Indian Journal of Software Engineering and Project Management (IJSEPM)
ISSN: 2582-8339(Online), Volume-1, Issue-1, January 2021

A Review on IoT Based Operating System and it's Security Challenges

Waheed Javed, Gulnaz Parveen, Adnan Abid

Abstract — ***Today Embedded devices connecting all kinds of physical objects to the internet such as smartphones, light bulbs, medical devices, smart devices, smart cars, and even smart cities it's all beholden of IoT. IoT connects a wide range of services to people. In coming years, it expects that billions of items will be implemented. Currently, there has been a lot of effort to map the operating systems For IoT devices, because IoT applications are not properly run on the Windows / Unix for real-time applications. So these operating systems cannot meet the necessities of the IoT applications. A trustworthy forum for the Internet of Things (IoT) expansion is thus important as a new architecture. The data that is gathered from IoT devices are mostly unsaturated and noisy, on the other hand, we needed efficient and accurate result, so more computation power required for analysis And an optimized authentication method was also needed for lightweight devices such as IoT, where very few computing power, resource constraints, limited memory, and limited battery life are needed. In this examination paper, an investigation is exhibited dependent on many open-source operating systems for IoTs and portray the qualities of some well-known operating systems and can best choice about working framework for application explicit prerequisites, and I will discuss in the last section possible current security challenges in IoT operating system.***

Keywords—Operating Systems; Mbed, Contiki, Internet of Things, TinyOS

I. INTRODUCTION

IoT operating systems enable users to execute simple programming functions inside an interface linked to the internet.[1] IoT operating systems are integrated into the IoT devices and connect to a larger device network. These operating systems have comparable capabilities to those of a device by providing memory and data storage processing power.[3] All applications operating on the computer will operate and manage certain programs. IoT operating systems connect to the software for managing IoT devices.[3]
With advancing technology, we go to mostly things automate, where We've got smart worlds, smart towns, intelligent houses; all fitted with smart IoT devices which can perform tasks by Self. IoT is a uniquely identifiable Integrated System Communicating devices that interchange data in a Network linked to give facilities. IoT Appliances are not just wireless built-in devices it's more than.[4]
IoT is Wireless Sensor Network Interconnection (WSN) Devices [5, 6] with Cloud room. Generally, IoT devices do have carbon depletion and memory capital.
This paper's work can be summed up as follows:

1 we will discuss important features of IoT for operating system design in section II.
2 Discussing recent articles which investigate the different operating system for the IoT in section III.
3 Investigating a different kind of challenges in operating systems for the internet of things in section IV,
4 Provide in Table1 various Comparative Analysis of Operating Systems for IoT Devices.
5 The conclusion of this paper is in the last section V. rectification is not possible.

II. IMPORTANT FEATURES IN IOT OS DESIGN

A. Architecture

The main part of the OS is a kernel. The Kernel Organization composes an operating system framework that influences both the size of the application programs and the way that they provide services.[7] There are a few OS structures. A portion of the notable ones is microkernel architectures, while others are monolithic. There is no structure in a monolithic architecture. It is a solitary, enormous cycle that runs in a single location space. The kernel will directly call for functions. Its administrations are delivered independently and each help offers an alternate interface. All device resources are packed into one image of the network.[8] A minimalist kernel is used in the microkernel architecture. In that, the kernel architecture is divided into Parts, called servers Parts. Some of the servers are running Space of the kernel and user space some running. All servers are kept separate and run in different address spaces absolutely. Due to the minimal functionalities, it has kernel size is reduced significantly.[9]..

B. Programming Model

The selection of the programming models is important because it affected by several factors. Parallelism, in particular, the hierarchy of memory and competition determine the model to use. The programming design itself influences the system's efficiency and profitability. Its function is to utilize the below architecture for top-level applications.[10] The programming model also aims to increase the performance of developers. The APIs and languages of programming adopt a programming model and sum up the fundamental framework. For hardware interface assembly language is the best option, [9, 11]. but hold up for significant level dialects is required for a calm simple turn of events. On restricted policy, however, it is difficult to provide high-level languages

Manuscript received on 21 December 2020 | Revised Manuscript received on 29 December 2020 | Manuscript Accepted on 15 January 2021 | Manuscript published on 30 January 2021.
***** Correspondence Author
Waheed Javed, University of Management and Technology (PAKISTAN)
Gulnaz Parveen, University of Management and Technology (PAKISTAN)
Dr. Adnan Abid, University of Management and Technology (PAKISTAN)

C. Scheduling

The planning procedure is one of the vital determinants of the productivity of the framework. The scheduling algorithm is for the most part rely upon turnaround time, reaction time, throughput, reasonableness, and pausing[12]. There are a lot of applications that given the diversity and time constraints of the IoT tasks.

To achieve the timeline, the queue manager should be a real-time queue manager to complete a job within specific periods. Besides, schedulers in IoT systems should be energy-efficient and multi-task[6]..

D. Networking

The prerequisites of IoT gadgets incorporate an Internet network. It should be conceivable to cooperate with low force utilization through IoT elements. Traditional

WSN network technologies and TCP / IP stacks are not available IoT-friendly. Although the previous one does not fulfill the objectives The last requirements less intricacy, less memory, and low force Middle intermediaries to permit developing correspondence channels to address peers. Besides, WSN protocols such as ZigBee, Bluetooth, Z-Wave, Wavenis etc.[13] met the individual smart devices requirement but did not fulfill the IoT's wide-ranging connectivity needs. We need an open norm for consistent Internet correspondence. Besides, a light-weight, solid, Internet-empowered IoT stack ought to likewise be accessible.

The stack should be versatile to satisfy the specifications of an extensive variety of IoT applications with minimal adjustments. The stack must be versatile. Ipv6 is obligatory to support IoT systems in huge networks with unique identities. Mechanisms such as 6LoWPAN, RPL (IPv6 Low-Power Routing Protocol [14]for the Network of Low-Powered Wireless Personnel Area), and CoAP are developed. For devices with low capacity Compression header and inclusion of minimum characteristics help to keep protocols IoT viable.

E. Management of memory

Management of memory works like abstraction technique in programming, which at backend handles cache, allocation, and de-allocation of memory, virtual memory, actual location planning lastly memory assurance.[15] The simple and small kernel is important In IoT devices, but IoT operating systems no memory of the executive's unit and Floating Point Unit on IoT devices where the main purpose is the small and simple kernel. Based on the type of application and platform support the required amount of memory management. Static or dynamic memory allocation can be used.[16] The static allocation of memory is easier, but with a dynamic approach, it is possible to obtain the flexibility of run-time storage.

F. Portability

OS to various hardware platforms should be easy to portable. A wide range of hardware architectures should be supported. IoT ranges between 8-bit to 32-bit microcontrollers. The OS ought to use the framework at the edge. Moreover, the IoT is a broad range area of application[17]. The OS should be according to the application's particular necessities and it should give the right information about the context.

G. Energy Efficiency

When we talk about reducing power is not just mean that power works long time its means saving money and increasing product life span.[18] In IoT battery-based devices efficient energy is required and it should be considered when IoT OS is being designed.

The IoT requires power management and power-efficient mechanism as compared to standard network protocols. In Radio the transceiver is the most power-consuming part in memory, due to those protocols like 6lowpan, RPL and LoRaWan use power-efficiencies to avoid as much as they can. Designers strive to optimize power on the user side of the microcontroller, well designed whenever possible.[6, 9].

III. EXISTING OPERATING SYSTEM

Some operating systems for IoTs are examined in this segment. These operating systems were chosen according to a number of factors including use, positive performance, interesting features, and Characteristics.

A. *Contiki*

Contiki is an adaptable and versatile OS. It develops in c but with restrictions. Contiki underpins both occasion driven and multi-stringing. Its architecture is monolithic. Protothreads provide less multi-threading. Multiple threads share an unwound stack for context adjustment. The key tasks are protothreads that deliver competitors and prevent preemptive CPU monopolies.[11]

They give contingent capacity hindering in a consecutive guidance succession. No clear device monitoring mechanism exists in Contiki. It provides different microcontroller devices such as Atmel ARM, Atmel AVR, STM32w,TIMSP430 /CC2430 /CC2538 /CC2630 /CC2650, LPC2103, Freescale MC13224, Microchip dsPIC, Mirochip PIC32. contiki

supports CoAP, 6LoWPAN and RPL networking protocols.[19].

B. Mbed

Silicon Labs and ARM ® have worked together for the Mbed OS Provide mbed power administration APIs that open the Save power capabilities for ARM Cortex ® -M for power-efficient In the mbed community, applications. APIs are efficient Saving still higher power on the EFM32 Gecko Silicon Labs MCUs.-MCUs. [20]

The APIs also allow you to save energy by EFM32 Input and Output operations to be carried out smoothly though EFM32 is completed The core of the processing is in sleep mode or other. In addition to deep sleep mood then save more power. Mbed supports Bluetooth, Wi-Fi, Zigbee IP / LAN , Cellular, and 6LoWPAN. [6].

C. Tiny

When you want the microcontroller to sleep as much as possible, then you can use the split stage and occasion driven execution model for a small operating system. If there is no work, the programmer sets the CPU in the sleep state. Thus, while the CPU wastes no energy waiting for other tasks and hardware components. TinyOS carry up Broadcast based Routing, Multi-Path Routing, Geographical Routing, Routing Reliability-based, TDMA base Routing.[21].

TABLE I: Comparative Analysis of Operating Systems for IoT Devices

OS → / properties ↓	Contiki	Mbed	RIOT	Tiny	Brillo	LiteOS
ROM	Less than 30kb	Less than 37kb	5kb	Less than 4kb	128 MB	128 kb
RAM	Less than 2kb	Less than 8kb	1.5kb	Less than 1kb	32 Mb	4 kb
Kernel	Modular	Layered	Microkernel	Monolithic	Microkernel	Modular
Real-point Sustain	Partial	definitely	Yes	Nope	-	unfair
Dev. Terminology	C (partial)	C, C++	C, C++	NesC	C, C++	C, LiteC++
Scheduling	Preemptive	Priority-based	Priority-based, Tickless	Preemptive FIFO, EDF	Completely Fair[7]	Priority-based, RR
Energy efficient	Yes	Yes	Yes	Yes	-	Yes
Event Driven	Yes	-	-	Yes	-	Yes
Reliability	Yes	-	Yes	Yes	-	Yes
Programming Model	Protothreads	Event-driven	Multi-threading	Event-driven, Threads	Multi-thread	Multi-thread
Open Source	Positively	Positively	Positively	Positively	-	Positively

D. RIOT

Riot Operating System Scheduler functions and implements without routine events a tickles planner that can work on a restricted To achieve maximum energy efficiency[22], equipment. Perhaps notably, Clock tikes are used by schedulers to wake up frequently to see if there must be something to do. However, on the off chance that the processor is set up, It needs to wake from the force sparing the rest of the inert State any clock, regardless of whether nothing is to be done Energy-limited frameworks are not alluring. One important thing in which is that many modes are available for energy saving. RIOT support IPv6 , 6LoWPAN , RPL , CoAP , UDP, CBOR, and Open WSN [23] communication protocols.

E. Brillo

The Brillo operating system of Google is the edition that is reduced in size to more than half of the global smartphones of the mobile operating system of Google's Android. 32MB/64MB of RAM is used and also connects with Google technologies. Brillo to meet interoperable standards introduced offers a protocol is used that is called 'Weave' that is used for synchronizing data between devices. it likewise gives underpins for the gadget to telephone correspondence, at that point clients effortlessly control the gadgets [25].

F. LiteOS

LiteOS is open source and uses in cyber-physical systems, smart homes, and wearable devices. It's just like the UNIX operating system because its environment resembles UNIX. Its three small modules of an operating system that is kernel LiteFS and LiteShell.these three subsystems work separately. LiteShell uses commands for interaction with the device. And at core level work kernel that execute commands. LiteFS used in directory operations.[26].

IV. SECURITY CHALLENGES OF IOT OPERATING SYSTEMS

We explore different forms of RPL attack will discuss them.

1. Smurf attack
2. Blackhole attack
3. Wormhole attack
4. Sybil
5. Sinkhole attack
6. Clone ID attack
7. Hello Flooding Attack

A. Smurf attack

Smurf attacks due to DoS in a network that creates a network that cannot be treatable. If we want to deal with vulnerabilities then must we know that (ICMP). Internet Control Message Protocol monitors the network nodes, and network administrator information will be altered. The status of other IoT nodes is also monitored. If the ping is returned, it means it's working.[27].

B. Blackhole Attack

Heterogeneous communication protocols (HCP) are vulnerable to various kinds of attacks like network sniffing, modification, Dos, etc. In [28] many attacks are described. Blackhole attack is on the network layer, this targets RPL implementation of contiki operating system. The author demonstrated this attack in, black hole attack starts with the compromised node which acts like malicious and drops packet which are routed through it and cause disruptions in a network data flow. Blackhole attacks can be easily concealed and the attacked network may have behaved like a health network[29]. It is very important to know that only contiki OS-based devices are vulnerable to this kind of attack. the best defense against black hole attack in RTL protocol is to implement RIOT OS, Tiny OS which are not vulnerable to this kind of attack.[30]

C. Wormhole Attack

In [31]author described that Wormhole attack is viewed as extreme assaults on IoT steering. A passage is set up between two hubs and the bundle is sent among one another. Such far off malignant hubs imagine that they are so nearby that neighboring nodes send packets. Figure 1 shows the overall idea of an attack with a wormhole. There are four operating modes.**1**. Encapsulation: Throughout this mode, one end of the IoT network has a colluding node and the other side of the Route Request packet, in which the tunnel is created and the nodes assume that they are connected near and directly. **2.** Packet Relay: Assailant hubs transfer bundles between two real hubs in this structure. In this method, a malicious node linked between the two legitimate nodes is two legit nodes that aren't directly interconnected.

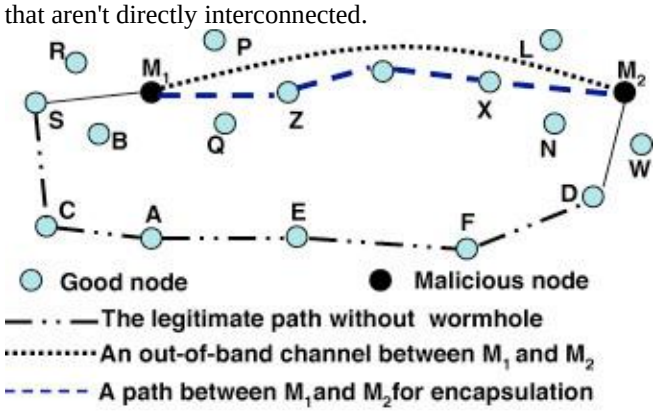

Fig 1: Generalized diagram of Wormhole Attack[31]

3. Out of Band Channel: Long wired and wireless links are used in the Wormhole[27] based attack mode. A special type of hardware is required to launch this attack.

To make a consistent link between two attacker nodes high bandwidth link is used. These two attacker nodes are physically are on long-distance, by attracting traffic these nodes turn to band channel mode

D. Sybil

The Sybil attack is a reputational assault in which identities are subverted in peer-to-peer networks. False GPS profiles may be replicated to manipulate social navigation systems due to a lack of identification in these networks. The weakness may be used improperly to threaten the health of citizens. To rouse a Good Samaritan to come to help in a desolate spot and cause harm, for example, the malicious user will simulate a fake catastrophe alarm. The intruder will transform the emergency teams' focus from a real disaster.[32]. Sybil Attack happens when a single malicious identity reveals multiple identities and gains network power. Different attacks such as intrusion nodes, replay attacks, etc ... primarily cryptographic protocols and key administration schemes are involved. Sybil attacks are a big WSN attack, a compromise attacker attack or catch several nodes in Sybil, to execute other attacks, inserting compromise nodes [33]across the network. Such knots cause high resource consumption.

E. Clone ID Attack

Clone attacks may be considered a special node compromise attack type in which two or more concessional nodes with the same ID are simultaneously available on the network. In other words, cloned nodes are correct copies of the compromise initial node[16]. Particularly if one node is compromised We only find the aftermath of agreement and cloning as the first step in the conduct of a clone attack. Note that a node compromise attack is different than a clone attack. The former typically concerns a situation where a particular node is compromised by the attacker, then places the compromised node in the network, while the latter applies to the situation where a given node is compromised by the attacker and the compromised node is placed in the network by several replicated copies of the compromised node. ScreeningClone attack strategies are often distinct and independent of a single node compromise being observed[34].

F. Hello Flooding Attack

A node will trigger the Hello flood attacks which transmit a very powerful Hello packet so that many nodes even in the distance it are selected by the network as the parent node. Every message now this parent needs to be routed with multi-hops. Delay rises. Hi, messages are sent to a wide audience Number of network nodes in a wide region. Nodes are then told that they have the intruder node neighbor to reply to all the nodes' Email HELLO, and the energy was wasted. Therefore, the network is confused [35].

G. Overload Attack

Overload Attacks are mostly when occurs when the intended user can not utilize network or resource services. Overload attack is directly effect on network. Face mostly problem Excess traffic and Energy scavenging[36].

H. Authentication Attack

When entering the network, 6LoWPAN does not have an authentication mechanism for nodes. Any malicious node may join because of this. The author identifies nodes with connections to the 6LoWPAN [37] network for authentication. The source of this is the administrative authorization. It consists of 4 steps: authorization of nodes, filtration of information, and propagation of the approved node list, and detection of nodes. List of any nodes with a layer 2 address will be specified in a border router, which allows the presence of a node to be specified with the help of these addresses[38].

V. CONCLUSION

In this survey, we have mostly focused on those operating systems that are open source. These IoT systems are assessed and compared with certain criteria and evidence, followed by all findings in this paper. The purpose of the study clearly generated important results which are very useful for researchers at all IoT rates. In real, for beginners is also important due to technical aspects. From the above research, we conclude that IoT systems are generally well organized and scalable, but some protocols and implementation problems remain, due to which various kinds of attacks such as RPL attacks are vulnerable over networks. but for more research is needed against 6LoWPAN and RPL attacks, Clone ID, Blackhole, wormhole, Sybil, etc such kind of attacks need IDS based detection mechanism.

Although the IoT required the optimum operating system that is not created yet, it has to take several things into account challenges studied in this article should be taken into consideration safety and customization of the operating system OS with certain specific Internet of Things OS services Requests.

REFERENCES

1. Zikria, Y. B., Yu, H., Afzal, M. K., Rehmani, M. H., & Hahm, O. (2018). Internet of things (iot): Operating system, applications and protocols design, and validation techniques.
2. Ain, Q. U., Iqbal, S., Khan, S. A., Malik, A. W., Ahmad, I., & Javaid, N. (2018). IoT operating system based fuzzy inference system for home energy management system in smart buildings. Sensors, 18(9), 2802.
3. Segura Garcia, J., Perez Solano, J. J., Cobos Serrano, M., Navarro Camba, E. A., Felici Castell, S., Soriano Asensi, A., & Montes Suay, F. (2016). Spatial statistical analysis of urban noise data from a WASN gathered by an IoT system: Application to a small city. Applied Sciences, 6(12), 380.
4. Rodriguez-Zurrunero, R., Utrilla, R., Rozas, A., & Araujo, A. (2019). Process management in IoT operating systems: cross-influence between processing and communication tasks in end-devices. Sensors, 19(4), 805.
5. Mainetti, L., L. Patrono, and A. Vilei. Evolution of wireless sensor networks towards the Internet of Things: A survey. in SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks. 2011.
6. Sabri, C., L. Kriaa, and S.L. Azzouz. Comparison of IoT constrained devices operating systems: A survey. in 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). 2017. IEEE.
7. Massalin, H., & Pu, C. (1992). A lock-free multiprocessor OS kernel. ACM SIGOPS Operating Systems Review, 26(2), 108.
8. Dautenhahn, N., Kasampalis, T., Dietz, W., Criswell, J., & Adve, V. (2015, March). Nested kernel: An operating system architecture for intra-kernel privilege separation. In Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems (pp. 191-206).
9. Gaur, P. and M.P. Tahiliani. Operating systems for IoT devices: A critical survey. in Proceedings of the 2015 IEEE Region 10 Symposium. 2015.
10. Tompson, J., & Schlachter, K. (2012). An introduction to the opencl programming model. Person Education, 49, 31.
11. Gaur, P. and M.P. Tahiliani. Operating Systems for IoT Devices: A Critical Survey. in 2015 IEEE Region 10 Symposium. 2015.
12. Goel, N., & Garg, R. B. (2013). A comparative study of cpu scheduling alg
13. Molina, A. J., Wikstrom, J. D., Stiles, L., Las, G., Mohamed, H., Elorza, A., ... & Shirihai, O. S. (2009). Mitochondrial networking protects β-cells from nutrient-induced apoptosis. Diabetes, 58(10), 2303-2315.
14. Mahajan, A., & Rajlich, L. (2014). U.S. Patent No. 8,832,676. Washington, DC: U.S. Patent and Trademark Office.
15. Liu, L., Yang, S., Peng, L., & Li, X. (2019). Hierarchical hybrid memory management in OS for tiered memory systems. IEEE Transactions on Parallel and Distributed Systems, 30(10), 2223-2236.
16. Kunimatsu, A., Nakai, H., Sakamoto, H., & Maeda, K. (2012). U.S. Patent No. 8,135,900. Washington, DC: U.S. Patent and Trademark Office.
17. Yu, M., Wundsam, A., & Raju, M. (2014). NOSIX: A lightweight portability layer for the SDN OS. ACM SIGCOMM Computer Communication Review, 44(2), 28-35.
18. Ijaodola, O. S., El-Hassan, Z., Ogungbemi, E., Khatib, F. N., Wilberforce, T., Thompson, J., & Olabi, A. G. (2019). Energy efficiency improvements by investigating the water flooding management on proton exchange membrane fuel cell (PEMFC). Energy, 179, 246-267.
19. Pongle, P. and G. Chavan. A survey: Attacks on RPL and 6LoWPAN in IoT. in 2015 International conference on pervasive computing (ICPC). 2015. IEEE.
20. Muhammad, A., Afzal, B., Imran, B., Tanwir, A., Akbar, A. H., & Shah, G. (2019, June). OneM2M architecture based secure MQTT binding in Mbed OS. In 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 48-56). IEEE.
21. Grgic, K., D. Zagar, and V. Krizanovic. Security in IPv6-based wireless sensor network—Precision agriculture example. in Proceedings of the 12th International Conference on Telecommunications. 2013. IEEE.
22. Baccelli, E., Hahm, O., Günes, M., Wählisch, M., & Schmidt, T. C. (2013, April). RIOT OS: Towards an OS for the Internet of Things. In 2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS) (pp. 79-80). IEEE.
23. Bugneac, D. (2017). Security protocols for IoT environments: combining existing communication protocols and security protocols for safer and cheaper communication (Master's thesis, Πανεπιστήμιο Πειραιώς).
24. Jaskani, F.H., et al., An Investigation on Several Operating Systems for Internet of Things. EAI Endorsed Trans. Creative Technol., 2019. **6**(18): p. e4.
25. Chandra, T.B., P. Verma, and A. Dwivedi. Operating systems for internet of things: A comparative study. in Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. 2016.
26. Cao, Q., Abdelzaher, T., Stankovic, J., & He, T. (2008, April). The liteos operating system: Towards unix-like abstractions for wireless sensor networks. In 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008) (pp. 233-244). Ieee.
27. De Donno, M., et al., DDoS-capable IoT malwares: Comparative analysis and Mirai investigation. Security and Communication Networks, 2018. **2018**.
28. Lee, C., et al. Securing smart home: Technologies, security challenges, and security requirements. in 2014 IEEE Conference on Communications and Network Security. 2014. IEEE.
29. Seyedi, B. and R. Fotohi, NIASHPT: a novel intelligent agent-based strategy using hello packet table (HPT) function for trust Internet of Things. The Journal of Supercomputing, 2020.
30. Patel, H.B. and D.C. Jinwala. Blackhole detection in 6LoWPAN based internet of things: an anomaly based approach. in TENCON 2019-2019 IEEE Region 10 Conference (TENCON). 2019. IEEE.
31. Deshmukh-Bhosale, S. and S.S. Sonavane, A real-time intrusion detection system for wormhole attack in the RPL based Internet of Things. Procedia Manufacturing, 2019. **32**: p. 840-847.
32. Gaire, R., et al., Crowdsensing and privacy in smart city applications, in Smart cities cybersecurity and privacy. 2019, Elsevier. p. 57-73.
33. Xiao, L., Greenstein, L. J., Mandayam, N. B., & Trappe, W. (2009). Channel-based detection of sybil attacks in wireless networks. IEEE Transactions on Information Forensics and Security, 4(3), 492-503.
34. Mirshahjafari, S.M.H. and B.S. Ghahfarokhi, Sinkhole+ CloneID: A hybrid attack on RPL performance and detection method. Information Security Journal: A Global Perspective, 2019. **28**(4-5): p. 107-119.
35. Dubey, A., D. Meena, and S. Gaur, A survey in hello flood attack in wireless sensor networks. Int. J. Eng. Res. Technol, 2014. **3**
36. Mangelkar, S., S.N. Dhage, and A.V. Nimkar. A comparative study on RPL attacks and security solutions. in 2017 International Conference on Intelligent Computing and Control (I2C2). 2017. IEEE.
37. Yibo, C., Hou, K. M., Zhou, H., Shi, H. L., Liu, X., Diao, X., ... & de Vaulx, C. (2011, September). 6LoWPAN stacks: A survey. In 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (pp. 1-4). IEEE.
38. Asim, M. and W. Iqbal, Iot operating systems and security challenges. International Journal of Computer Science and Information Security, 2016. **14**(7): p. 314.

AUTHORS PROFILE

Waheed Javed was born in Multan, Pakistan, in 1994.He received the BS degree from the Preston University ,Pakistan, in 2017.He is currently MSCS student in the School of Computer Science, university of management and technology, Lahore, Pakistan. He is currently working as an Lecturer with the Department of Computer Science, Inspire Group of Colleges, Pakistan. . His research interests include computer science education, Blockchain, and Data mining..

Gulnaz Parveen is a MSCS student in the School of Computer Science, university of management and technology, Lahore, Pakistan. She obtained the MCS degree in Computer Science from BZU, Multan, Pakistan. Her research interests include Internet of Things, Computer Vision and Pattern Recognition**.**

Dr. Adnan Abid was born in Gujranwala, Pakistan, in 1979. He received the B.S. degree from the National University of Computer and Emerging Science, Pakistan, in 2001, the M.S. degree in information technology from the National University of Science and Technology, Pakistan, in 2007, and the Ph.D. degree in computer science from the Politecnico Di Milano, Italy, in 2012. He spent one year in EPFL, Switzerland, to complete his M.S. thesis. He is currently working as an Associate Professor with the Department of Computer Science, University of Management and Technology, Pakistan. He has almost 70 publications in different international journals and conferences. His research interests include computer science education, information retrieval, and data management. He has served as a Reviewer in many international conferences.