by Huang et al. [99]. The proposed scheme incorporates password authentication while using smart cards. A fuzzy extractor is then used for extracting secret random string from biometrics. The authentication protocol supports four major operations related to creation of security parameters, storing registration information in a database, authentication, and modification of authentication credentials. The authors also suggest a stand-alone authentication mechanism where the connectivity to the authentication server is not functional.

A distributed framework for secure communication among IoT networks is proposed by Henze et al. [59]. To protect an IoT network from a malicious cloud services provider, the proposed framework allows for configuration of IoT network from a central location. It logs control messages at multiple locations, in order to be verified through different gateways. The size of log messages is minimized by removing old messages continuously. The verification of log messages is then used to indicate malicious behavior which in turn protects cloud-based IoT from modification, withholding, insertion and reordering of messages. Aimed at preserving privacy for identity and location on cloud-based IoT, an authentication mechanism with secure packet forwarding is given by Zhou et al. [20]. The proposed algorithm uses symmetric homomorphic mapping for delay tolerant networks which lack a consistent end-to-end connectivity thereby requiring the intermediate nodes to cooperate during transmission of messages. Similarly, a platform for securing data shared among IoT devices is proposed in the SMARTIE project [100]. The data platform proposed by the SMARTIE project defines an authentication scheme for accessing the service along with different libraries for management of cryptographic keys. For providing a secure channel for communication between IoT devices and cloud, the project defines the light weight secure CoAP protocol using elliptic curve cryptography. Similarly, for preserving privacy during data sharing and providing secure tracking of IoT objects, it includes middleware and location based services.

For providing end-to-end security, the usage of TLS-PSK is suggested by Brachmann et al. [53] while making communication possible between HTTP and CoAP. This requires message translation from the DTLS layer and other high-level protocols. Similarly, for securing multicast messages, an extension of DTLS incorporating PSK and *nonces* is suggested to support negotiation of session keys. For transport level security, a delegated authentication mechanism is also proposed using the 6LoWPAN Border Router (6LBR) which intercepts the packets, performs computation for the public key authentication and forwards the packets [54]. An Access Control server is incorporated to support authentication between 6LBR and the sensing devices. The Elliptic Curve Cryptography (ECC) is employed for implementation of transport level security. The end-to-end communication is secured with 6LBR negotiating keys and communicating for other authentication steps between both the ends. The computation delegated to 6LBR results in better performance for secure communication despite heavy computations required by ECC. Another framework termed BlinkToSCoAP for providing end-to-end security in IoT is suggested in [55]. The proposed framework incorporates lightweight implementations of CoAP, DTLS, and 6LoWPAN for securing IoT. The DTLS cipher is based on the 128-bit AES and 26-bit SHA algorithms. On resource constrained devices, the proposed framework is shown to work with minimum requirements in terms of RAM size, flash memory size, and energy consumption. A strategy incorporating header compression for 6LoWPAN protocol for reducing DTLS overhead is given by Sinthan et al. [52]. The proposed strategy performs the DTLS header compression and uses software based AES implementation. The compression strategy improves energy consumption as well as the network response time.

The RERUM project [104] proposes a framework for Smart City IoT applications to ensure privacy and security. For developing trustworthy applications, the data integrity and authentication based approaches are being adapted. New access control mechanisms for dynamically switchable systems, along with hop-to-hop and end-to-end authentication are used to secure communication to/from objects in IoT. It also aims to ensure privacy through signature schemes and compressive sensing techniques. Another framework for experimenting with security protocols in an IoT-based infrastructure is implemented in the ARMOUR project [107]. The project aims at vaildation of trust and security for IoT-based scenarios like Smart City and Healthcare. The ARMOUR experimentation defines the security architecture, establishes testbeds, executes experiments, and generates certification labels. The experiments can be used to ensure reliable end-to-end connectivity as well as layer specific security requirements. Similarly, the BUTLER project [105] provides support of context-aware information systems for IoT systems including smart homes, smart shopping, smart healthcare, and smart cities. The services implemented in the project enable reliable communication of IoT objects using context information. The project included lightweight cryptographic protocols with the aim to improve confidentiality and integrity of data.

Different header compression techniques have been proposed for providing transport level end-to-end-security. Raza et al. [103] describe an approach for compressing DTLS Record and Handshake headers together with different Handshake messages so as to fit

within a single MTU of 6LoWPAN. The proposed approach encodes the header bits for combined encoding of Record and Handshake byte as well as for individual encoding of Record header after the Handshake is completed. Similarly, an enhanced version of DTLS incorporating header compression for securing IoT is presented in [102]. For UDP based next header compression (NHC), special 05 bits in the DTLS header are used to identify compressed headers, whereas the remaining 03 bits are used to represent checksum and ports. Similarly, for the record and handshake headers having size of 13 bytes and 12 bytes, the proposed strategy compresses the headers to the size of 05 bytes and 03 bytes, respectively. For CoAP, the enhancement in DTLS incorporating header compression reduces the DTLS overhead, thereby improving energy consumption and response time. A lightweight implementation of Internet Key Exchange (IKE) aimed at improving key management for 6LoWPAN is proposed by Shahid et al. [56]. The IKE protocol is used by IPSec for managing keys, however, it is considered to be inappropriate for resource constrained devices. The authors propose a compressed version of IKEv2 using a compressed UDP format which may be recognized as IKE header. Different fields in the IKE header are compressed while using NHC encoding mechanism. It is also proposed to use protocol ID field in the security association payload of IKEv2 for the IEEE 802.15.4 link layer security.

A mutual authentication scheme for secure session management using symmetric key based encryption methods is given by Park et al. [57]. The proposed scheme initially selects a random number, and performs encryption, and generates a session key which is subsequently used for encryption of another random number. The encrypted value is then used for authentication. For each session, a new session key may be generated without requiring repetition of parameters. Similarly, another method of encryption using hashes for resource-constrained devices supporting hash functions is also proposed. It works in an efficient manner due to small overhead of computations. Another scheme of mutual authentication for fog computing based environments having resource constrained devices is suggested in [58]. The proposed scheme termed Octopus requires to have a long-lived secret key which is then used for authentication with any of fog servers.

An adaptation of the HIP Diet Exchange has been used for improving IoT security by Hummen et al. [106]. By incorporating an efficient session resumption technique, the overhead of public key based operations is reduced. The session resumption results in peers performing heavy operations while initializing the session establishment. The session state is then stored which subsequently improves the session resumption with re-authentication. The negotiation required for session resumption may also be integrated into DTLS and IKEv2.

4.3. High-level security solutions

To secure CoAP based Low-power and Lossy Network (LLN) connected with internet, an approach incorporating TLS and DTLS is proposed by Brachmann et al. [60]. The proposed approach works for scenarios where the 6LoWPAN Border Router (6LBR) connects the LLN with internet in order to access devices remotely. The LLN nodes are used to provide services to CoAP and HTTP clients. A mapping of TLS and DTLS is proposed to provide end-to-end security which protects LLNs from internet-based attacks. The mapping computation when delegated to the resource-constrained devices may however incur a substantial overhead. Another approach of securing messages for applications communicating through internet using various CoAP security options is suggested by Granjal et al. [61]. The new security options related to CoAP are: *SecurityOn*, *SecurityToken* and *SeurityEncap*. The *SecurityOn* option relates to protection of CoAP messages at the application level. The *SecurityToken* option facilitates identification and authorization for providing access to CoAP resources at the application level. The *SecurityEncap* option uses configuration of the *SecurityOn* option and mainly performs transmission of data required for authentication and protection against replays. An AES/CCM based security is incorporated for protecting the messages. Using the above options, the proposed approach is shown to perform well in terms of the packet payload space, energy consumption and the communication rate. Similarly, for IoT based on IP networks, a security model with 6LBR being used for message filtration in order to provide end-to-end security is suggested in [108]. The TLS-DTLS tunnel can be created while 6LBR is used for mapping during the handshake. Similarly, with two hosts sharing a common key, the message verification or detecting replays is suggested to be performed at the CoAP device.

An energy efficient security model using public key cryptography for IoT based CoAP is proposed by Sethi et al. [62]. The suggested security model implemented through a prototype uses a Mirror Proxy (MP) and Resource Directory representing the server to serve requests during sleep state and the list of resources on the server (or endpoints), respectively. The MP registers the endpoints, adds the resources in a resource tree and also stores the public keys of endpoints. The resources are accessed by clients through resource identifiers. The public keys is transmitted to the client which is subsequently used for authentication of data updates. The prototype implementation is shown to require small amount of energy for resource-constrained devices.

The OWASP project [23] provides recommendations of countermeasures for securing IoT. To cope with insecure high-level interfaces, the security mechanisms include the configurations which discourage weak passwords, testing the interface against the well-known vulnerabilities of software tools (SQLi and XSS), and the usage of *https* along with the firewalls. Moreover, the software or firmware installed on the device should be updated regularly through an encrypted transmission mechanism. The updated files should be downloaded from a secure server and these files must be signed and properly validated prior to installation.

The VIRTUS middleware [63] proposed by Conzon et al. [63] implements authentication and encryption to secure distributed applications running in an IoT environment. The middleware uses an event-drive communication approach while incorporating TLS and SASL for data integrity, XML stream encryption and validation. The authentication mechanism ensures data exchange and access to resources only for authorized users. The VIRTUS middleware integrated with web services results in implementation of reliable and scalable IoT applications. A semantic framework called *Otsopack* [109] acts as a middleware to make heterogeneous implementations interact in a secure manner. For interoperability, it uses Triple Space Computing (TSC) based semantic format for interaction between applications running within a virtual space. For secure data exchange, an Open-ID based security solution is proposed. An identity provider is used to grant access of limited data to authorized users.

A middleware server which supports data filtering during communication among heterogeneous IoT environments is proposed by Liu et al. [64]. The proposed middleware supports efficient mechanism for naming, addressing, and profiling across heterogeneous environments. The standard authentication, authorization, and accounting (AAA) features are implemented through a key hierarchy with keys for root, applications, and services. For service registration, a web-based portal is implemented to provide access to services only to authorized users. For machine to machine (M2M) communications in the IoT environment, a standard architecture with different layers for security is proposed [110]. The proposed architecture includes layers for security services corresponding to security functionality, environment, and abstraction. For M2M service layer security, the resource contents are suggested to be encrypted along with secure message exchange

using TLS or DTLS sessions. Another security architecture for IoT middleware is proposed by Ferreira et al. [111]. It uses standard encryption methods such as AES to provide data privacy. Similarly, the end-to-end security and open authentication mechanisms are implemented. The deployments based on proposed architecture are able to secure communication for IoT entities including users, objects, and services.

5. Blockchain solutions for IoT security

Blockchain technology has been foreseen by industry and research community as a disruptive technology that is poised to play a major role in managing, controlling, and most importantly securing IoT devices. This section describes how blockchain can be a key enabling technology for providing viable security solutions to todays challenging IoT security problems. The section first gives a brief background about blockchain, and then outlines open research IoT security problems and challenges which blockchain may provide solutions for. The section also surveys the literature of blockchain-based solutions for IoT security problems.

5.1. Background

A blockchain is fundamentally a decentralized, distributed, shared, and immutable database ledger that stores registry of assets and transactions across a peer-to-peer (P2P) network. It has chained blocks of data that have been timestamped and validated by miners. The blockchain uses elliptic curve cryptography (ECC) and SHA-256 hashing to provide strong cryptographic proof for data authentication and integrity [112]. Fundamentally, the block data contains a list of all transactions and a hash to the previous block. The blockchain has a full history of all transactions and provides a cross-border global distributed trust. Trusted Third Parties (TTP) or centralized authorities and services can be disrupted, compromised or hacked. They can also misbehave and become corrupt in the future, even if they are trustworthy now. In blockchain, each transaction in the shared public ledger is verified by a majority consensus of miner nodes which are actively involved in verifying and validating transactions. In a bitcoin network [113], miners validate the block by computing a hash with leading zeros to meet the difficulty target. Once transactions are validated and verified by consensus, block data are immutable, i.e. data can never be erased or altered. Blockchain can be built as: (1) permissioned (or private) network that can be restricted to a certain group of participants, or (2) permission-less or public network that is open for anyone to join in. Permission blockchains provide more privacy and better access control.

Fig. 4 depicts a typical design structure of a Blockchain. The design structure is composed mainly of the block header and the block body which contains a list of transactions. The block header contains various fields, one of which is a version number to track software of protocol upgrades. Also, the header contains a timestamp, block size, and the number of transactions. Merkle root field represents the hash value of the current block. Merkle tree hashing is commonly used in distributed systems and P2P networks for efficient data verification. The *nonce* field is used for the proof-of-work algorithm, and it is the trial counter value that produced the hash with leading zeros. The difficulty target specifies the number of leading zeros, and is used to keep the blocktime approximately 10 min for Bitcoin [114], and 17.5 s for Ethereum [115]. The difficulty target is adjustable periodically and is increased (with more leading zeros) as the computation power of hardware increases over time. The blocktime is set by design to account for the propagation time of blocks to reach all miners, and for all miners to reach a consensus.

Bitcoin is one of the first and the most popular applications that runs on the top of blockchain infrastructure. In general, bitcoin blockchain has been the underlying platform and technology of many of todays most popular cryptocurrencies. However, with the advent of the Ethereum blockchain, which implements smart contracts, the potential use space for blockchain has become endless. Ethereum blockchain was launched and opened for use to the public in July 2015. Afterward, similar smart-contract blockchain platforms have recently emerged. Those include Hyperledger [116], Eris [117], Stellar [118], Ripple [119], and Tendermint [120]. As opposed to bitcoin blockchain which is primarily used for digital currency transactions, Ethereum blockchain has the ability to store records, and more importantly run smart contracts. The term smart contracts was first coined by Nick Szabo in 1994. A smart contract is basically a computerized transaction protocol that executes the terms of the contract. In the simplistic definition, smart contracts are programs written by users to be uploaded and executed on the blockchain. The scripting or programming language for smart contracts is called Solidity which is a JavaScript-like language. Ethereum Blockchain provides EVM (Ethereum Virtual Machines) which are basically the miner nodes. These nodes are capable of providing cryptographically tamper-proof trustworthy execution and enforcement of these programs or contracts.

Ethereum supports its own digital currency called Ether. As in bitcoin, in Ethereum, users can transfer coins to each other using normal transactions which get recorded on the ledger, and for such transactions, there is no need for a blockchain state in bitcoin. However, for Ethereum to support smart contract execution, a blockchain state is used, as shown in Fig. 4. A smart contract has its own account and address, and associated with it is its own executable code and balance of Ether coins. The storage is persistent and holds the code to be executed on the EVM nodes. EVM storage is relatively expensive, and for large storage to be uploaded to the blockchain, another remote decentralized data store like BitTorrent, IPFS, or Swarm can be used. The smart contracts, however, can hold a validation hash of such remotely stored information.

The possible use cases and applications of smart-contract blockchain applications are immense and endless, extending from cryptocurrency and trading to autonomous machine-to-machine transactions, from supply chain and asset tracking to automated access control and sharing, and from digital identity and voting to certification, management, and governance of records, data, or items [121]. The commercial deployments based on blockchains are increasing rapidly. For instance, SafeShare [122] has offered insurance solution using blockchain based on bitcoin. Similarly, IBM has launched its blockchain framework using Hyperledger Fabric platform [123]. The framework supports development of blockchain applications, and in contrast to other frameworks, it does not require cryptocurrency. The IBM blockchain is being used commercially in banks, supply chain systems, and cargo shipping companies.

5.2. Potential blockchain solutions

In the context of IoT, blockchain based on smart contracts is expected to play a major role in managing, controlling, and most importantly securing IoT devices. In this section, we discuss and summarize some of the intrinsic features of blockchain that can be immensely useful for IoT in general, and IoT security in particular.

Address Space. Blockchain has a 160-bit address space, as opposed to IPv6 address space which has 128-bit address space [112]. A blockchain address is 20 bytes or a 160-bit hash of the public key generated by ECDSA (Elliptic Curve Digital Signature Algorithm). With 160-bit address, blockchain can generate and allocate addresses offline for around $1.46 * 10^{48}$ IoT devices. The probability of address collision is approximately 10^{48}, which is considered sufficiently secure to provide a GUID (Global Unique Identifier) which

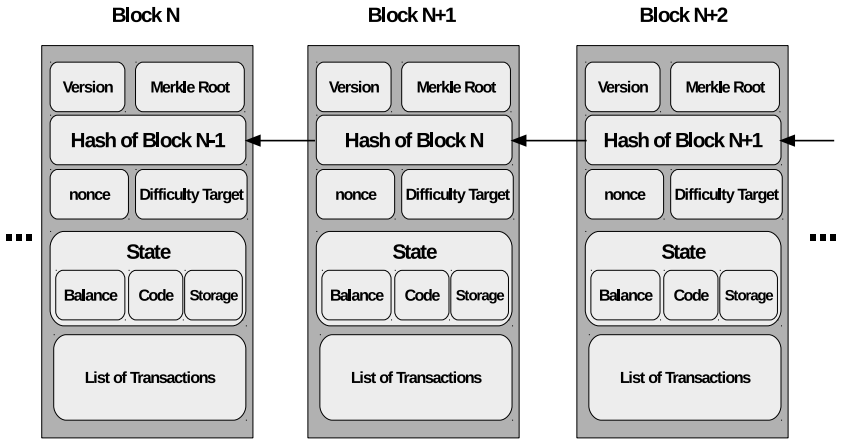

Fig. 4. Blockchain design structure showing chained blocks with header and body fields.

requires no registration or uniqueness verification when assigning and allocating an address to an IoT device. With blockchain, a centralized authority and governance, as that of the Internet Assigned Numbers Authority (IANA), is eliminated. Currently, IANA oversees the allocation of global IPv4 and IPv6 addresses, Furthermore, blockchain provides 4.3 billion addresses more than IPv6, therefore making blockchain a more scalable solution for IoT than IPv6. Lastly, it is worth noting that many IoT devices are constrained in memory and computation capacity, and therefore will be unfit to run an IPv6 stack.

Identity of Things (IDoT) and Governance. Identity and Access Management (IAM) for IoT must address a number of challenging issues in an efficiently, secure, and trustworthy manner. One primary challenge deals with ownership and identity relationships of IoT devices. Ownership of a device changes during the lifetime of the device from the manufacturer, supplier, retailer, and consumer [124,51]. The consumer ownership of an IoT device can be changed or revoked, if the device gets resold, decommissioned, or compromised. Managing of attributes and relationships of an IoT device is another challenge. Attributes of a device can include manufacturer, make, type, serial number, deployment GPS coordinates, location, etc. Apart from attributes, capabilities, and features, IoT devices have relationships. IoT relationships may include device-to-human, device-to-device, or device-to-service. An IoT device relationships can be deployed by, used by, shipped by, sold by, upgraded by, repaired by, sold by, etc.

Blockchain has the ability to solve these challenges easily, securely, and efficiently. Blockchain has been used widely for providing trustworthy and authorized identity registration, ownership tracking and monitoring of products, goods, and assets. The approaches like *TrustChain* [125] are proposed to enable trusted transactions using blockchain while maintaining the integrity of the transactions in a distributed environment. IoT devices are no exception. Blockchain can be used to register and give identity to connected IoT devices, with a set of attributes and complex relationships that can be uploaded and stored on the blockchain distributed ledger.

Blockchain also provides a trustworthy decentralized management, governance, and tracking at every point in the supply chain and lifecycle of an IoT device, as depicted in Fig. 5. The supply chain can include multiple players such as factory, vendor, supplier, distributor, shipper, installer, owner, repairer, re-installer, etc. As shown in Fig. 5, keypairs can be changed and re-issued at multiple points during the lifecycle of an IoT device. Issuance of keypairs can be done initially by the manufacturer, then by the owner, periodically after deployment.

Data Authentication and Integrity. By design, data transmitted by IoT devices connected to the blockchain network will always be cryptographically proofed and signed by the true sender that holds a unique public key and GUID, and thereby ensuring authentication and integrity of transmitted data. In addition, all transactions made to or by an IoT device are recorded on the blockchain distributed ledger and can be tracked securely.

Authentication, Authorization, and Privacy. Blockchain smart contracts have the ability to provide a de-centralized authentication rules and logic to be able to provide single and multi-party authentication to an IoT Device. Also, smart contracts can provide a more effective authorization access rules to connected IoT devices with way less complexity when compared with traditional authorization protocols like Role Based Access Management (RBAC), OAuth 2.0, OpenID, OMA DM and LWM2M. These protocols are widely used these days for IoT device authentication, authorization, and management. Moreover, data privacy can be also ensured by using smart contracts which set the access rules, conditions, and time to allow certain individual or group of users or machines to own, control, or have access to data at rest or in transit. The smart contracts can spell out also who has the right to update, upgrade, patch the IoT software or hardware, reset the IoT device, provision of new keypairs, initiate a service or repair request, change ownership, and provision or re-provision of the device.

Secure Communications. IoT application communication protocols as those of HTTP, MQTT, CoAP, or XMPP, or even protocols related to routing as those of RPL and 6LoWPAN, are not secure by design. Such protocols have to be wrapped within other security protocols such as DTLS or TLS for messaging and application protocols to provide secure communication. Similarly, for routing, IPSec is typically used to provide security for RPL and 6LoWPAN protocols. DTLS, TLS, IPSec, or even the light-weight TinyTLS protocols are heavy and complex in terms of computation and memory requirements, and complicated with a centralized management and governance of key management and distributions using the popular protocol of PKI. With blockchain, key management and distribution are totally eliminated, as each IoT device would have his own unique GUID and asymmetric key pair once installed and connected to the blockchain network. This will lead also to significant simplification of other security protocols as that of DTLS, with no need to handle and exchange PKI certificates at the handshake phase in case of DTLS or TLS (or IKE in case of IPSec) to negotiate the cipher suite parameters for encryption and hashing and to establish the master and session keys. Therefore, light-weight security protocols that would fit and stratify the requirements for the compute and memory resources of IoT devices become more feasible.

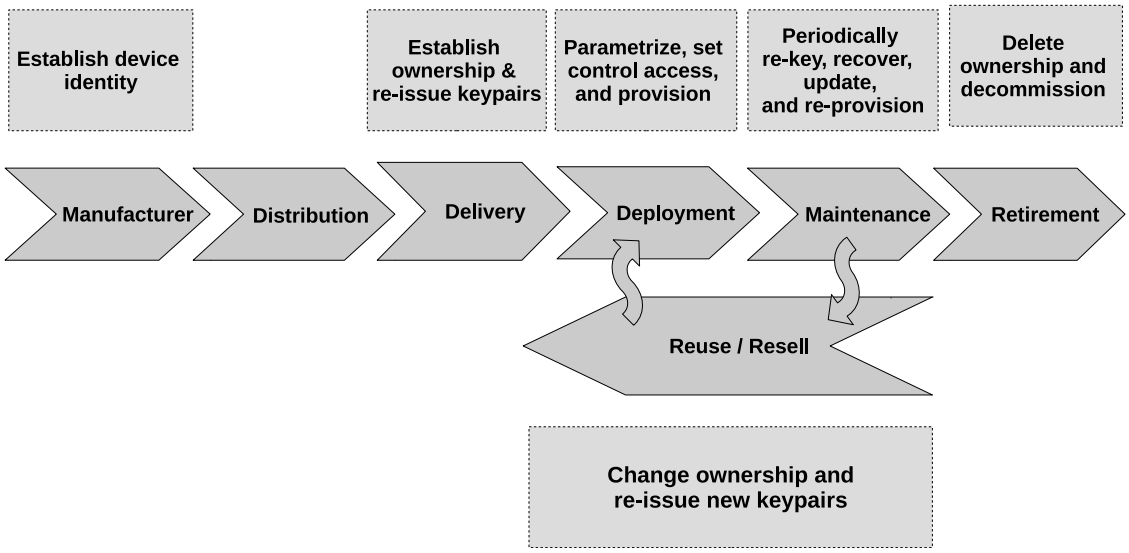

Fig. 5. IoT device lifecycle security management.

5.3. Blockchain and IoT related work

In the literature, research work on IoT security and blockchain is limited, with the majority of work being focused on leveraging blockchain technology to benefit IoT in general. The authors in [126] have categorized 18 use cases of blockchain, out of which four cases are for IoT. The four use case categories for IoT include an immutable log of events and management of access control to data [127], trading of collected IoT data [128,129], and symmetric and asymmetric key management for IoT devices [130,131]. The authors in [124] have laid out the challenges for the identity in IoT. These challenges primarily include ownership and identity relationships, authentication and authorization, governance of data and privacy. In Section 5.1, we discussed how blockchain can be a key enabler for solving these challenges.

The authors in [132] propose a blockchain-based framework for industrial IoT (or IIoT). The framework enables IIoT devices to communicate with the cloud as well as the blockchain network. Each IIoT device is equipped with single-board computer (SBC) having control and communication interface capabilities for both cloud and the Ethereum blockchain. IIoT devices are designed to send data to the cloud for storage and analysis, and send/receive transactions to other devices on the blockchain network, and also to trigger executions of smart contracts. As a proof of concept, the authors implement a simple platform using Arduino Uno board and Ethereum smart contracts and describe briefly how the platform can be used for machine maintenance and smart diagnostics.

The applications of blockchain smart contracts to IoT are reviewed by Christidis et al. [133]. The authors describe how smart contracts of blockchain can facilitate and support the autonomous workflow and the sharing of services among IoT devices, as proposed in [134]. Moreover, the authors argue how IoT can benefit from blockchain networks in aspects related to billing, e-trading, shipping and supply chain management. Furthermore, they describe a scenario where blockchain can facilitate the buying and selling of energy automatically among IoT device like smart meters. Smart contracts can be used to set user-defined criteria for energy trading. The authors also describe another scenario for asset tracking of container shipment using smart contracts and IoT.

6. Open challenges and future research directions

This section discusses the challenges being envisaged for effective implementation of security for IoT devices.

6.1. Resource limitations

The resource-constrained architecture of IoT has been a main hindrance in defining a robust security mechanism. In contrast to the conventional paradigms, the cryptographic algorithms have to be limited to work within these constraints. With any broadcasts, or multicasts required for exchange of keys or certificates, the storage as well as the energy requirements need to be coped with in order to provide a successful implementation of security and communication protocols for IoT. This entails re-designing of these protocols to be lightweight and energy efficient despite requiring complex computations along with improvement of energy harvesting techniques [135].

6.2. Heterogeneous devices

As with heterogeneous devices ranging from small low power devices with sensors to high-end servers, a multi-layer security framework needs to be implemented. The framework should initially adapt itself to existing resources, make decisions regarding selection of security mechanisms at IoT layers before any services are provided to end-users. Such a dynamically adaptable security framework requires intelligence, which is subject to the standardization of resources to be deployed in IoT architectures.

6.3. Interoperability of security protocols

For standardizing a global security mechanism for IoT, the protocols implemented at different layers need to interoperate by providing conversion mechanisms. Within the global mechanism, an effective combination of security standards at each layer can then be defined through consideration of architectural constraints.

6.4. Single points of failure

With the heterogeneous networks, architectures, and protocols, the IoT paradigm becomes more vulnerable to single points of failure than any other paradigm. A significant amount of research work yet needs to be carried out to ensure adequate availability of IoT elements, especially for mission-critical applications. It would require mechanisms and standards to introduce redundancy while keeping in view the trade-off between the costs and the reliability of the entire infrastructure.

6.5. Hardware/firmware vulnerabilities

With the low-cost and low-power devices becoming ubiquitous, the IoT architecture may become more exposed to hardware vulnerabilities. It is not just the physical malfunctioning, instead, implementation of security algorithms in the hardware, routing and packet processing mechanisms also need to be verified before deployment in IoT. Any vulnerabilities exploited after deployment become difficult to detect and alleviate. A standard verification protocol is therefore an essential requisite for harnessing the IoT security.

6.6. Trusted updates and management

One of the key open issues for future research is providing scalable and trusted management and updates of software to millions of IoT devices. Moreover, the issues related to secure and trusted governance of IoT device ownership, supply chain, and data privacy are open research problems that need to be addressed by the research community to foster a wide and massive scale adoption for IoT. The blockchain technology can be an enabler for such IoT security solutions. However, the blockchain technology in itself poses research challenges to be tackled with regards to its scalability, efficiency, arbitration/regulations, and key collision.

6.7. Blockchain vulnerabilities

Despite providing robust approaches for securing IoT, the blockchain systems are also vulnerable [136]. The consensus mechanism depending upon the miner's hashing power can be compromised, thereby allowing the attacker to host the blockchain. Similarly, the private keys with limited randomness can be exploited to compromise the blockchain accounts. Effective mechanisms yet need to be defined to ensure the privacy of transactions and avoid race attacks which may result in double spending during transactions.

7. Conclusion

Todays IoT devices are insecure and incapable of defending themselves. This is due to mainly the constrained resources in IoT devices, immature standards, and the absence of secure hardware and software design, development, and deployment. The efforts of defining a robust global mechanism for securing the IoT layers are also being hampered due to diversity of resources in IoT. In this paper, we survey and review main IoT security issues. We categorize these issues depending upon the high-level, intermediate-level, and low-level IoT layers. We discuss succinctly the mechanisms suggested in the literature for leveraging IoT security at different levels. A parametric analysis of attacks in IoT and their possible solutions is also provided. We consider the attack implications and map them to possible solutions proposed in the literature. We also discuss how the blockchain can be used to address and solve some of the most pertaining IoT security problems. The paper also outlines and identifies future and open research issues and challenges that need to be addressed by the research community in order to provide reliable, efficient, and scalable IoT security solutions.

References

[1] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805.
[2] D. Giusto, A. Iera, G. Morabito, L. Atzori, The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications, Springer Publishing Company, Incorporated, 2014.
[3] B. Heater, Lenovo shows off a pair of intel-powered smart shoes, 2016. URL https://techcrunch.com/2016/06/09/lenovo-smart-shoes/.
[4] M. Rouse, I. Wigmore, Internet of things, 2016. URL http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT..
[5] A.A. Khan, M.H. Rehmani, A. Rachedi, Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions, IEEE Wirel. Commun. 24 (3) (2017) 17–25. http://dx.doi.org/10.1109/MWC.2017.1600404.
[6] F. Akhtar, M.H. Rehmani, M. Reisslein, White space: Definitional perspectives and their role in exploiting spectrum opportunities, Telecommun. Policy 40 (4) (2016) 319–331. http://dx.doi.org/10.1016/j.telpol.2016.01.003.
[7] F.A. Alaba, M. Othman, I.A.T. Hashem, F. Alotaibi, Internet of things security: A survey, J. Netw. Comput. Appl. 88 (Suppl. C) (2017) 10–28. http://dx.doi.org/10.1016/j.jnca.2017.04.002.
[8] J. Granjal, E. Monteiro, J.S. Silva, Security for the internet of things: A Survey of existing protocols and open research issues, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1294–1312. http://dx.doi.org/10.1109/COMST.2015.2388550.
[9] R. Roman, C. Alcaraz, J. Lopez, N. Sklavos, Key management systems for sensor networks in the context of the internet of things, Comput. Electr. Eng. 37 (2) (2011) 147–159. Modern Trends in Applied Security: Architectures, Implementations and Applications.
[10] J. Granjal, R. Silva, E. Monteiro, J.S. Silva, F. Boavida, Why is IPSec a viable option for wireless sensor networks, in: 2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2008, pp. 802–807. http://dx.doi.org/10.1109/MAHSS.2008.4660130.
[11] S. Cirani, G. Ferrari, L. Veltri, Enforcing security mechanisms in the IP-based internet of things: An algorithmic overview, Algorithms 6 (2) (2013) 197–226. http://dx.doi.org/10.3390/a6020197.
[12] I. Butun, S.D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, IEEE Commun. Surv. Tutor. 16 (1) (2014) 266–282. http://dx.doi.org/10.1109/SURV.2013.050113.00191.
[13] A. Abduvaliyev, A.-S.K. Pathan, J. Zhou, R. Roman, W.-C. Wong, On the vital areas of intrusion detection systems in wireless sensor networks, IEEE Commun. Surv. Tutor. 15 (3) (2013) 1223–1237. http://dx.doi.org/10.1109/SURV.2012.121912.00006.
[14] R. Mitchell, I.-R. Chen, Review: a survey of intrusion detection in wireless network applications, Comput. Commun. 42 (2014) 1–23. http://dx.doi.org/10.1016/j.comcom.2014.01.012.
[15] S. Yi, Z. Qin, Q. Li, Security and privacy issues of fog computing: A survey, in: Wireless Algorithms, Systems, and Applications the 10th International Conference on, 2015, pp. 1–10.
[16] Y. Wang, T. Uehara, R. Sasaki, Fog computing: Issues and challenges in security and forensics, in: 2015 IEEE 39th Annual Computer Software and Applications Conference, vol. 3, 2015, pp. 53–59. http://dx.doi.org/10.1109/COMPSAC.2015.173.
[17] S. Sicari, A. Rizzardi, L. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Comput. Netw. 76 (Suppl. C) (2015) 146–164. http://dx.doi.org/10.1016/j.comnet.2014.11.008.
[18] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges, Future Gener. Comput. Syst. (2016). http://dx.doi.org/10.1016/j.future.2016.11.009.
[19] V. Oleshchuk, Internet of things and privacy preserving technologies, in: 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009, pp. 336–340. http://dx.doi.org/10.1109/WIRELESSVITAE.2009.5172470.
[20] J. Zhou, Z. Cao, X. Dong, A.V. Vasilakos, Security and privacy for cloud-based IoT: Challenges, IEEE Commun. Mag. 55 (1) (2017) 26–33. http://dx.doi.org/10.1109/MCOM.2017.1600363CM.
[21] Z.K. Zhang, M.C.Y. Cho, C.W. Wang, C.W. Hsu, C.K. Chen, S. Shieh, IoT security: Ongoing challenges and research opportunities, in: 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, 2014, pp. 230–234. http://dx.doi.org/10.1109/SOCA.2014.58.
[22] IoT-A, Internet of Things–Architecture IoT-A Deliverable D1.5 –Final architectural reference model for the IoT v3.0, 2013. URL http://iotforum.org/wp-content/uploads/2014/09/D1.5-20130715-VERYFINAL.pdf.
[23] OWASP, Top IoT Vulnerabilities, 2016. URL https://www.owasp.org/index.php/Top_IoT_Vulnerabilities.
[24] IEEE, IeEEE Standard for Local and metropolitan networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), 2012. URL https://standards.ieee.org/findstds/standard/802.15.4-2011.html.
[25] T. Winter, P. Thubert, A. Brandt, J.W. Hui, R. Kelsey, Rfc 6550 - rpl: ipv6 routing protocol for low-power and lossy networks, 2012. URL https://tools.ietf.org/html/rfc6550.
[26] J. Postel, User datagram protocol, 1980. URL https://tools.ietf.org/html/rfc768.
[27] J.W. Hui, P. Thubert, Compression format for IPv6 datagrams over IEEE 802.15.4-based networks, 2011. URL https://tools.ietf.org/html/rfc6282.
[28] A. Conta, S. Deering, M. Gupta, Internet control message protocol (ICMPv6) for the internet protocol version 6 (IPv6) specification, 2006. URL https://tools.ietf.org/html/rfc4443.

[29] Z. Shelby, K. Hartke, C. Bormann, The constrained application protocol (CoAP), 2014. URL https://tools.ietf.org/html/rfc7252.

[30] W. Xu, W. Trappe, Y. Zhang, T. Wood, 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, MobiHoc '05, ACM, New York, NY, USA, 2005, pp. 46–57. http://dx.doi.org/10.1145/1062689.1062697.

[31] G. Noubir, G. Lin, Low-power DoS attacks in data wireless LANs and countermeasures, SIGMOBILE Mob. Comput. Commun. Rev. 7 (3) (2003) 29–30.

[32] S.H. Chae, W. Choi, J.H. Lee, T.Q.S. Quek, Enhanced secrecy in stochastic wireless networks: Artificial noise with secrecy protected zone, Trans. Info. for. Sec. 9 (10) (2014) 1617–1628. http://dx.doi.org/10.1109/TIFS.2014.2341453.

[33] Y.-W.P. Hong, P.-C. Lan, C.-C.J. Kuo, Enhancing physical-layer secrecy in multiantenna wireless systems: An overview of signal processing approaches, IEEE Signal Process. Mag. 30 (5) (2013) 29–40.

[34] L. Xiao, L.J. Greenstein, N.B. Mandayam, W. Trappe, Channel-Based detection of sybil attacks in wireless networks, IEEE Transa. Inf. Forensics Secur. 4 (3) (2009) 492–503.

[35] Y. Chen, W. Trappe, R.P. Martin, Detecting and localizing wireless spoofing attacks, in: 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2017, pp. 193–202.

[36] T. Bhattasali, R. Chaki, A survey of recent intrusion detection systems for wireless sensor network, in: D.C. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, D. Nagamalai (Eds.), Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, July 15–17, 2011, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 268–280.

[37] H. Kim, Protection against packet fragmentation attacks at 6LoWPAN adaptation layer, in: 2008 International Conference on Convergence and Hybrid Information Technology, 2008, pp. 796–801. http://dx.doi.org/10.1109/ICHIT.2008.261.

[38] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, K. Wehrle, 6LoWPAN Fragmentation attacks and mitigation mechanisms, in: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '13, ACM, New York, NY, USA, 2013, pp. 55–66. http://dx.doi.org/10.1145/2462096.2462107.

[39] R. Riaz, K.-H. Kim, H.F. Ahmed, Security analysis survey and framework design for IP connected LoWPANs, in: 2009 International Symposium on Autonomous Decentralized Systems, 2009, pp. 1–6. http://dx.doi.org/10.1109/ISADS.2009.5207373.

[40] A. Dvir, T. Holczer, L. Buttyan, VeRA - version number and rank authentication in RPL, in: 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, 2011, pp. 709–714. http://dx.doi.org/10.1109/MASS.2011.76.

[41] K. Weekly, K. Pister, Evaluating sinkhole defense techniques in RPL networks, in: Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP), ICNP '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 1–6. http://dx.doi.org/10.1109/ICNP.2012.6459948.

[42] F. Ahmed, Y.-B. Ko, Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks, Secur. Commun. Netw. 9 (18) (2016) 5143–5154 SCN-16-0443.R1.

[43] A.A. Pirzada, C. McDonald, Circumventing sinkholes and wormholes in wireless sensor networks, in: International Workshop on Wireless Ad-Hoc Networks, 2005.

[44] W. Wang, J. Kong, B. Bhargava, M. Gerla, Visualisation of wormholes in underwater sensor networks: A distributed approach, Int. J. Secur. Netw. 3 (1) (2008) 10–23.

[45] M. Wazid, A.K. Das, S. Kumari, M.K. Khan, Design of sinkhole node detection mechanism for hierarchical wireless sensor networks, Sec. Commun. Netw. 9 (17) (2016) 4596–4614. http://dx.doi.org/10.1002/sec.1652.

[46] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the internet of things, IEEE Internet Things J. 1 (5) (2014) 372–383. http://dx.doi.org/10.1109/JIOT.2014.2344013.

[47] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, B.Y. Zhao, Social turing tests: Crowdsourcing sybil detection, in: Symposium on Network and Distributed System Security, NDSS, 2013.

[48] J. Granjal, E. Monteiro, J.S. Silva, Network-layer security for the Internet of Things using TinyOS and BLIP, Int. J. Commun. Syst. 27 (10) (2014) 1938–1963. http://dx.doi.org/10.1002/dac.2444.

[49] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, U. Roedig, Securing communication in 6LoWPAN with compressed IPsec, in: 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011, pp. 1–8. http://dx.doi.org/10.1109/DCOSS.2011.5982177.

[50] J. Granjal, E. Monteiro, J.S. Silva, Enabling network-layer security on IPv6 wireless sensor networks, in: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, 2010, pp. 1–6. http://dx.doi.org/10.1109/GLOCOM.2010.5684293.

[51] P.N. Mahalle, B. Anggorojati, N.R. Prasad, R. Prasad, Identity authentication and capability based access control (iacac) for the internet of things, J. Cyber Secur. Mobility 1 (4) (2013) 309–348.

[52] D.U. Sinthan, M.-S. Balamurugan, Identity authentication and capability based access control (IACAC) for the Internet of Things, J. Cyber Secur. Mob. 1 (4) (2013) 309–348.

[53] M. Brachmann, O. Garcia-Morchon, M. Kirsche, Security for practical CoAP applications: Issues and solution approaches, in: 10th GI/ITG KuVS Fachgespraech Sensornetze (FGSN 2011), 2011.

[54] J. Granjal, E. Monteiro, J.S. Silva, End-to-end transport-layer security for internet-integrated sensing applications with mutual and delegated ecc public-key authentication, in: 2013 IFIP Networking Conference, 2013, pp. 1–9.

[55] G. Peretti, V. Lakkundi, M. Zorzi, BlinkToSCoAP: An end-to-end security framework for the Internet of Things, in: 2015 7th International Conference on Communication Systems and Networks (COMSNETS), 2015, pp. 1–6. http://dx.doi.org/10.1109/COMSNETS.2015.7098708.

[56] S. Raza, T. Voigt, V. Jutvik, Lightweight IKEv2: a key management solution for both the compressed IPsec and the IEEE 802.15. 4 security, in: Proceedings of the IETF Workshop on Smart Object Security, vol. 23, 2012.

[57] N. Park, N. Kang, Mutual authentication scheme in secure internet of things technology for comfortable lifestyle, Sensors 6 (1) (2016) 20–20.

[58] M.H. Ibrahim, Octopus: An edge-fog mutual authentication scheme, Internat. J. Netw. Secur. 18 (6) (2016) 1089–1101.

[59] M. Henze, B. Wolters, R. Matzutt, T. Zimmermann, K. Wehrle, Distributed configuration, authorization and management in the cloud-based internet of things, in: 2017 IEEE Trustcom/BigDataSE/ICESS, 2017, pp. 185–192. http://dx.doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.236.

[60] M. Brachmann, S.L. Keoh, O.G. Morchon, S.S. Kumar, End-to-end transport security in the IP-based Internet of Things, in: 2012 21st International Conference on Computer Communications and Networks, ICCCN, 2012, pp. 1–5. http://dx.doi.org/10.1109/ICCCN.2012.6289292.

[61] J. Granjal, E. Monteiro, J.S. Silva, Application-layer security for the WoT: extending CoAP to support end-to-end message security for internet-integrated sensing applications, in: International Conference on Wired/Wireless Internet Communication, Springer Berlin Heidelberg, 2013, pp. 140–153.

[62] M. Sethi, J. Arkko, A. Kernen, End-to-end security for sleepy smart object networks, in: 37th Annual IEEE Conference on Local Computer Networks - Workshops, 2012, pp. 964–972. http://dx.doi.org/10.1109/LCNW.2012.6424089.

[63] D. Conzon, T. Bolognesi, P. Brizzi, A. Lotito, R. Tomasi, M.A. Spirito, The VIRTUS middleware: An XMPP based architecture for secure IoT communications, in: 2012 21st International Conference on Computer Communications and Networks, ICCCN, 2012, pp. 1–6. http://dx.doi.org/10.1109/ICCCN.2012.6289309.

[64] C.H. Liu, B. Yang, T. Liu, Efficient naming, addressing and profile services in Internet-of-Things sensory environments, Ad Hoc Netw. 18 (Suppl. C) (2014) 85–101. http://dx.doi.org/10.1016/j.adhoc.2013.02.008.

[65] M. Young, R. Boutaba, Overcoming adversaries in sensor networks: A survey of theoretical models and algorithmic approaches for tolerating malicious interference, IEEE Commun. Surv. Tutor. 13 (4) (2011) 617–641. http://dx.doi.org/10.1109/SURV.2011.041311.00156.

[66] W. Xu, T. Wood, W. Trappe, Y. Zhang, Channel surfing and spatial retreats: Defenses against wireless denial of service, in: Proceedings of the 3rd ACM Workshop on Wireless Security, WiSe '04, ACM, New York, NY, USA, 2004, pp. 80–89. http://dx.doi.org/10.1145/1023646.1023661.

[67] T. Pecorella, L. Brilli, L. Muchhi, The role of physical layer security in IoT: A novel perspective, Information 7 (3) (2016).

[68] M. Demirbas, Y. Song, An RSSI-based scheme for sybil attack detection in wireless sensor networks, in: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, WOWMOM '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 564–570. http://dx.doi.org/10.1109/WOWMOM.2006.27.

[69] Q. Li, W. Trappe, Light-weight detection of spoofing attacks in wireless networks, in: 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2006, pp. 845–851.

[70] L. Xiao, L. Greenstein, N. Mandayam, W. Trappe, Fingerprints in the ether: Using the physical layer for wireless authentication, in: 2007 IEEE International Conference on Communications, 2007, pp. 4646–4651. http://dx.doi.org/10.1109/ICC.2007.767.

[71] R. Harkanson, Y. Kim, Applications of elliptic curve cryptography: A light introduction to elliptic curves and a survey of their applications, in: Proceedings of the 12th Annual Conference on Cyber and Information Security Research, CISRC '17, ACM, New York, NY, USA, 2017, pp. 6:1–6:7. http://dx.doi.org/10.1145/3064814.3064818.

[72] D. Eastlake, P.E. Jones, RFC 3174 - US Secure Hash Algorithm 1 (SHA1), 2001. URL https://tools.ietf.org/html/rfc3174.

[73] H. Krawczyk, M. Bellare, R. Canetti, HMAC: keyed-hashing for message authentication, 1997. URL https://tools.ietf.org/rfc/rfc2104.txt.

[74] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126. http://dx.doi.org/10.1145/359340.359342.

[75] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, M. Chai, The impact of rank attack on network topology of routing protocol for low-power and lossy networks, IEEE Sens. J. 13 (10) (2013) 3685–3692. http://dx.doi.org/10.1109/JSEN.2013.2266399.

[76] Y.-C. Hu, A. Perrig, D.B. Johnson, Ariadne: A secure on-demand routing protocol for ad hoc networks, Wirel. Netw. 11 (1) (2005) 21–38.

[77] I. Krontiris, T. Dimitriou, T. Giannetsos, M. Mpasoukos, Intrusion detection of sinkhole attacks in wireless sensor networks, in: M. Kutyłowski, J. Cichoń, P. Kubiak (Eds.), Algorithmic Aspects of Wireless Sensor Networks: Third International Workshop, ALGOSENSORS 2007, Wroclaw, Poland, July 14, 2007, Revised Selected Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 150–161.

[78] I. Raju, P. Parwekar, Detection of sinkhole attack in wireless sensor network, in: S.C. Satapathy, K.S. Raju, J.K. Mandal, V. Bhateja (Eds.), Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 3, Springer India, New Delhi, 2016, pp. 629–636.

[79] E.C.H. Ngai, J. Liu, M.R. Lyu, On the intruder detection for sinkhole attack in wireless sensor networks, in: 2006 IEEE International Conference on Communications, vol. 8, 2006, pp. 3383–3389. http://dx.doi.org/10.1109/ICC.2006.255595.

[80] R. Poovendran, L. Lazos, A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks, Wirel. Netw. 13 (1) (2007) 27–59. http://dx.doi.org/10.1007/s11276-006-3723-x.

[81] S.A. Salehi, M.A. Razzaque, P. Naraei, A. Farrokhtala, Detection of sinkhole attack in wireless sensor networks, in: 2013 IEEE International Conference on Space Science and Communication (IconSpace), 2013, pp. 361–365. http://dx.doi.org/10.1109/IconSpace.2013.6599496.

[82] C. Tumrongwittayapak, R. Varakulsiripunth, Detecting Sinkhole attacks in wireless sensor networks, in: 2009 ICCAS-SICE, 2009, pp. 1966–1971.

[83] J. Jang, T. Kwon, J. Song, A time-based key management protocol for wireless sensor networks, in: Proceedings of the 3rd International Conference on Information Security Practice and Experience, ISPEC'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 314–328.

[84] S. Sharmila, G. Umamaheswari, Detection of sinkhole attack in wireless sensor networks using message digest algorithms, in: 2011 International Conference on Process Automation, Control and Computing, 2011, pp. 1–6. http://dx.doi.org/10.1109/PACC.2011.5978973.

[85] H. Yu, M. Kaminsky, P.B. Gibbons, A. Flaxman, SybilGuard: defending against sybil attacks via social networks, SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 267–278. http://dx.doi.org/10.1145/1151659.1159945.

[86] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, A. Panconesi, SoK: the evolution of sybil defense via social networks, in: Proceedings of the 2013 IEEE Symposium on Security and Privacy, SP '13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 382–396. http://dx.doi.org/10.1109/SP.2013.33.

[87] Q. Cao, X. Yang, SeybilFence: Improving Social-Graph-Based Sybil Defenses with User Negative Feedback. Tech. Rep., Duke, Duke University, USA, 2012 URL https://users.cs.duke.edu/~qiangcao/.

[88] A. Mohaisen, N. Hopper, Y. Kim, Keep your friends close: incorporating trust into social network-based sybil defenses, in: 2011 Proceedings IEEE INFOCOM, 2011, pp. 1943–1951. http://dx.doi.org/10.1109/INFCOM.2011.5934998.

[89] D. Quercia, S. Hailes, Sybil attacks against mobile users: Friends and foes to the rescue, in: Proceedings of the 29th Conference on Information Communications, INFOCOM'10, IEEE Press, Piscataway, NJ, USA, 2010, pp. 336–340 URL http://dl.acm.org/citation.cfm?id=1833515.1833583.

[90] S. Kent, RFC 4302 - ip authentication header, 2005. URL https://tools.ietf.org/html/rfc4302.

[91] S. Kent, RFC 4303 - IP Encapsulating Security Payload (ESP), 2005. URL https://tools.ietf.org/html/rfc4303.

[92] S. Raza, T. Chung, S. Duquennoy, D. Yazar, T. Voigt, U. Roedig, Securing Internet of Things with Lightweight IPsec, SICS, Lancaster University, UK, 2011. URL http://soda.swedishict.se/4052/2/reportRevised.pdf.

[93] S. Raza, S. Duquennoy, J. Hglund, U. Roedig, T. Voigt, Secure communication for the Internet of Things–a comparison of link-layer security and IPsec for 6LoWPAN, Secur. Commun. Netw. 7 (12) (2014) 2654–2668. http://dx.doi.org/10.1002/sec.406.

[94] J.W. Hui, P. Thubert, Compression Format for IPv6 Datagrams in 6LoWPAN Networks draft-ietf-6lowpan-hc-13, 2010. URL https://tools.ietf.org/html/draft-ietf-6lowpan-hc-13.

[95] G. Montenegro, N. Kushalnagar, J.W. Hui, D.E. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, 2007. URL https://tools.ietf.org/html/rfc4944.

[96] T. Kothmayr, C. Schmitt, W. Hu, M. Brnig, G. Carle, 37th Annual IEEE Conference on Local Computer Networks - Workshops, 2012, pp. 956–963. http://dx.doi.org/10.1109/LCNW.2012.6424088.

[97] T. Kothmayr, C. Schmitt, W. Hu, M. Brnig, G. Carle, {DTLS} based security and two-way authentication for the Internet of Things, Ad Hoc Netw. 11 (8) (2013) 2710–2723. http://dx.doi.org/10.1016/j.adhoc.2013.05.003.

[98] S.L. Kinney, Trusted Platform Module Basics: Using TPM in Embedded Systems, Newnes, Newton, MA, USA, 2006.

[99] X. Huang, Y. Xiang, E. Bertino, J. Zhou, L. Xu, Robust multi-factor authentication for fragile communications, IEEE Trans. Dependable Secure Comput. 11 (6) (2014) 568–581. http://dx.doi.org/10.1109/TDSC.2013.2297110.

[100] J.M. Bohli, A. Skarmeta, M.V. Moreno, D. Garca, P. Langendörfer, SMARTIE project: Secure IoT data management for smart cities, in: 2015 International Conference on Recent Advances in Internet of Things (RIoT), 2015, pp. 1–6. http://dx.doi.org/10.1109/RIOT.2015.7104906.

[101] I. Stojmenovic, S. Wen, X. Huang, H. Luan, An overview of fog computing and its security issues, Concurr. Comput. Pract. Exp. 28 (10) (2016) 2991–3005. http://dx.doi.org/10.1002/cpe.3485.

[102] A.A. Chavan, M.K. Nighot, Secure CoAP using enhanced DTLS for Internet of Things, Internat. J. Innovative Res. Comput. Commun. Eng. 2 (12) (2014) 7601–7608.

[103] S. Raza, D. Trabalza, T. Voigt, 6LoWPAN compressed DTLS for CoAP, in: 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems, 2012, pp. 287–289. http://dx.doi.org/10.1109/DCOSS.2012.55.

[104] H.C. Phls, V. Angelakis, S. Suppan, K. Fischer, G. Oikonomou, E.Z. Tragos, R.D. Rodriguez, T. Mouroutis, RERUM: Building a reliable IoT upon privacy- and security- enabled smart objects, in: 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2014, pp. 122–127. http://dx.doi.org/10.1109/WCNCW.2014.6934872.

[105] BUTLER-Consortium, BUTLER smartlife –uBiquitous, secUre inTernet-of-things with Location and contExt-awaReness, 2014. URL http://cordis.europa.eu/docs/projects/cnect/1/287901/080/deliverables/001-287901BUTLERD25.pdf.

[106] R. Hummen, H. Wirtz, J.H. Ziegeldorf, J. Hiller, K. Wehrle, Tailoring end-to-end IP security protocols to the Internet of Things, in: 2013 21st IEEE International Conference on Network Protocols (ICNP), 2013, pp. 1–10. http://dx.doi.org/10.1109/ICNP.2013.6733571.

[107] S. Prez, J.A. Martnez, A.F. Skarmeta, M. Mateus, B. Almeida, P. Mal, ARMOUR: Large-scale experiments for IoT security trust, in: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016, pp. 553–558. http://dx.doi.org/10.1109/WF-IoT.2016.7845504.

[108] M. Brachmann, O. Garcia-Morchon, S.-L. Keoh, S.S. Kumar, Security considerations around end-to-end security in the IP-based Internet of Things, in: 2012 Workshop on Smart Object Security, in Conjunction with IETF83, 2012, pp. 1–3.

[109] A. Gmez-Goiri, P. Ordua, J. Diego, D.L. de Ipiña, Otsopack: Lightweight semantic framework for interoperable ambient intelligence applications, Comput. Hum. Behav. 30 (Suppl. C) (2014) 460–467. http://dx.doi.org/10.1016/j.chb.2013.06.022.

[110] OneM2M, Security solutions –OneM2M Technical Specification, 2017. URL http://onem2m.org/technical/latest-drafts.

[111] H.G.C. Ferreira, R.T. de Sousa, F.E.G. de Deus, E.D. Canedo, Proposal of a secure, deployable and transparent middleware for Internet of Things, in: 2014 9th Iberian Conference on Information Systems and Technologies, CISTI, 2014, pp. 1–4. http://dx.doi.org/10.1109/CISTI.2014.6877069.

[112] A.M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Crypto-Currencies, first ed., O'Reilly Media, Inc., 2014.

[113] The-Bitcoin-Foundation, How does Bitcoin work?, 2014. URL https://bitcoin.org/en/how-it-works.

[114] BitInfoCharts, Block - Bitcoin Wiki, 2016. URL https://en.bitcoin.it/wiki/Block.

[115] EtherScan, Ethereum Average BlockTime Chart, 2016. URL https://etherscan.io/chart/blocktime.

[116] Linux-Foundation, Blockchain technologies for business, 2017. URL https://www.hyperledger.org/.

[117] C. Kuhlman, What is eris? 2016 Edition, 2016. URL https://monax.io/2016/04/03/wtf-is-eris/.

[118] Stellar, Stellar network overview, 2014. URL https://www.stellar.org/developers/guides/get-started/..

[119] Ripple, Ripple network, 2013. URL https://ripple.com/network.

[120] All-In-Bits, Introduction to tendermint, 2017. URL https://tendermint.com/intro.

[121] J. Mattila, The blockchain phenomenon: The disruptive potential of distributed consensu architectures, ETLA working papers: Elinkeinoelämän Tutkimuslaitos, Research Institute of the Finnish Economy, 2016 URL https://books.google.com.pk/books?id=StNQnQAACAAJ.

[122] EconoTimes, Safeshare releases first blockchain insurance solution for sharing economy, 2016. URL https://www.econotimes.com/SafeShare-Releases-First-Blockchain-Insurance-Solution-For-Sharing-Economy-181326.

[123] IBM, IBM blockchain based on hyperledger fabric from the linux foundation, 2017. URL https://www.ibm.com/blockchain/hyperledger.html.

[124] I. Friese, J. Heuer, N. Kong, Challenges from the Identities of Things: Introduction of the Identities of Things discussion group within Kantara initiative, in: 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014, pp. 1–4. http://dx.doi.org/10.1109/WF-IoT.2014.6803106.

[125] P. Otte, M. de Vos, J. Pouwelse, TrustChain: A Sybil-resistant scalable blockchain, Future Gener. Comput. Syst. (2017). http://dx.doi.org/10.1016/j.future.2017.08.048.

[126] M. Conoscenti, A. Vetro, J.C.D. Martin, Blockchain for the Internet of Things: A systematic literature Review, in: The 3rd International Symposium on Internet of Things: Systems, Management, and Security, IOTSMS-2016, 2016.
[127] G. Zyskind, O. Nathan, A. Pentland, Enigma: decentralized computation platform with guaranteed privacy, 2015. URL http://enigma.media.mit.edu/enigma~full.pdf.
[128] Y. Zhang, J. Wen, An IoT electric business model based on the protocol of bitcoin, in: 2015 18th International Conference on Intelligence in Next Generation Networks, 2015, pp. 184–191. http://dx.doi.org/10.1109/ICIN.2015.7073830.
[129] D. Wörner, T. von Bomhard, When your sensor earns money: Exchanging data for cash with bitcoin, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, UbiComp '14 Adjunct, ACM, New York, NY, USA, 2014, pp. 295–298. http://dx.doi.org/10.1145/2638728.2638786.
[130] L. Axon, Privacy-awareness in Blockchain-based PKI, Tech. Rep. 2015. URL https://ora.ox.ac.uk/objects/uuid:f8377b69-599b-4cae-8df0-f0cded53e63b/datastreams/ATTACHMENT01.
[131] C. Fromknecht, D. Velicanu, S. Yakoubov, CertCoin: A namecoin based decentralized authentication system, 2014. URL https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf.
[132] A. Bahga, V.K. Madisetti, Blockchain platform for industrial Internet of Things, Tech. Rep. 2016. URL http://file.scirp.org/pdf/JSEA_2016102814012798.pdf.
[133] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the Internet of Things, IEEE Access 4 (2016) 2292–2303. http://dx.doi.org/10.1109/ACCESS.2016.2566339.
[134] V. Pureswaran, P. Brody, Device Democracy - Saving the future of the Internet of Things, IBM, 2014. URL http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=GBE03620USEN.
[135] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V.C.M. Leung, Y.L. Guan, Wireless energy harvesting for the Internet of Things, IEEE Commun. Mag. 53 (6) (2015) 102–108. http://dx.doi.org/10.1109/MCOM.2015.7120024.
[136] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Gener. Comput. Syst. (2017). http://dx.doi.org/10.1016/j.future.2017.08.020.

Minhaj Ahmad Khan completed his M.S. and Ph.D. degrees in Computer Science from University of Versailles, France. He also holds a post-doc from University of Bordeaux-I, France. He is currently working as Associate Professor at Bahauddin Zakariya University, Multan. His research interests include Computer Networks and High Performance Computing. He has several publications in prestigious journals and conferences on these topics.

Prof. Khaled Salah is a full professor at the Department of Electrical and Computer Engineering, Khalifa University, UAE. He received the B.S. degree in Computer Engineering with a minor in Computer Science from Iowa State University, USA, in 1990, the M.S. degree in Computer Systems Engineering from Illinois Institute of Technology, USA, in 1994, and the Ph.D. degree in Computer Science from the same institution in 2000. Khaled has over 145 publications and 3 patents. He has been the recipient of several prestigious awards for his outstanding research. He is a senior member of IEEE, and serves on the Editorial Boards of many WOS-listed journals including IET Communications, IET Networks, Elsevier's JNCA, Wiley's SCN, Wiley's IJNM, J.UCS, and AJSE.