Adamu Murtala Zungeru
Joseph M. Chuma · Caspar K. Lebekwe
Pendukeni Phalaagae
Jwaone Gaboitaolelwe

Green Internet of Things Sensor Networks

Applications, Communication Technologies, and Security Challenges

Springer

Green Internet of Things Sensor Networks

Adamu Murtala Zungeru • Joseph M. Chuma
Caspar K. Lebekwe • Pendukeni Phalaagae
Jwaone Gaboitaolelwe

Green Internet of Things Sensor Networks

Applications, Communication Technologies, and Security Challenges

Springer

Adamu Murtala Zungeru
Faculty of Engineering and Technology
Botswana International University
of Science & Technology
Palapye, Botswana

Caspar K. Lebekwe
Faculty of Engineering and Technology
Botswana International University
of Science & Technology
Palapye, Botswana

Jwaone Gaboitaolelwe
Faculty of Engineering and Technology
Botswana International University
of Science & Technology
Palapye, Botswana

Joseph M. Chuma
Faculty of Engineering and Technology
Botswana International University
of Science & Technology
Palapye, Botswana

Pendukeni Phalaagae
Faculty of Engineering and Technology
Botswana International University
of Science & Technology
Palapye, Botswana

ISBN 978-3-030-54982-4 ISBN 978-3-030-54983-1 (eBook)
https://doi.org/10.1007/978-3-030-54983-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The global concerns over the inappropriate utilization of abundant renewable energy sources, damages due to the instability of prices of fuel, and fossil fuels' effect on the environment have led to an increased interest in green energy. The generation of green energy has opened doors for wireless sensor networks that can sustain themselves through energy harvesting techniques. The topics discussed in this book are selected in such a way as to help undergraduates and researchers with interest in wireless sensor networks. A distinguishing feature of the book is that it highlights the need for a robust wireless sensor network that can self-harvest energy to sustain its nodes. The strategies of energy harvesting to support the networks' nodes and communication protocols to be able to reduce the amount of power consumed in the transmission of data in the Internet of Things sensor networks are discussed in detail throughout the book.

The book is divided into seven chapters: Chapter 1 gives an overview of the Green Internet of Things sensor networks. Chapter 2 presents and discusses applications of wireless sensor networks by characterizing wireless sensor networks into three application domains: consumer, commercial, and Industrial Internet of Things (IIoT). Chapter 3 shows and describes the design and implementation of smart IoT devices through a specific example. Chapter 4 describes the design and simulation of a standalone photovoltaic system to support the Internet of Things sensor devices following modeling and sizing procedures. Chapter 5 gives an overview of research issues and challenges in IoT sensor networks through the 4 tier IoT architecture. In Chap. 6, methods, mechanisms, and techniques for securing devices and communication of sensor data are presented and analyzed. Finally, Chap. 7 highlights some areas where further research is needed to come up with robust and lightweight security mechanisms for future IoT networks.

We enjoyed working on this book, and we hope you enjoy learning from it. We hope that the book will be highly useful to students and researchers at large.

Palapye, Botswana — Adamu Murtala Zungeru
Palapye, Botswana — Joseph M. Chuma
Palapye, Botswana — Caspar K. Lebekwe
Palapye, Botswana — Pendukeni Phalaagae
Palapye, Botswana — Jwaone Gaboitaolelwe

Introduction

The global concerns over the inappropriate utilization of abundant renewable energy sources, damages due to the instability of prices of fuel, and fossil fuels' effect on the environment have led to an increased interest in green energy. Besides, conventional/traditional building switching systems (TBSS) used in buildings face many problems such as rising electricity prices and insecure wall sockets and switches that are vulnerable to misuse. These problems pose inconveniences for the residents of such buildings. As a result, in both private and public buildings, there is a desire to reduce electric usage, automate appliances, and move towards optimizing the electricity usage of buildings. This book presents methods for advancing green IoT sensor networks and IoT devices. Three main methods are presented: a standalone system to support IoT devices that is informed by the amount of energy the solar array system can produce; a model of securing a building's main power supply against unauthorized use; and security of the IoT devices and their networks. For each, the book outlines the methods, presents security and privacy issues and their solutions. The work suggests a layered approach to expose security issues and challenges at each layer of the IoT architecture and proposes techniques used to mitigate these challenges. Finally, directions and perspectives are drawn and discussed for future directions in securing IoT sensor networks, which involves artificial intelligence, blockchain technology, sensor Internet of People, context-aware sensing, cloud infrastructure, security and privacy, and the Internet of Everything.

Contents

1 Introduction to Green Internet of Things Sensor Networks 1
Adamu Murtala Zungeru, Lucia K. Ketshabetswe, Bokani Mtengi, Caspar K. Lebekwe, and Joseph M. Chuma
1.1 Background 1
1.2 Internet of Things Sensor Networks 2
1.2.1 Green Internet of Things Sensor Networks (GIoTSNs) 3
1.3 Energy Harvesting 4
1.4 Communication Protocols 6
1.5 Data Compression 6
1.6 Summary 7
References 8

2 Applications and Communication Technologies in IoT Sensor Networks 9
Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong
2.1 Introduction 9
2.2 IoT Sensor Networks Applications 9
2.2.1 Consumer IoT 10
2.2.2 Commercial IoT 12
2.2.3 Industrial IoT 13
2.3 IoT Sensor Networks Communication Technologies 13
2.3.1 Wireless Body Area Networks (WBAN) 15
2.3.2 Wireless Personal Area Networks (WPANs) 16
2.3.3 Wireless Local Area Network (WLAN) 17
2.3.4 Wireless Wide Area Network (WWAN) 18
2.4 Summary 20
References 21

3 Smart Internet of Things Devices and Applications Specific 25
Jwaone Gaboitaolelwe, Adamu Murtala Zungeru, Joseph M. Chuma, Nonofo Ditshego, and Caspar K. Lebekwe
3.1 Introduction 25
3.2 Description of the Example Problem 26
3.2.1 Problem Definition 26
3.2.2 Aim 26
3.2.3 Objectives 26
3.3 Description of the Example Solution 27
3.3.1 Secured Smart Home Switching System 27
3.3.2 Smart Hub Design 28
3.3.3 Smart Switch And Smart Socket Design 39
3.3.4 Smart Switchboard Design 46
3.4 Implementation of the Designed Solution 50
3.5 Summary 51
References 53

4 Design of Photovoltaic System for IoT Devices 55
Adamu Murtala Zungeru, Joseph M. Chuma, Dauda Duncan, Bakary Diarra, Modisa Mosalaosi, Bokani Mtengi, and Jwaone Gaboitaolelwe
4.1 Introduction 55
4.2 Technical Specifications of the Design 57
4.2.1 Equivalent Peak Solar Radiation Hours and Photovoltaic Array Sizing 63
4.2.2 Equivalent Battery Sizing in the Photovoltaic Systems 66
4.3 Design of Photovoltaic System and MPPTs 67
4.3.1 Influence of Ambient Solar Irradiance on PV Cell 70
4.3.2 Influence of Ambient Temperature on PV Cell 70
4.3.3 Maximum Power Point Tracking 70
4.4 Photovoltaic Modules Connected to a Load via Battery 74
4.4.1 Lead Acid Battery 74
4.4.2 Battery Charge and Discharge System 75
4.5 Simulation Results 77
4.6 Summary 78
References 79

5 Security Challenges in IoT Sensor Networks 83
Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong
5.1 Introduction 83
5.2 Background 83
5.3 Perception Layer 84
5.3.1 Node Capture/Tempering 85
5.3.2 Fake Node and Malicious Data 85

5.3.3 Replay Attack . . . 85
5.3.4 Side-Channel Attack . . . 85
5.3.5 DDoS/DoS . . . 86
5.3.6 Sleep Deprivation . . . 87
5.3.7 Booting Attack . . . 87
5.3.8 False Data Injection . . . 87
5.3.9 Malicious Code Injection . . . 87
5.4 Network Layer . . . 88
5.4.1 Eavesdropping . . . 88
5.4.2 Phishing Site . . . 88
5.4.3 Access Attack . . . 89
5.4.4 DoS/DDoS . . . 89
5.4.5 Data Transit . . . 89
5.4.6 Routing Attack . . . 89
5.4.7 Man-in-the-Middle Attack . . . 90
5.5 Cloud Layer . . . 90
5.5.1 Man-in-the-Cloud Attack . . . 91
5.5.2 SQL Injection . . . 92
5.5.3 Malware Injection . . . 92
5.5.4 Flooding Attack . . . 92
5.6 User Interface/Application Layer . . . 93
5.6.1 Access Control Attacks . . . 93
5.6.2 Malicious Code Injection . . . 93
5.6.3 Sniffing Attacks . . . 94
5.6.4 Application-Specific Vulnerabilities . . . 94
5.7 Summary . . . 94
References . . . 95

6 IoT Sensor Networks Security Mechanisms/Techniques . . . 97
Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong
6.1 Introduction . . . 97
6.2 Perception Layer . . . 97
6.2.1 Authentication . . . 98
6.2.2 Hardware Security . . . 98
6.2.3 Lightweight Encryption . . . 99
6.2.4 Protecting Sensor Data . . . 100
6.3 Network Layer . . . 102
6.3.1 Encryption Mechanism . . . 102
6.3.2 Secure Communication . . . 102
6.3.3 Trust Recommendation . . . 103
6.4 Cloud Layer . . . 104
6.4.1 Secure Cloud Computing . . . 104
6.4.2 Secure Multi-Party Computation . . . 104
6.5 User Interface Layer . . . 105

6.5.1 Information Privacy . . . 105
6.5.2 Key Agreement . . . 106
6.5.3 Data Management . . . 106
6.6 Future Works . . . 107
6.6.1 Artificial Intelligence . . . 107
6.6.2 Blockchain for IoT Security . . . 107
6.6.3 Machine Learning for Data Security . . . 108
6.6.4 Context-Aware Sensing . . . 108
6.6.5 Cloud Infrastructure . . . 109
6.6.6 Sensor Internet of People . . . 109
6.6.7 Internet of Everything . . . 110
6.7 Summary . . . 111
References . . . 112

7 Future Challenges of IoT Sensor Networks . . . 119
Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong
7.1 Introduction . . . 119
7.2 Hardware Security . . . 119
7.3 Lack of Lightweight Cryptographic Algorithms . . . 120
7.4 Lack of Lightweight Trust Management Systems . . . 120
7.5 Lack of Lightweight Secure Routing Protocols . . . 120
7.6 Lack of Lightweight Anti-Malware Solutions . . . 121
7.7 Summary . . . 121
References . . . 122

Index . . . 123

Chapter 1
Introduction to Green Internet of Things Sensor Networks

Adamu Murtala Zungeru, Lucia K. Ketshabetswe, Bokani Mtengi, Caspar K. Lebekwe, and Joseph M. Chuma

1.1 Background

It has been observed over the years that an escalating rate of deployments of real-life applications of wireless sensor networks (*WSNs*) is realized. The various applications include environmental monitoring, medicine, health care, surveillance, power monitoring, structural monitoring, disaster detection, home automation, smart building, rescue, smart agriculture, traffic control, and object tracking [1]. Research has identified *WSNs* as networks with more limited energy than other wireless networks. Power consumption has risen as a major setback that limits network lifetime in these networks [1, 2]. This is usually the power lost during transmission of information from a source to its destination. This loss is greatly experienced during the transmission of this information than during processing [2, 3]. Other limitations of the conventional sensor networks include restricted radio bandwidth, memory, processing capability, packet size, and high rates of packet loss. Various approaches that aim to reduce this loss of power as an effort to extend the network lifetime have been proposed. Communication between sensor networks cannot be achieved physically through sensor nodes. There is a need for an Internet-based network (Network to Networks communications).

Since wireless sensor networks are deployed in larger volumes through several tiny elements called sensor nodes that are powered by small rechargeable batteries, the networks are faced with a number of limitations that pose a threat to the network lifetime. The advancement of the traditional WSNs gave birth to the Internet of Things sensor networks. In general, the Green Internet of Things Sensor Networks (GIoTSNs)

A. M. Zungeru (✉) · L. K. Ketshabetswe · B. Mtengi · C. K. Lebekwe · J. M. Chuma
Faculty of Engineering and Technology, Botswana International University of Science and Technology, Palapye, Botswana
e-mail: zungerum@biust.ac.bw

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_1

combined four critical components: “Energy Harvesting,” “Internet,” “Things,” and “Sensor Networks.” The simple explanation to this is, “Any device having the capability and is compatible to connect to the Internet will come under Things.” The term refers to a network that connects anything with the Internet, according to established protocols. The connection can be through equipment designed for information sensing that conducts an exchange of information for communication purposes. Besides communication devices, “Things” include physical objects such as computers, personal devices including cars, home appliances, or smart devices, medical instruments, and industrial systems that are controlled through wireless communication, sensors, and “Miscellaneous Objects.” The nodes in the network interact with one another and are able to access anything anytime and from anywhere in the system (Network) [4]. IoT involves collecting, processing, and use of data for communication. Processing big data requires a large capacity for storage and high power consumption. Internet of Things is an Internet of three things: (1) people-to-people, (2) People to machine/things, and (3) Machine/things to machine/things [5]. The increased energy demands across the globe already have adverse environmental effects on society. The development of technologies to meet the needs of the smart world and sustainability by implementing green IoT aims to reduce carbon emission and power consumption. In this, it makes it possible for a network to survive for a longer time as it is “Green,” meaning that it can self-harvest energy to sustain the nodes in the network. The “Green” IoT technologies make it environmentally friendly by focusing on optimization of data centers through techniques of sharing infrastructure, which leads to increased energy efficiency and lower cost of operation. The inexpensive, low powered sensors will expand the application of IoT to even smaller objects in any kind of environment at affordable prices.

1.2 Internet of Things Sensor Networks

Wireless sensor networks are recognized as vital enablers for the Internet of Things [6]. They form a network that senses and control an environment while enabling interaction between computers, persons, and the environment. The *Internet of Things sensor network* is a network of networks that can overcome most of the sensor networks’ limitations. This network is made up of smart devices that can be identified in the networks. They should also be able to collect and share data over the Internet as well as process it. For real-time applications to be realized, storage services are made available to cloud platforms where other services can be performed on the data for end-users. End usage of data includes modeling and data analysis of the data gathered from the different *IoT* elements for informed decision making [4]. Figure 1.1 illustrates an IoT sensor network, with numerous small-size sensor nodes spatially distributed over chosen fields of networks. This makes the networks suitable for large-scale deployments. The hardware of a sensor includes four parts: the power module (battery), sensing unit, processing unit (usually a microcontroller for analog to digital conversion), and the transceiver unit [7, 8].

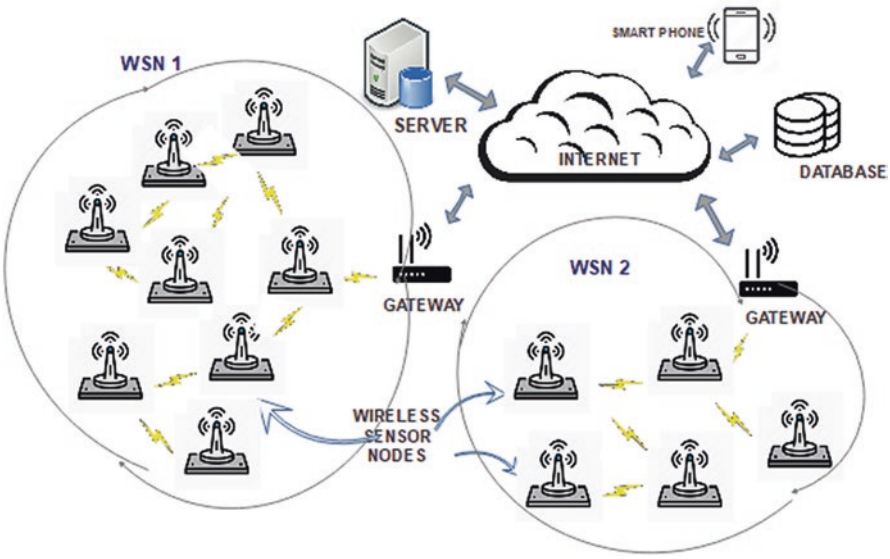


Fig. 1.1 An Internet of Things sensor network

They work collectively to monitor the sensor field and gather information about the environment. They are independent in their organization and communicate with each other wirelessly to achieve desired goals. The information that is gathered is propagated wirelessly from the source nodes to the sink node by multi-hop or single-hop communication [7].

The *WSNs* connect to the Internet through gateways and base stations. Wireless devices like smart phones, servers, and others can connect to the Internet to exchange information. Wireless sensor network technologies are advancing, making the cost of equipment affordable, thus expanding the market size of applications. The low-cost and low-power transceivers make it possible for WSN use in home automation and industrial monitoring applications. The goal for researchers and developers is to collect and analyze every piece of information around us to improve production efficiency and ensure optimal resource consumption.

1.2.1 Green Internet of Things Sensor Networks (GIoTSNs)

Due to the growing awareness of environmental issues around the world, green IoT technology initiatives should be taken into consideration. Environmental hazards like chemical emissions and energy dissipation normally accompany new inventions and innovations that are brought by new technologies that are needed by the world today. Sensors consume lots of power while performing tasks. In networking, green IoT aims to identify the location of the relay and the number of nodes that satisfy energy-saving and budget constraints [9]. There are three green IoT con-

cepts, namely design, leverage, and enabling technologies. Design technologies are concerned with the energy efficiency of devices, communications protocols, network architectures, and interconnections, while leverage technologies deal with cutting carbon emissions and enhancing energy efficiency. The reduction of negative consequences that come with these technologies to save the environment is crucial and inevitable. This can be achieved by considering implementing green IoT technologies to achieve the development of energy-efficient products and services associated with *IoT Sensor Networks*. This action will result in preserving natural resources, minimizing the technology impact on the environment and human health, waste reduction, efficient utilization of energy, and lower cost of operation. Principles of *GIoTSNs* highlight conservation of energy, that is, switching on only facilities that are required while those that are not needed are switched off. Using natural energy supply sources like wind, solar, and many others that are freely available is another way of generating clean energy. Communication technologies that improvise on the overall spectral efficiency, modulation techniques, usage of power, and frequency operation can be implemented to save energy. Improving *IoT sensor networks* lifetime through the deployment of energy-efficient communication protocols is another option of saving energy as well as migrating to cloud-based processing and storage to reduce operation costs and overcome security, bandwidth, and latency problems. Technologies that are associated with *GIoTSNs* are Smart Home, Smart Cities, Smart Grid, and many other Smart technologies [4]. The green Internet of Things is expected to bring significant technological developments in the wireless sensor network and its applications. The technology will make it possible to have a massive amount of sensors, devices, and "things," which will enable the new smart objects to perform certain functions autonomously. In this, communication between people and things, and between things themselves enable lower power consumption, and bandwidth utilization is maximized [10].

This book intends to combine the strategies of Energy Harvesting, Communication Protocols, and Data Compression to improve the lifetime of the *Internet of Things sensor networks*. Energy Harvesting will help to support the networks' sensor nodes' energy. At the same time, Communication Protocols will allow the reduction of the amount of power consumed in the transmission of data in the networks. The reduction of this power consumption can also be achieved through the application of Data Compression.

1.3 Energy Harvesting

Sensor nodes have limited energy in their power storage unit, making it challenging for a sensor node to remain operational for long periods. The solution is for the sensor nodes to be able to harvest ambient energy from the surroundings to recharge the batteries, which can then directly power the sensor nodes. Energy Harvesting is a process whereby energy is extracted from the surroundings and stored to supply low-power wireless devices. This is a natural energy that is freely

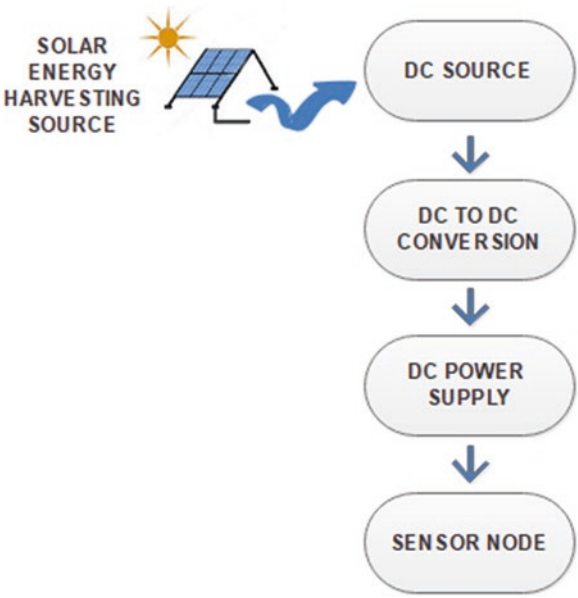


Fig. 1.2 Solar energy harvesting process

available in the environment. Sources of harvested energy can be radiation, e.g., solar, radio frequency (*RF*), thermal, e.g., heat, mechanical energy sources like blood or water flow, vibrations, wind, and many others [11]. Since this energy cannot directly power small devices like sensor nodes or batteries, they go through a process of signal conditioning, rectification, and power conversion. Figure 1.2 below illustrates a solar energy harvesting process. Optical energy from the sun is converted to direct current (dc) energy by the solar panel, and it is further converted by a *dc to dc* converter to a suitable dc power source, which is sufficient to power a sensor node. An energy harvesting system is, therefore, made up of an energy harvester, which collects energy from surrounding sources and transforms it into electric energy that can power the components of the sensor node like sensor unit, processor unit, and communications unit.

The energy management unit then receives the electric energy and processes it further. An energy storage unit also forms part of the harvesting system to store energy that can be used at a later stage, eliminating the dependency of sensor nodes battery power, thereby reducing costs. The reliance on batteries puts limitations on implementing WSNs for environmental monitoring applications. Due to the challenges with changing batteries of nodes regularly, nodes that have depleted their batteries are considered dead and cannot participate in the network operation. This challenge makes it even more critical to explore energy harvesting aware protocols [12]. The choice of a suitable energy harvesting system for a *WSN* should consider its application and area of deployment, where the energy source is abundant [13]. Over the past years, there has been a significant improvement in energy harvesting technologies, especially in their efficiencies. Energy harvesting devices that are

capable of providing continuous power output from various energy sources such as light and temperature based energy for smart building lighting and air monitoring applications have entered the market. Other products, such as those capable of converting mechanical vibration into electrical energy use by wireless sensor nodes, are also available. The energy made by fingers knocking the desk can support the sensor node sending 2 kb data to 100 m away every 60 s [14].

1.4 Communication Protocols

Communication protocols are another effective technique that is used to save power in an *Internet of Things sensor network*. They assist in discovering optimal paths for information sharing under existing sets of constraints in the network [7]. Their key role is to provide an efficient exchange of information between sensor nodes with less or no interruption. A good communication protocol determines better performance, reliability, and service of a sensor network [15]. They should be energy efficient and adopt optimization methods to enhance sensor nodes' efficiency in routing [4]. Different classes of communication protocols exist depending on the various ways in which the data is sent from the source point to the destination and also on the sensor network application. Traditionally information can be sent from a source node directly to the destination node, a communication process known as a single hop, which usually requires high transmission power. Alternatively information can be routed through intermediate nodes, which further passes the information to the destination node. This is called multi-hop communication and requires less transmission power. The whole process of gathering, processing, and forwarding information is termed routing and is handled by communication protocols.

1.5 Data Compression

Data Compression reduces the amount of data to be transmitted in the network and consequently saves a significant amount of power used for transmission. It is a simple and effective power-saving technique that is also robust and without loss. This power saving is achieved by eliminating redundant data and, to some extent, at the expense of the quality of data [3]. A lossless Data Compression technique is generally used in order to prevent loss of data. With lossless Data Compression, the original data and reconstructed data after compression (decompressed data) are the same. A block diagram that illustrates a simple Data Compression model is shown in Fig. 1.3.

Considering a stream of data "*ABABCCC*" in Fig. 1.3, redundant data is removed during compression, so fewer bits of data, "*2AB3C,*" are transmitted along the communications line. During decompression, redundant data is added to reconstruct the original symbols. Different Data Compression techniques exist for different applications. The choice of a Data Compression technique depends on the type of data to be transmitted and its application. An energy-efficient Data Compression scheme

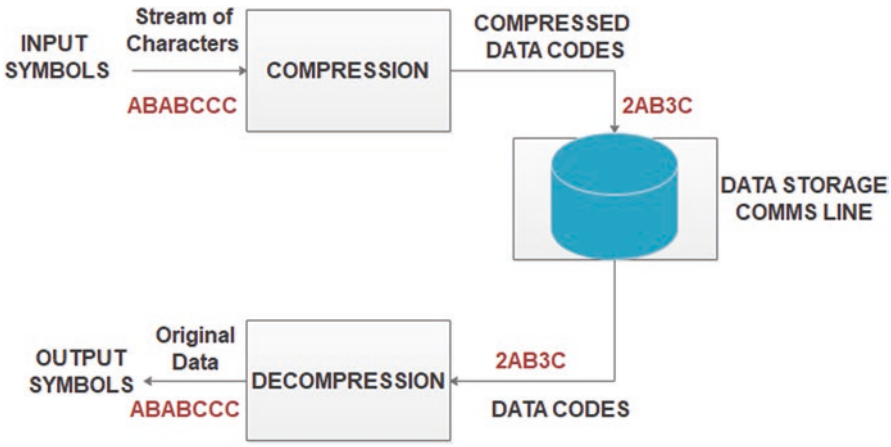


Fig. 1.3 A simple Data Compression model [16]

provides a desirable trade-off between energy used for transmission and energy used for processing information [2]. The technique must not be complicated and should require fewer resources for compression of data.

1.6 Summary

It is anticipated that combining Energy Harvesting, Communications Techniques, and Data Compression will yield high results that will help to improve the network lifetime of *Internet of Things sensor networks*. Since these networks are suitable for large-scale deployments, large quantities of data are handled during transmission from source to sink. This also includes data that is irrelevant and may add to the energy that is wasted during the transmission of data that eventually results in reducing the network lifetime. The need for a robust and energy-efficient technique to minimize data before transmission and transmit the reduced data along optimal transmission paths motivated this book. Since wireless sensor networks are deployed in larger volumes through several tiny elements called sensor nodes that are powered by small non-rechargeable batteries, the networks are faced with a number of limitations that pose a threat to the network lifetime. The advancement of the traditional *WSNs* gave birth to the *Internet of Things sensor networks*. In general, the *Green Internet of Things Sensor Networks (GIoTSNs)* combine four critical components: "Energy Harvesting," "Internet," "Things," and "Sensor Networks." The simple explanation to this is, "Any device having the capability and is compatible to connect to the Internet will come under Things." Things can be referred to as "Smart Devices," "Sensors," and "Miscellaneous Objects." The nodes in the network interact with one another and can access anything, anytime, and from anywhere in the system (Network) [4]. The network will survive for a longer time as it

is "Green," meaning that it can self-harvest energy to sustain the nodes in the network. Hence, the batteries carried by the nodes in the network will not deplete as they usually do.

References

1. J.G. Kolo, S.A. Shanmugam, D.W.G. Lim, L.M. Ang, Fast and efficient lossless adaptive compression scheme for wireless sensor networks. Comput. Electr. Eng. **41**(C), 275–287 (2015). https://doi.org/10.1016/j.compeleceng.2014.06.008
2. J.G. Kolo, S.A. Shanmugam, D.W.G. Lim, L.M. Ang, K.P. Seng, An adaptive lossless data compression scheme for wireless sensor networks. J. Sen. **2012** (2012). https://doi.org/10.1155/2012/539638
3. J. Uthayakumar, T. Vengattaraman, P. Dhavachelvan, A new lossless neighborhood indexing sequence (NIS) algorithm for data compression in wireless sensor networks. Ad Hoc Netw. **83**, 149–157 (2019). https://doi.org/10.1016/j.adhoc.2018.09.009
4. A. Solanki, A. Nayyar, Green Internet of Things (G-IoT): ICT technologies, principles, applications, projects, and challenges. Comput. Sci., 379–405 (2019). https://doi.org/10.4018/978-1-5225-7432-3.ch021
5. K.K. Patel, S.M. Patel, Internet of Things-IoT: Definitions, characteristics, architecture, enabling technologies, application & future challenges. Int. J. Eng. Sci. Comput. **6**(5), 6122–6123 (2016). https://doi.org/10.4010/2016.1482
6. M.T. Lazarescu, Wireless sensor networks for the Internet of Things: Barriers and synergies, in *Components and Services for IoT Platforms*, (2017). https://doi.org/10.1007/978-3-319-42304-3_9
7. L.K. Ketshabetswe, A.M. Zungeru, M. Mangwala, J.M. Chuma, B. Sigweni, Communication protocols for wireless sensor networks: A survey and comparison. Heliyon **5**(5), e01591 (2019). https://doi.org/10.1016/j.heliyon.2019.e01591
8. T. Nottingham, N.E. User, School of electrical and electronic engineering energy-efficient routing algorithms based on swarm intelligence for wireless sensor networks, Adamu Murtala Zungeru, B. Eng., M.Sc. Thesis submitted to the University of Nottingham for the degree of Doc, (2013).
9. F. Al-Turjman, A. Kamal, M.H. Rehmani, A. Radwan, A.-S. Pathan, The Green Internet of Things (G-IoT). Hindawi Wirel. Commun. Mobile Comput. **2019**, 1–2 (2019). https://doi.org/10.1155/2019/6059343
10. S.H. Alsamhi, O. Ma, S. Ansari and Q. Meng, Greening Internet of Things for smart everythings with a green environment life: A survey and future prospects. 1–3 (2018). [Online] Available: https://arxiv.org/ftp/arxiv/papers/1805/1805.00844.pdf
11. R.S. Lakshmi, RF energy harvesting for wireless devices. Int. J. Eng. Res. **11**(4), 39–52 (2015). [Online]. Available: www.ijerd.com.
12. International Electrotechnical Commission, Internet of Things: Wireless sensor networks. 20–21 (2014). [Online] Available: https://www.ipeea.org.
13. S.O. Olatinwo, T.H. Joubert, Energy efficient solutions in wireless sensor systems for water quality monitoring: A review. IEEE Sens. J. **19**(5), 1596–1625 (2019). https://doi.org/10.1109/JSEN.2018.2882424
14. K.S. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, W. Heinzelman, Energy-harvesting wireless sensor networks. ACM Trans. Sens. Netw. **14**(2), 2–3 (2018). https://doi.org/10.1145/3183338
15. N. Shabbir, S.R. Hassan, Routing protocols for wireless sensor networks (WSNs). Wirel. Sens. Netw. Insights Innov. (2017). https://doi.org/10.5772/intechopen.70208
16. T.A. Welch, Welch_1984_Technique_for.Pdf. IEEE Comp. **17**(6), 8–19 (1984)

Chapter 2
Applications and Communication Technologies in IoT Sensor Networks

Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong

2.1 Introduction

The application of IoT sensor networks in everyday settings greatly influences the quality of people's lifestyle and transforms conventional products and services into innovative solutions. IoT sensor networks enable physical objects to collect information from their environment, to share information, and to make intelligent decisions on the information where human decision making may be difficult. IoT sensor networks have gained popularity in recent years in real-time applications such as smart health, agriculture, Finance, industries, cities, and homes. IoT sensor networks involve heterogeneous objects communicating with each other locally or over the Internet globally. IoT sensor nodes communicate through wireless technologies to transform the traditional operation of objects to smart operation of those objects. The devices in the Internet of Things Sensor Networks are wireless and mobile with the ability to connect to the Internet through various wireless Internet technologies such as RFID, Zig-Bee, Bluetooth, Wi-Fi, and 6LowPAN. Communication technologies in IoT sensor networks are characterized by low-power consumption, low

P. Phalaagae · A. M. Zungeru (✉) · B. Sigweni · J. M. Chuma
Faculty of Engineering and Technology, Botswana International University of Science and Technology, Palapye, Botswana
e-mail: zungerum@biust.ac.bw

T. Semong
Faculty of Sciences, Botswana International University of Science and Technology, Palapye, Botswana

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_2

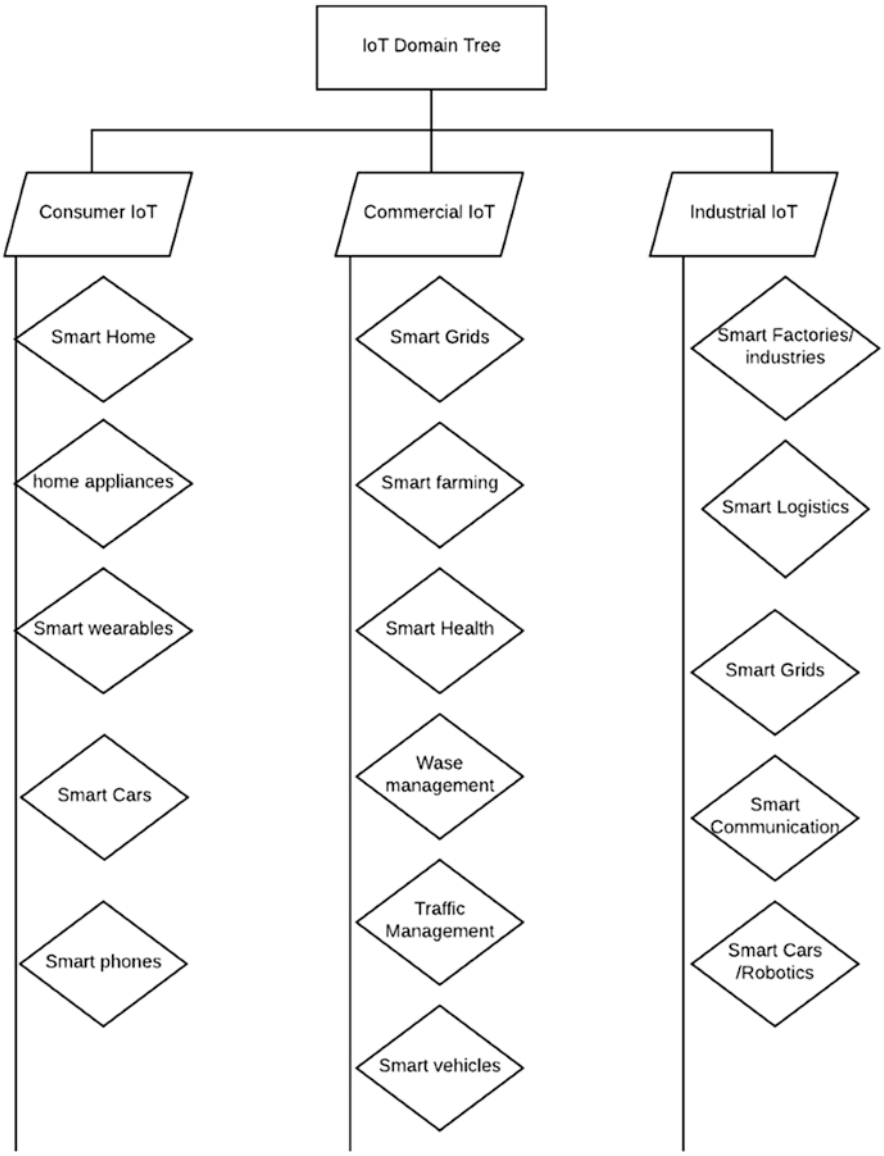


Fig 2.1 IoT sensor networks application domains

bandwidth usage, low computation power, and seamless communication of connected devices. IoT sensor networks applications and communication technologies are further explained in the next sections.

2.2 IoT Sensor Networks Applications

This chapter presents and discusses applications of wireless sensor networks. In this chapter, we characterize wireless sensor networks into three application domains: consumer, commercial, and Industrial IoT (IIoT). Figure 2.1 provides a summary of domains and their applications [20–22].

2.2.1 Consumer IoT

An example of a consumer IoT application is a smart home. Smart homes, popularly known as home automation, are residential areas using Internet-connected devices to enable remote monitoring and management of home appliances. The goal of smart homes is to provide security, comfort, and convenience to homeowners at the same time giving them control over their home appliances often through a smart home mobile application on their smart devices [23–25]. A smart home control appliances such as lighting, climate, appliances, entertainment, and security systems such as access controls and alarm in a home. The functional diagram of a smart home is shown in Fig. 2.2. Protocols such as Z-Wave and Zig-Bee enable communication

Fig. 2.2 Smart home

between smart home systems through mesh network technologies over short-range, low-power radio signals. A centralized master home automation controller (smart home hub) controls devices in a smart home. The smart home hub is a hardware device that interfaces other smart home devices with the ability to sense, process, and communicate data wirelessly. The smart hub integrates several applications into a single smart home application for purposes of remote control and management by homeowners. Smart home hubs include Google Home, Samsung Smart Things, Amazon Echo, and many others. Other smart home systems can be developed from scratch using prototyping boards such as Raspberry Pi or even purchased as bundled smart home kits. Artificial intelligence and machine learning are increasingly applied in smart homes to allow home automation applications to learn about their environment. This also offers personalized services to homeowners according to their preferences and patterns using applications that use virtual assistants such as voice-activated systems, e.g., Google Chrome. An example of a smart home scenario is a time-triggered event, for example, lowering blinds at a time every single day, or when the user's smartphone approaches the door, the smart lock will unlock or lock the house.

2.2.2 Commercial IoT

Smart cities are more than a trend, but the wave of the future as the world becomes more urban by combining smart technology initiatives across the cities. Smart cities are urban areas that revolutionize IoT by using sensors to collect data and use the data to manage assets and resources efficiently to achieve sustainable urbanization [26–28]. With the advent of Internet technologies, cities are digitally transformed to improve urban, environmental, financial, and social aspects of urban life. Recently, countries such as China, Singapore, the USA, India, and Australia have developed a growing interest in building smart cities to improve the quality of life for its citizens. In New York, surveillance cameras coupled with AI were installed after the tragic 9/11 attacks to protect public spaces through public–private partnerships (PPP) driven by cities and corporations without public review. In China, social profiling is used to monitor political and social behavior and to control access to services such as education, housing, and travel. Emerging technologies such as IoT automation and machine learning are critical drivers for smart city adoption. Smart city components include but not limited to smart manufacturing, smart governance [29], smart grids (energy/utilities) [30], smart transportation, smart farming, smart health, smart buildings, and smart citizens. Applications of these components include Waste Management, Traffic Management, Parking Management, Emergency Services, Smart vehicles (Infotainment, Diagnostics), Remote Management [31]. Smart city technologies are used to improve public safety with applications ranging from crime monitoring to warning systems using sensors for extreme events such as droughts, hurricanes, and floods. Intelligent motion sensors are used to conserve energy by dimming streetlights when there are no pedestrians or cars on the roadway. Smart grids are used to manage power outages and supply power on demand in cities. Despite the several advantages of smart cities, challenges such as integrating and

processing the big data resulting from a wide variety of data sources remains a challenge. Other challenges, as outlined in [32] that affect the design of smart city applications, include cost, smart network infrastructure, data privacy and security, population, advanced algorithms, and big data management [33, 34]. The urbanization of cities can be achieved through the exponential integration of technologies such as IoT, AI, blockchain, and virtual reality for sustainable smart cities.

2.2.3 Industrial IoT

Industrial IoT, also known as Industry 4.0, is a crucial element for factory automation integrating modern cloud computing, AI, and industries to create intelligent, self-optimizing industrial machinery and facilities [29, 35]. Industrial IoT (IIoT) uses modern sensor technology to optimize the operational efficiency of different types of equipment through remote monitoring and maintenance capabilities in the industrial sector of the economy [36]. IIoT takes advantage of data using smart machines to capture and perform real-time data analytics for better communication and faster and more accurate decisions to drive businesses. IIoT is applied in manufacturing specifically for quality control, predictive maintenance, supply chain traceability (asset management), and overall efficiency as well as sustainable energy management practices. Industrial IoT applications include factories, industries (logistics, oil and gas, mining, aviation), smart grids, smart communications, smart utilities, smart cities, smart cars, and robotics [37–40]. IIoT technologies such as predictive maintenance are used to identify potential issues in equipment before they wear and tear and enabling technicians to track and check the status of assets within the supply chain and perform preventative measures on the assets. IIoT evolved from the distributed control system (DCS) to cloud computing to refine and optimize process controls [41].

2.3 IoT Sensor Networks Communication Technologies

The adoption of IoT has birthed the convergence of communication technologies and standards to govern how devices connect and share information in the network [1–3]. Devices in the wireless IoT network are connected through various forms of Internet technologies to enable communication between connected nodes in a network. The IoT devices have different built-in sensors and communication interfaces for specific environments. Wireless technologies that support communication at different levels of the IoT architecture to support various modern applications in the IoT network are summarized in Table 2.1 [4–7]. Wireless IoT communication technologies have been classified in terms of transmission range or communication coverage, namely Wireless Body Area Networks (WBANs), Wireless Personal Area Networks (WPANs), Wireless Local Area Network (WLAN), and Wireless Wide Area Network (WWAN). This section will provide an overview of these state-of-

Table 2.1 Comparison of IoT sensor networks communication technologies

Classification by network type	WBAN			WPAN			WLAN			WWAN/LPWAN		
Characteristics	NFC	RFID	Bluetooth	Zig-Bee LR-WPAN	6LowPAN	Z-Wave	Wi-Fi	Sigfox	LoRa	WiMAX	Cellular Technologies	EnOcean
Standard	ISO/IEC 14443 A&B,JIS X-6319-4	RFID	IEEE 802.15.1	IEEE 802.15.4	IEEE 802.15.4	ITU-T	IEEE 802.11 a/b/g	Sigfox	LoRa WAN R1.0	IEEE 802.16	4G-LTE,2G-GSM,3G-UMTS, CDMA 2000, 5G	ISO/IEC 14543-3-1X
Transmission range	Short 0–10 cm 0-1m 10cm-1m	Short 1.5 m	Short 8–10 m	Short 10–20 m	Short 10–100 m	Short 30 m (indoor)-100 m (outdoor)	Short 20–100 m	Long 10 (Urban) km-50(Rural)km	Long <30 km	Long 50 km	Long Cellular Area 80 km	30 m
Data rate	424 Kbps	4 Mbps	1–24 Mbps	40–250 Kb/s	250 Kbps	40 Kbps	1 Mbps–6.75 Gbps	100 bps(UL) 600 bps (DL)	0.3–50 Kbps	1 Mbps–1 Gbps (Fixed)	2G: 50–100 kb/s 3G:200 kb/s 4G:0.1–1 Gb/s	125 Kb/s
Frequency band	13.56 MHz	125 kHz 13.56 MHz 902–928 MHz	2.4 GHz	868/915 MHz-2.4 GHz	868 MHz 915 MHz 2.4 GHz	868 MHz-908 MHz	5–60 GHz	868/902 MHz	868/900 MHz	2–66 GHz	865 MHz-2.4 GHz	1 GHz
Energy Consumption	Very low 50 mA	Ultra-low power	Medium BLE: Very Low <10 mW	Low <10 mW	Low Power 1–2 years battery lifetime	Very Low 2.5 mA	High >100 mW	10 Mw–100 Mw	Very Low	Medium	Medium 10–100 Mw	Low
Security	AES RSA	RC4	AES E0 Stream	AES	AES	AES-128	AES	Partially addressed	Symmetric key	AES	RC4	CRC
Cost	Low	Low	Low	Low	Low	Low	Medium	Medium	High	High	High	High
Applications	Payment access	Tracking, inventory	Audio Applications, wireless headset	Home, industry monitoring and control	Monitoring and control via Internet	Home monitoring and control	Audio, industrial, medical, remote connection	Street lighting, energy meters	Mining, smart cities	Connects cities, Internet access	Mobile devices, Internet	Building Automation Smart homes

the-art communication technologies in IoT sensor networks. A comparison of these communication technologies is presented in Table 2.1.

2.3.1 Wireless Body Area Networks (WBAN)

Wireless Body Area Networks (WBANs) involve technologies with the proximity of bodies of individuals. WBAN offers reliable wireless communication within the surround of the human body. The proximity technologies involve applications such as smart wearables employed in Near Field Communication (NFC) and Radio Frequency Identification (RFID). These technologies can interact with various technologies such as Bluetooth, Zig-Bee, and not limited to wireless sensor networks. WBAN is commonly used in health applications as smart wearables, implants, and remote patient monitoring also for entertainment applications, emergency services, and applications supporting real-time streaming such as voice, data, and video.

A. NFC
Near Field Communication is a technology that enables data transmission between objects within a short-range. NFC is like RFID; devices in this technology are assigned a tag for identification purposes. The tag can be read-only or rewritable and could later be altered by a device. NFC technology supports the connection, commission, and control of IoT devices in different environments and is widely adopted in mobile phones, payment systems, and industrial applications. Authors of [8] proposed an NFC m-ticketing prototype for urban transport systems intending to provide valuable real-time transportation information such as ticket and seat availability through mobile phones and smart cards to clients. Other services for m-ticketing include self-check-in and out of self-ticketing. Mobile ticketing in public transport has been widely adopted in public transport in the Netherlands, Japan, Korea using NFC technology, and the results have been confirmed by [7, 9, 10].

B. RFID
Radio-frequency identification (RFID) wirelessly uses electromagnetic fields to identify objects, collect data about them, and enter the data into the computer with little human interaction [11–13]. The transmission range for RFID is between 10 cm up to 200 m for long-range communication. The RFID system is composed of a tag which composes of an integrated circuit and an antenna that is responsible for relaying information to the RFID reader [14, 15]. The RFID reader then translates information collected from the reader in the form of radio waves into a meaningful form. The information collected is then sent to the computer through a communication interface. The RFID technology is less secure, consuming less power and does not need to be position precisely relative to the scanner, like in the case of barcodes. RFID is used in various applications such as inventory management, asset tracking, ID badging, and access control. RFID has, however, raised security issues due to its nature of reading personally linked information without any concerns. Several methods have been proposed

to address security and privacy issues by securing communication through tag/reader authentication or digital signatures. Other challenges include easy wear of the battery for RFID tags, RFID reader or tag collision, and lack of global standards for governing communication among IoT devices.

2.3.2 Wireless Personal Area Networks (WPANs)

Wireless Personal Area Networks (WPANs) define personal networks interconnecting devices centered on an individual's workspace. WPAN is also known as a short wireless distance network for its short distance coverage. A commonly used technology in WPAN includes Bluetooth, Z-Wave, Zig-Bee, and 6LowPAN for connecting personal user devices such as laptops, PDA, mobile phones, and peripherals [15].

A. Bluetooth
Bluetooth is a wireless technology based on the IEEE 802.15.1 standard for short-range communication. This technology covers a transmission range of about 10 m using a short wavelength of 5.5 GHz. Bluetooth is characterized by high energy consumption and less secure. Bluetooth 4.0, known as Bluetooth Low Energy, was introduced by Nokia using fast and low energy consumption. The technology applies to various operating systems such as Android, Windows phone, Linux, iOS. Today Bluetooth technology has evolved to the latest Bluetooth 5, which has four times the transmission of the latter 4.0 technology. Bluetooth supports the discovery and setup of services between devices such as mobile phones, tablets, and media players.

B. Z-Wave
Z-Wave is a low-power wireless IoT communication technology used for smart homes and small commercial firms. This technology transmits low data rate packets for data rate up to 30 kb/s, covering a transmission range of about 30 m Z-Wave. It follows a Mesh topology where there are a master device and slave nodes, which are low-cost devices-Wave apply to small messages IoT applications such as lighting and energy control [16]. Z-Wave architecture is presented in Fig. 2.3.

C. Zig-Bee
Zig-Bee is a low-power wireless network technology designed for low-power communication that can transmit data over long distances. This technology is used in an application that requires a low data rate, long battery life, and security features on the networked devices. Zig-Bee is widely used in smart homes and industrial control and monitoring applications [17, 18].

D. 6LowPAN
6LowPAN is a communication technology confirmed by the IETF working group that combines the latest version of Internet Protocol (IPV6) and Low-Power Wireless Personal Area Networks (Low PANs). This technology allows

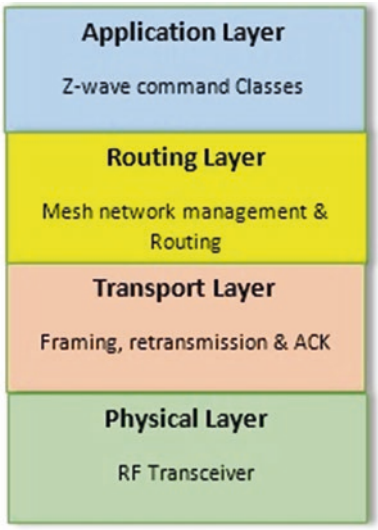


Fig. 2.3 The Z-Wave architecture

IoT low-power smart devices with limited processing capabilities to transmit information wirelessly using the Internet Protocol. This is a network layer encapsulation protocol designed for small networks that allow low power, low data rate, lossy networks to build routes and connect to the network and share routing information to resource-constrained nodes. The protocol supports crucial pairwise encryption, which is not very effective in protecting nodes from attacks. Application areas for 6LoWPAN include automation and entertainment applications at home. Thread is an advancement of 6LowPAN technology to enable home automation [14, 19].

2.3.3 Wireless Local Area Network (WLAN)

Wireless Local Area Network (WLAN), commonly known as Wireless LAN, is a short distance network connection allowing a user to connect to a local area network through a wireless radio connection or Bluetooth technology instead of using physical cables. The Wireless LAN is limited to a small geographical area such as a home or office building. WLAN consists of Access points that serve the purpose of transmitting and receiving signals and clients, which include end devices such as IP phones, workstations, and personal computers.

A. Wi-Fi
Wireless Fidelity (Wi-Fi) is a wireless network technology using radio waves based on IEEE 802.11 standard to provide wireless high-speed Internet and network connections. This technology uses a non-wired technology using radio

frequency within an electromagnetic spectrum associated with a radio wave. Wi-Fi supports various applications and devices such as home networks, mobile phones, and remote connections [20, 21].

2.3.4 *Wireless Wide Area Network (WWAN)*

Wireless Wide Area Network (WWAN) is a wireless network technology that sends wireless signals beyond a single building. WWAN spans over a large geographical area through mobile and public networks. WWAN may be low power, and a low bit-rate wireless network commonly referred to as Low-Power Wide Area Network (LPWAN) to carry small packets of information between battery-operated sensors [22]. WWAN technologies include cellular, Sigfox, LoRa, and WiMAX.

A. Cellular Technologies
   Technologies behind smartphones enable or empower IoT innovation by connecting physical things to the Internet through mobile networks. Cellular networks can connect a smartphone to various applications such as Google Maps, Facebook, Email, home appliances, and connection to the Internet using technologies such as GSM, LTE, 3/4/5G [2]. Cellular technologies have a high throughput making them an excellent fit for long-distance communication.
B. Sigfox
   Sigfox is a low-power technology for IoT sensor networks, transferring small amounts of data over energy-constrained smart objects. Sigfox uses Ultra Narrow Band technology, which is designed to handle low data transfer rates of

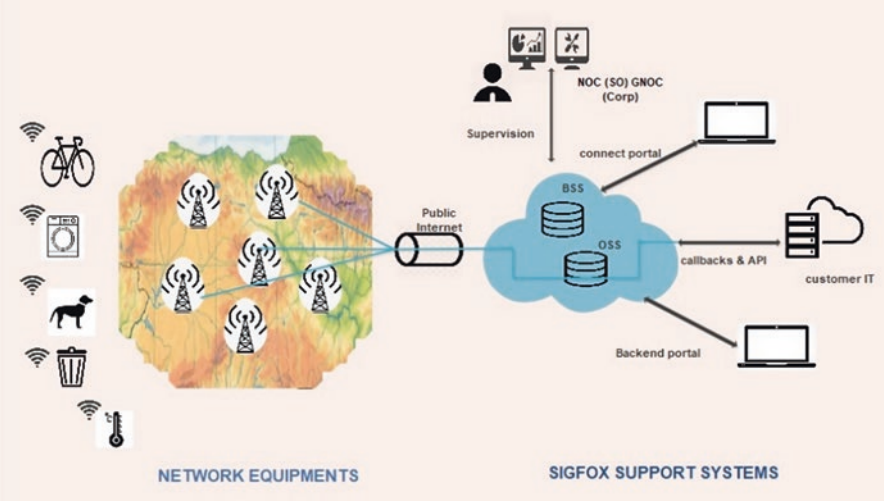


Fig. 2.4 The Sigfox architecture

10–1000 bits per second. The application of Sigfox is in smart meters, street lighting, and environment sensors [23–25]. Sigfox architecture is presented in Fig. 2.4.

C. LoRa

LoRa (which stands for long-range) is a de facto technology for wireless IoT communication using wireless radio frequency over a long-range, low-power wireless technology. LoRa technology easily plugs into existing infrastructure enabling secure, low-cost applications through private, public, or hybrid networks. This technology spans over a transmission range of about 30 km at a transmission rate of 0.3–50 kbps. The interface for LoRa has been designed to allow shallow signals to be received at low-power significant transmission ranges. LoRa is used across IoT applications ranging from smart homes, buildings, agriculture, industries, and cities. LoRa is not suitable for real-time applications and only favorable to applications that can tolerate delays due to its nature of being limited to the duty cycle. Applications of LoRa technology include smart metering, inventory tracking, monitoring, utility applications, and automotive industry [26, 27]. The architecture for LoRa technology is presented in Fig. 2.5.

D. WiMAX

Worldwide Interoperability for Microwave Access (WiMAX) is a long-range wireless networking technology designed for mobile and fixed connections based on the IEEE 802.16 standard. WiMAX was proposed as an alternative to

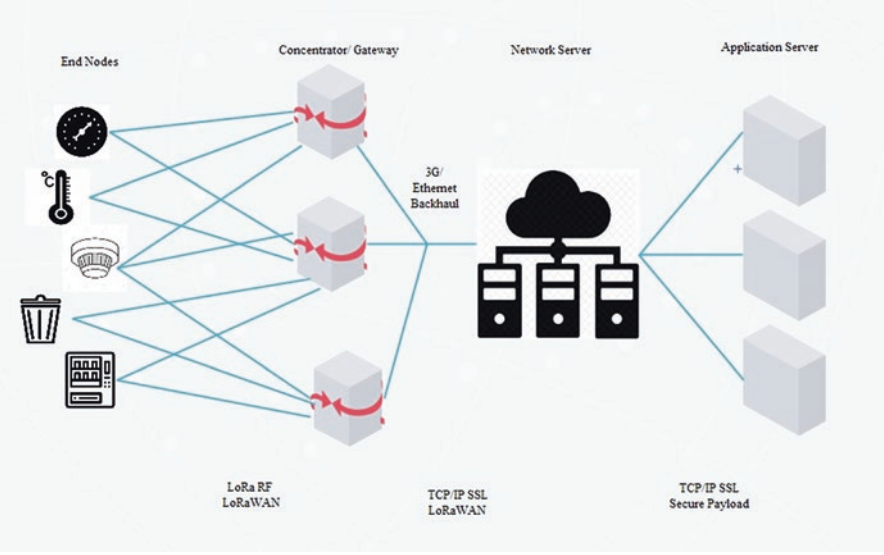


Fig. 2.5 LoRa technology architecture

cable or DSL Internet communication, which is expensive to implement. This technology is flexible, offering high-speed broadband service, and easy to implement at a low cost compared to a standard DSL connection [1]. WiMAX supports voice and video transmissions at transmission rates ranging from 1 Mbps to 1 Gbps for fixed stations. WiMAX supports several network models, such as transferring data over an Internet Service Provider network, fixed wireless broadband Internet access, mobile Internet access, and cable Internet access for remote users [28, 29]. A WiMAX System consists of two components; A WiMAX tower that connects directly to the Internet using high bandwidth and a WiMAX receiver in the form of a small box or card or could be built into a laptop. This technology uses two forms of wireless connections; the non-line-of-sight link where an antenna that sits on a computer connects to a tower transferring data at low frequencies ranging from 2 to 11 GHz and the line-of-sight connection where the fixed dish antenna points straight at the WiMAX tower reaching high data transfer frequencies of 66 Ghz. WiMAX application includes Internet access to individuals and cities through a wide range of devices.

E. EnOcean

The EnOcean wireless standard (ISO/IEC 14543-3-1X) is an emerging technology for wireless sensor networks with ultra-low power consumption [2, 27]. The wireless sensor network uses energy harvesting technology to draw energy from their surroundings. The wireless technology is optimized for use in buildings with a radio range of 30 m indoors. The architecture for the EnOcean technology is presented in the figure below, as specified by ISO & IEC. Self-powered wireless switches and sensors for building automation play a key role in digitization and enabling innovative services by providing reliable sensor data. Wireless sensors collect data without the need of a battery using the energy harvesting principle. The architecture of EnOcean technology is presented in Fig. 2.6.

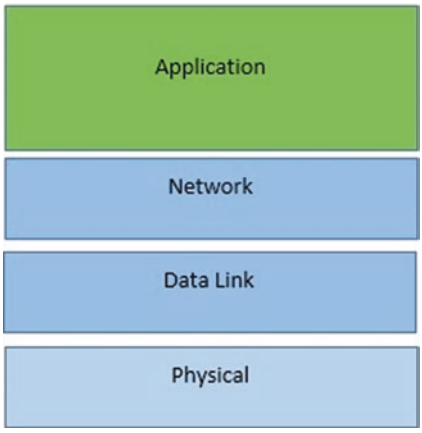


Fig. 2.6 The architecture for EnOcean technology

2.4 Summary

In conclusion, IoT sensor networks in the last few years have evolved from just interconnected computers or nodes to linking vast areas of the economy ranging from personal use to healthcare, supply chain, logistics, transport, infrastructure, enterprises, communities, and cities. IoT has not only transformed lives but has also affected everyday settings by transforming conventional products and services into innovative solutions such as smart cities, smart vehicles, smart industries, fleet tracking, and air pollution monitoring, to mention a few. It has been forecasted that by the year 2020, 20.4 billion IoT devices will be deployed across the world. As the IoT adoption grows rapidly, there is an increasing demand for heterogeneous devices to connect to the Internet through low-power communication technologies such as Low-Power Wide Area Networks (LPWANs) through various mediums such as LoRa, Sigfox, WiMAX, and EnOcean, etc. Other types of communication technologies are Wireless Body Area Networks, which are within proximity of a person or object such as NFC and RFID; WPAN including Bluetooth, Z-Wave, ZigBee, and 6LowPAN for connecting personal user devices. However, the selection of an appropriate communication protocol depends on the characteristics and requirements of the network, such as communication range, data rate requirement, energy consumption, frequency band cost, and the nature of the application. With an increase in the use of IoT devices, several IoT vulnerabilities are increased. The next chapter provides insight into IoT wireless sensor networks security issues and challenges.

References

1. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4) (2015). https://doi.org/10.1109/COMST.2015.2444095
2. S. Mittal et al., Secure routing in IoT networks with SISLOF. J. Netw. Comput. Appl. **4**(1), 1–6 (2018). https://doi.org/10.1016/j.jnca.2017.08.006
3. B.N. Silva, M. Khan, K. Han, Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. IETE Tech. Rev. **4602**, 1–16 (2017). https://doi.org/10.1080/02564602.2016.1276416
4. B.H. Çorak, F.Y. Okay, M. Güzel, Ş. Murt, S. Ozdemir, Comparative analysis of IoT communication protocols. in *2018 International Symposium on Networks, Computers and Communications*. ISNCC 2018 2018, Doi: https://doi.org/10.1109/ISNCC.2018.8530963
5. A.B.A. Rahman, Comparison of Internet of Things (IoT) data link protocols. Comp. Internet Things 1–21 (2015) [Online]. Available: http://www.cse.wustl.edu/~jain/cse570-15/index.html.
6. K. Mekki, E. Bajic, F. Chaxel, F. Meyer, A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express **5**(1), 1–7 (2019). https://doi.org/10.1016/j.icte.2017.12.005
7. S. Mukherjee, G.P. Biswas, Networking for IoT and applications using existing communication technology. Egypt. Informatics J. **19**(2), 107–127 (2018). https://doi.org/10.1016/j.eij.2017.11.002

8. C. Coelho, D. Coelho, An IoT smart home architecture for long-term care of people with special needs, in *2015 IEEE 2nd World Forum Internet Things*, (2015), pp. 626–627. https://doi.org/10.1109/WF-IoT.2015.7389126
9. K.E. Psannis, S. Xinogalos, A. Sifaleras, Convergence of internet of things and mobile cloud computing. Syst. Sci. Control Eng. **2**(1), 476–483 (2014). https://doi.org/10.1080/21642583.2014.913213
10. A. Triantafyllou, P. Sarigiannidis, T.D. Lagkas, Network protocols, schemes, and mechanisms for Internet of Things (IoT): Features, open challenges, and trends. Wirel. Commun. Mob. Comput. **2018**, 1–24 (2018). https://doi.org/10.1155/2018/5349894
11. 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. Tutorials **17**(3), 1294–1312 (2015). https://doi.org/10.1109/COMST.2015.2388550
12. S. Al-Sarawi, M. Anbar, K. Alieyan, M. Alzubaidi, Internet of Things (IoT) communication protocols: Review. in *ICIT 2017 – 8th Int. Conf. Inf. Technol. Proc.*, pp. 685–690 (2017). Doi: https://doi.org/10.1109/ICITECH.2017.8079928.
13. Z. Wang, X. Qin, B. Liu, An energy-efficient clustering routing algorithm for WSN-assisted IoT. *IEEE Wirel. Commun. Netw. Conf. WCNC* **2018**(April), 1–6 (2018) Doi: https://doi.org/10.1109/WCNC.2018.8377171.
14. M. Mallick, P. Kodeswaran, S. Sen, R. Kokku, N. Ganguly, TSFS: An integrated approach for event segmentation and ADL detection in IoT enabled smarthomes. *IEEE Trans. Mob. Comput.* **14**(8) (2018). Doi: https://doi.org/10.1109/TMC.2018.2880206.
15. X. Jia, Q. Feng, T. Fan, Q. Lei, *RFID technology and its applications in Internet of Things (IOT)*. July, (2018). doi: https://doi.org/10.1109/CECNet.2012.6201508.
16. L. Nobrega, A. Tavares, A. Cardoso, P. Goncalves, Animal monitoring based on IoT technologies. in *2018 IoT Vert. Top. Summit Agric. – Tuscany, IOT Tuscany* **2018**, pp. 1–5 (2018). doi: https://doi.org/10.1109/IOT-TUSCANY.2018.8373045.
17. A. Ali, G. A. Shah, M. O. Farooq, U. Ghani, Technologies and challenges in developing Machine-to-Machine applications: A survey. J. Netw. Comput. Appl. **83**(September 2016), 24–139 (2017). doi: https://doi.org/10.1016/j.jnca.2017.02.002.
18. M. Tao, X. Hong, C. Qu, J. Zhang, W. Wei, Fast access for ZigBee-enabled IoT devices using Raspberry Pi. in *2018 Chinese Control Decis. Conf.*, pp. 4281–4285 (2018). doi: https://doi.org/10.1109/CCDC.2018.8407868.
19. F.H.M.D.R. Esmail, Survey on IoT services: Classifications and applications. *Int. J. Sci. Res.* **4**(1), 2124–2127 (2015). [Online]. Available: https://www.ijsr.net/archive/v4i1/SUB15640.pdf.
20. X. Li, X. Lu, Design of a ZigBee wireless sensor network node for aquaculture monitoring. in *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 – Proc.*, pp. 2179–2182 (2017). doi: https://doi.org/10.1109/CompComm.2016.7925086.
21. I.U. Din, S. Hassan, B. Kim, K. Khan, *The Internet of Things: A review of enabled technologies and future* (December 2018). https://doi.org/10.1109/ACCESS.2018.2886601
22. J. Sanchez-Gomez, R. Sanchez-Iborra, A. Skarmeta, Transmission technologies comparison for IoT communications in smart-cities. in *2017 IEEE Glob. Commun. Conf. GLOBECOM 2017 – Proc.*, **2018**(January), pp. 1–6 (2018). doi: https://doi.org/10.1109/GLOCOM.2017.8254530.
23. M. Radovan, B. Golub, Trends in IoT security. in *2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 – Proc.* no. October, pp. 1302–1308 (2017). doi: https://doi.org/10.23919/MIPRO.2017.7973624.
24. M. Lauridsen, H. Nguyen, B. Vejlgaard, I. Z. Kovacs, P. Mogensen, and M. Sorensen, Coverage comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km area. *IEEE Veh. Technol. Conf.* **2017**(June), 2–6 (2017). doi: https://doi.org/10.1109/VTCSpring.2017.8108182.
25. T. Janssen, M. Aernouts, R. Berkvens, M. Weyn, Outdoor fingerprinting localization using Sigfox. in *IPIN 2018 – 9th Int. Conf. Indoor Position. Indoor Navig.* no. September 2018, pp. 1–6 (2018). doi: https://doi.org/10.1109/IPIN.2018.8533826.

26. M. Elkhodr, S. Shahrestani, H. Cheung, Emerging wireless technologies in the Internet of Things: A comparative study. Int. J. Wirel. Mob. Netw. **8**(5), 67–82 (2016). https://doi.org/10.5121/ijwmn.2016.8505
27. M. Centenaro, L. Vangelista, A. Zanella, M. Zorzi, Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. IEEE Wirel. Commun. **23**(5), 60–67 (2016). https://doi.org/10.1109/MWC.2016.7721743
28. S. Kraijak and P. Tuwanut, A survey on Internet of Things; architecture, protocols, possible applications, security, privacy, real world implementation & future trends. in *2015 IEEE 16th Int. Conf. Commun. Technol.* pp. 26–31 (2015). doi: https://doi.org/10.1109/ICCT.2015.7399787.
29. I.R. Tebepah, WiMAX for online service transmission. **7**(3), 55–62 (2017). doi: https://doi.org/10.5923/j.ijnc.20170703.02.
30. S. Nisha, M. Farik, RSA public key cryptography algorithm-A review. Int. J. Sci. Technol. Res. **6**(July), 7 (2017). [Online]. Available: www.ijstr.org.
31. M.A. Ferrag, L.A. Maglaras, H. Janicke, J. Jiang, L. Shu, Authentication protocols for Internet of Things: A comprehensive survey. Secur. Commun. Netw. **2017**, (2017). doi: https://doi.org/10.1155/2017/6562953.
32. S. Madakam, R. Ramaswamy, S. Tripathi, Jcc_2015052516013923. J. Comput. Commun. May, 164–173 (2015). doi: https://doi.org/10.4236/jcc.2015.35021.
33. I. Ali, S. Sabir, and Z. Ullah, Internet of things security, device authentication and access control: A review. IJCSIS **14**(8), 456–466 (2019). [Online]. Available: http://arxiv.org/abs/1901.07309.
34. A. Tewari, B.B. Gupta, Security, privacy and trust of different layers in Internet-of-things (IoTs) framework. Futur. Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.04.027
35. J. Deogirikar and A. Vidhate, Security attacks in IoT: A survey. in *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, pp. 32–37, (2017). doi: https://doi.org/10.1109/I-SMAC.2017.8058363.
36. A.A.A. Ari et al., Enabling privacy and security in Cloud of things: Architecture, applications, security & privacy challenges. Appl. Comput. Informatics (2019). https://doi.org/10.1016/j.aci.2019.11.005
37. M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, R. Brooks, The sleep deprivation attack in sensor networks: Analysis and methods of defense. Int. J. Distrib. Sens. Networks **2**(3), 267–287 (2006)
38. V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access **7**, 82721–82743 (2019). https://doi.org/10.1109/ACCESS.2019.2924045
39. B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in Internet of things. J. Netw. Comput. Appl. **84**(2016), 25–37 (2017). https://doi.org/10.1016/j.jnca.2017.02.009
40. K. Chen et al., Internet-of-things security and vulnerabilities: Taxonomy, challenges, and practice. J. Hardw. Syst. Secur. **2**(2), 97–110 (2018). https://doi.org/10.1007/s41635-017-0029-7
41. X. Li, J. Xu, H.N. Dai, Q. Zhao, C.F. Cheang, Q. Wang, On modeling eavesdropping attacks in wireless networks. J. Comput. Sci. **11**, 196–204 (2015)

Chapter 3
Smart Internet of Things Devices and Applications Specific

Jwaone Gaboitaolelwe, Adamu Murtala Zungeru, Joseph M. Chuma, Nonofo Ditshego, and Caspar K. Lebekwe

3.1 Introduction

In Chap. 2, we discussed applications and communication technologies in IoT sensor networks. By combining the technologies discussed in Chap. 2 with data gathering, data processing, and decision making, a form of useful intelligence or smartness can be added to devices and systems and hence, in turn, make them into smart IoT devices and systems. Smart IoT devices can be simply described as devices with hardware and software that allow them to exhibit a form of intelligence through data gathering, data processing, and decision making with little to no human being involvement that is useful to an end-user. Almost any device can be made into a smart IoT device, even a simple day-to-day home appliances. Limitations of what can be made into a smart IoT device are based on the cost and practicality of the device. Examples of smart IoT devices that can be found in a building include smart light bulbs, smart kettles, smart switches, smart refrigerators, and many more.

This chapter explores and shows the design and implementation of smart IoT devices through an example problem. This is carried out through the solving of a specific application example concerning the problem of securing a building's main power supply against unauthorized use. Preliminary results and discussions of the work reported in this chapter are published in [1]. In this chapter, a description of the said problem is made. Following that, the solution and the steps taken to reach it are shown. Each section in this chapter shows the methods and procedures undertaken to design the smart IoT devices used to solve the given example problem. Section 3.2 covers the description of the example problem. This includes the problem definition, aim, and objectives. In Sect. 3.3, the solution design process and procedures are covered. This includes system structure, hardware design, and software design. Section 3.4 shows the results of the system implementation, and last is Sect. 3.5 of which is a summary of this chapter.

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_3

3.2 Description of the Example Problem

This section describes the example problem being considered.

3.2.1 Problem Definition

Electricity is an essential resource with growing demand and increasing cost, hence in both the private and public sectors, there is a desire to proactively save and use it wisely to save money. However, existing smart home/building systems do not have any security and access control measures to control the accessing and use of electricity from the sockets and switches in the buildings. The absence of security and access control for sockets and switches in a building leaves them vulnerable to misuse by unauthorized users, which can lead to electricity wastage or damage to electrical appliances and property. This is especially problematic for large buildings or buildings with shared spaces such as schools, hotels, and flats.

3.2.2 Aim

The aim is to design, implement, and test a wireless sensor network-based smart home switching system with two main functions. First, a secured switching system to introduce security and access control to a building's electricity supply and second, a model of energy harvesting and storage system to remove the operational costs of the smart sockets and switches from a home user electricity bill.

3.2.3 Objectives

The objectives of the work are as follows:

1. To design security (access control system) for a building's power supply, which adds a locking feature such that only authorized personnel can alter the power state of the smart sockets and switches in a building.
2. To design a model of an energy harvesting and storage system for the active electronic components, the circuitries and wireless communication for smart switches and sockets.
3. To implement and test the performance of the designed smart home/building system and energy harvesting system.

3.3 Description of the Example Solution

Based on the aim and objectives that have been stated, the design of the system can be broken down into two sections. These being secured smart home switching system and an energy harvesting system. This is shown in Fig 3.1. The secured smart home switching system has two main functions. Firstly, to provide basic smart home functionalities such as centralized local control and remote control. Secondly, it adds access control security to a building power supply, starting from the switch box up to the sockets and switches.

The energy harvesting system converts solar energy to electrical energy to power the electronics components and circuits of the secured smart home switching. In this chapter, we shall limit ourselves to the design of the secured smart home switching system and its sub-system. Energy harvesting aspects and considerations for a system will be discussed in Chap. 4 of the book.

3.3.1 Secured Smart Home Switching System

This section describes the design of the secured smart home switching system. The design of the system is divided into four components. Each component is considered a sub-system of the secured smart home switching system. As shown in Fig. 3.1, the secured switching system is divided into four sub-systems: (1) Smart hub; (2) Smart socket; (3) Smart switch; and (4) Smart switchboard.

The smart hub is the center of the secured smart home switching system. It provides centralized control and monitoring of a building's electricity by creating the system's network and acting as a gateway to bridge the internal wireless sensor network based on Wi-Fi wireless network technology to an external cellular network using a Global System for Mobile (GSM) capable Universal Serial Bus (USB) modem. Through interaction with the smart hub, an authorized user can remotely as well as local monitor, control, and lock the power states of smart switches, smart

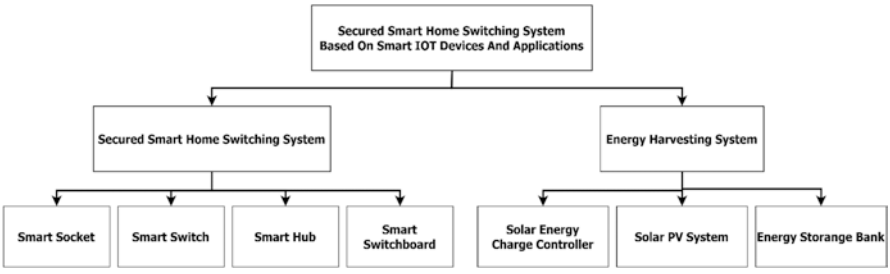


Fig. 3.1 Structure of secured smart home switching system based on smart IoT devices and applications

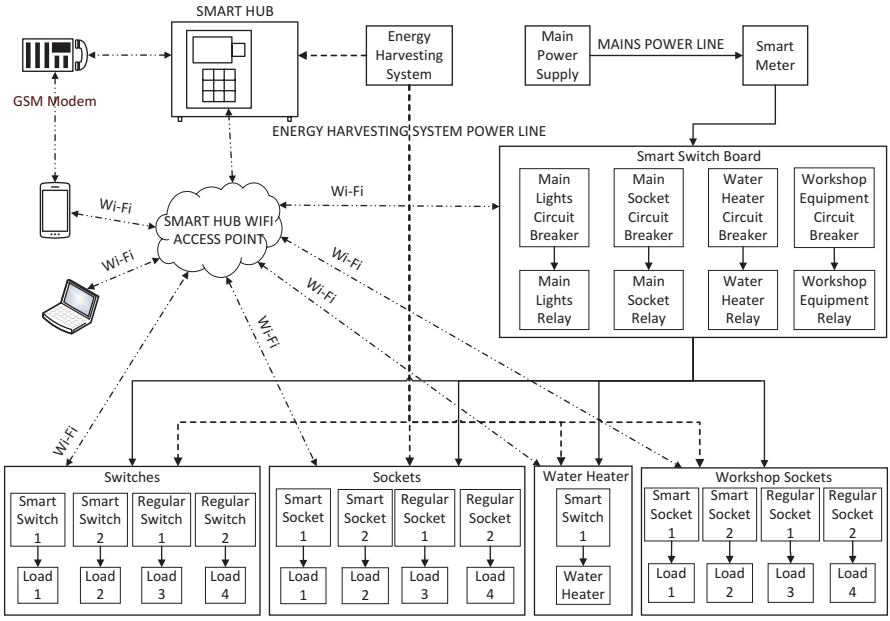


Fig. 3.2 Overview of secured smart home switching system

sockets, and smart switchboard in a building and hence monitor and control the state of appliances connected to a buildings power supply.

The smart switches, smart sockets, and smart switchboards are both sensors and actuators of the system. They are used to control the flow of AC electricity to appliances in a building based on the commands wirelessly communicated from the smart hub or based on sensed input from a user's physical interaction with the switches present on them.

Figure 3.2 shows an overview of the entire system. The system has two power sources; one is the energy harvesting system, and the other is the main powerline from a power grid into a building. The solid black lines represent Alternating current (AC) electricity from the power grid, and the thick dashed lines depict the Direct current (DC) electricity from the energy harvesting system. By implementing this system in homes and buildings, it can be used to monitor, control, and secure the electricity supply of a building.

3.3.2 Smart Hub Design

3.3.2.1 Smart Hub Functions

The role of the smart hub is to act as an interface between the user, smart switchboard, smart switches, and the smart socket. It is responsible for processing the data it receives and transmitting responses to the end nodes or displaying the relevant

information based on user requests. The requirements of the smart home hub are as follows:

1. Have two methods of user input. That is via internal keypad or wirelessly through Short Message Service (SMS).
2. Have two methods of relaying information to a user. That is via the internal display monitor or wirelessly through SMS service.
3. Wirelessly turn on and off the power state of the smart switches, smart sockets, and smart switchboard.
4. Wirelessly lock and unlock the ability to change the power state of the smart switches and smart sockets.

3.3.2.2 Smart Hub Hardware Selection

Processing Unit
In selecting the processing unit for the smart hub, a series of considerations are made. Since the smart hub acts as the core of the smart house system, it performs most of the duties that require processing, storing, and transmitting data from the smart sockets, switches, and switchboard devices. Based on the anticipated workload and leaving room for flexibility in terms of hardware interfaces and programming language options, a single-board (SBC) computer is a better suit as compared to microcontrollers. Out of the available SBC's, the selection is narrowed down to products form the Raspberry Pi foundation due to their large support community, easily accessible documentation, and good technical support. From the available SBC's the Raspberry Pi 3 Model B+ board computer [2] is selected.

Display Unit
One of the very important features of a display is to provide users with a good visual interface that is readable and understandable. The bigger the display, the easier it is to interact with a system, and the smaller the display is, the harder and longer it is to retrieve visual information. The price of displays is influenced by the technology used in its design and also the size. Hence bigger displays cost more. Touch screen displays act as both input and output devices. Their main disadvantage is their high cost compared to regular displays. The monochrome 20 × 4 LCD is selected to be the balance between screen size and readability. It is big enough to not strain the user's eyes when reading but also not too costly as compared to touch screens.

Input Method
A 4 × 4 matrix membrane is chosen as the input device for the smart hub. The keypad has both numerical and alphabetic characters, and hence through a thoughtful user interface design, a membrane keypad can be used to navigate through the different options and features provided by the smart hub. These options include actions such as entering pin codes for access control or interacting with the alphabet keys to perform some special tasks such as selecting options and navigating through menus in the smart hub.

GSM Module
To add SMS functionality to the smart hub, the Huawei E 303 USB modem capable of GSM/GPRS communication is used to create a link between a user's GSM/GPRS enabled phone and the smart hub processing unit. GSM is an architecture used for mobile communication in most countries. GPRS is a packet-oriented mobile data service that is an enhancement of GSM systems that enables a higher data transmission rate and connection to the Internet. A GSM/GPRS module is a chip or circuit that is used to establish communication between devices and a GSM or GPRS system. GSM/GPRS modems consist of GSM/GPRS modules assembled with a power supply circuit and communication interfaces (UART, RS 232, USB, and others).

3.3.2.3 Smart Hub Block Diagram

Figure 3.3 shows the block diagram of the smart hub. The smart hub blocks comprise of a Raspberry Pi single-board computer, a USB dongle GSM modem, a 4×4 membrane keypad, and a 20×4 monochrome LCD screen. The GSM modem creates a link between the smart hub and a user's cell phone. The LCD screen provides visual data for the user interface. The keypad is an input device for the system's user

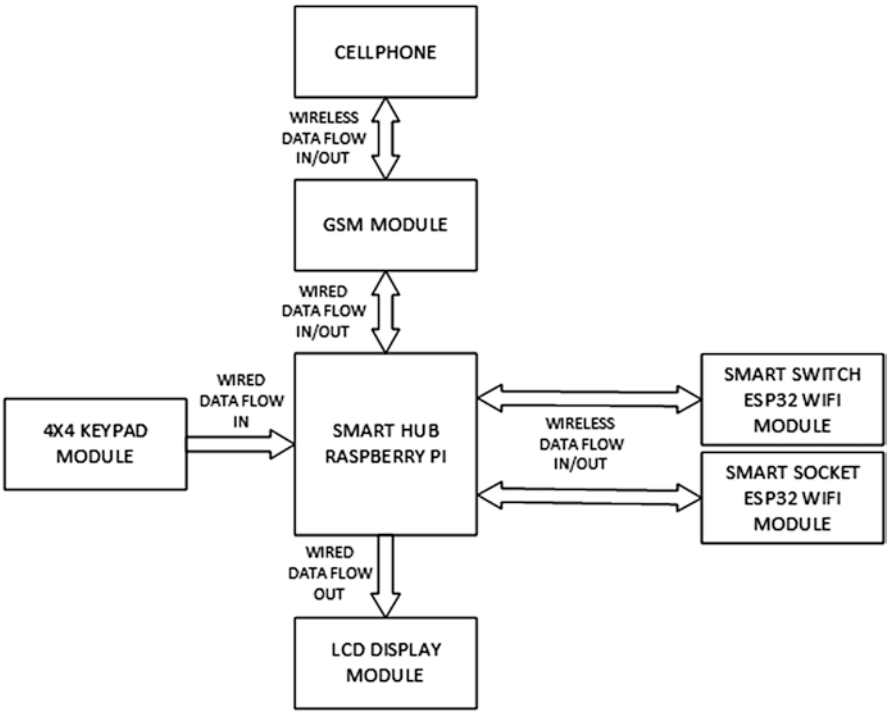


Fig. 3.3 Data flow between smart hub components

interface. The Raspberry Pi acts as both the smart home system control and network access point for the smart switchboard, smart sockets, and smart switches.

3.3.2.4 Smart Hub Circuit Schematic Design

Figure 3.4 shows the schematic diagram of the smart hub device made using circuit design and simulation software Proteus. In designing the smart hub circuit, all the pins of the keypad and signal pins of the LCD screen are connected to the Raspberry Pi GPIO pins. The GSM module is connected to the Raspberry Pi through the inbuilt USB port. Through software, instructions will be sent to write on the LCD screen and to read the state of the keypad buttons. The LCD screen contrast is controlled by the use of a potentiometer.

3.3.2.5 Smart Hub PCB Designs

Figure 3.5 shows the PCB design of the smart hub designed using PCB design software KiCAD. The PCB design of the smart hub is based on the use of 2.54 mm male pin headers and a female to female pin header cables to connect the Raspberry Pi to the PCB board. To connect the LCD screen and the keypad, 2.54 mm Female pin header is used.

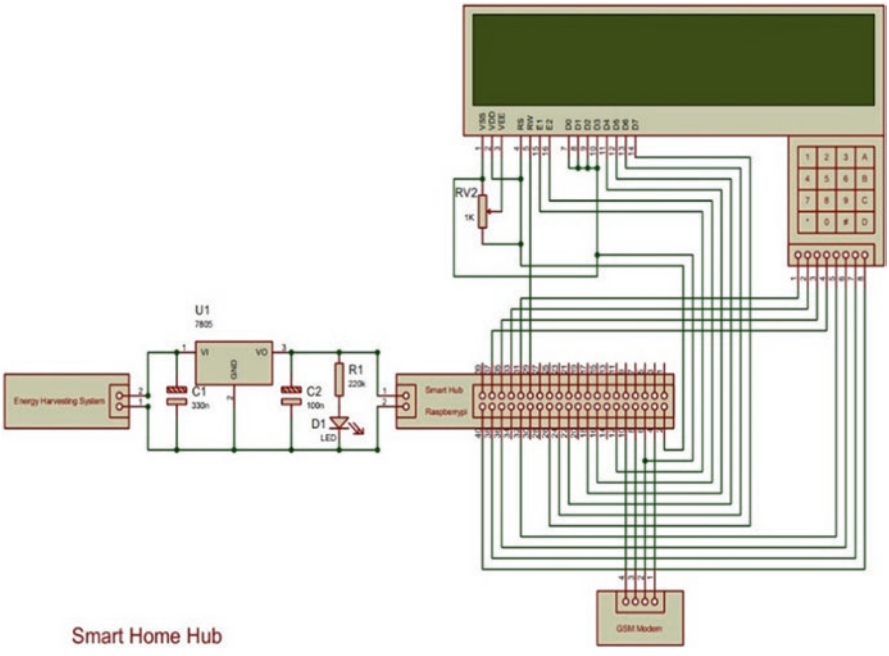


Fig. 3.4 Smart hub circuit diagram

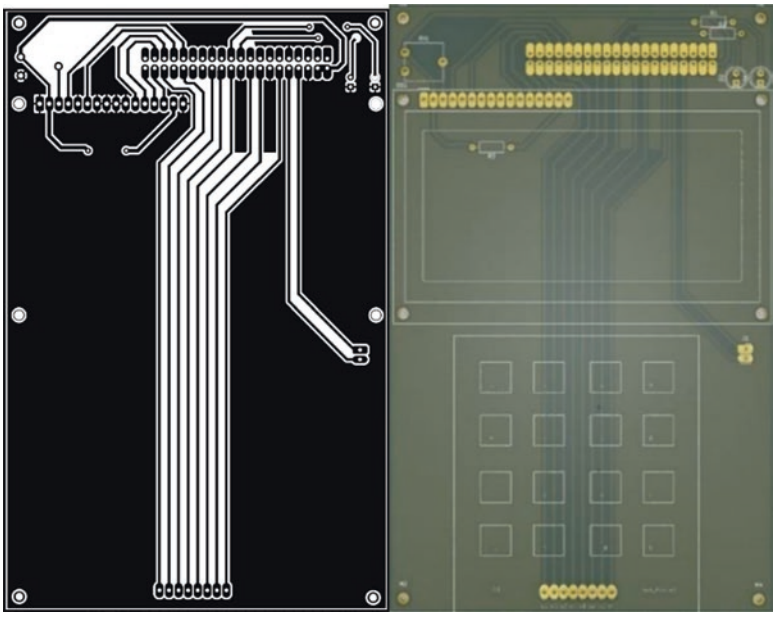


Fig. 3.5 Smart hub PCB designs

3.3.2.6 Smart Hub Software Design

Procedures and flowcharts 1–5 show the operation of the smart hub and are used in system software design. The procedures are used to aid in the software implementation of the smart home hub of which is written in a python script language.

Procedure 1: System-Locked *Description*: When the system is locked, a pin code request is displayed on the smart hub LCD screens. A user must enter the correct pin code using the inbuilt keypad to be granted access to the system. Entering a wrong pin code by a user leads to an error message to be displayed on the smart hub LCD screen. The smart hub LCD screen then displays a new pin code request for a user to retry. A correctly entered pin code leads to the display of access granted message to indicate successful unlocking of the smart hub. After the display of a successful unlock message, the smart hub proceeds to the system-unlocked procedure. The flowchart in Fig. 3.6 shows the performed actions.

Procedure 2: System-Unlocked *Description*: When the system is unlocked, a user is granted permission to look at the status of the available smart sockets and smart switches. Whenever there is any change caused by the manual switching off/on of an unlocked smart socket/switch in the network, the system refreshes the LCD screen and reflects the real-life state of the smart sockets and smart switches. If there is an SMS sent by a user to the smart hub, the system processes the message and sends commands to the affected smart sockets/switches and then updates the LCD only if the SMS command is valid. Apart from looking at the states of the smart

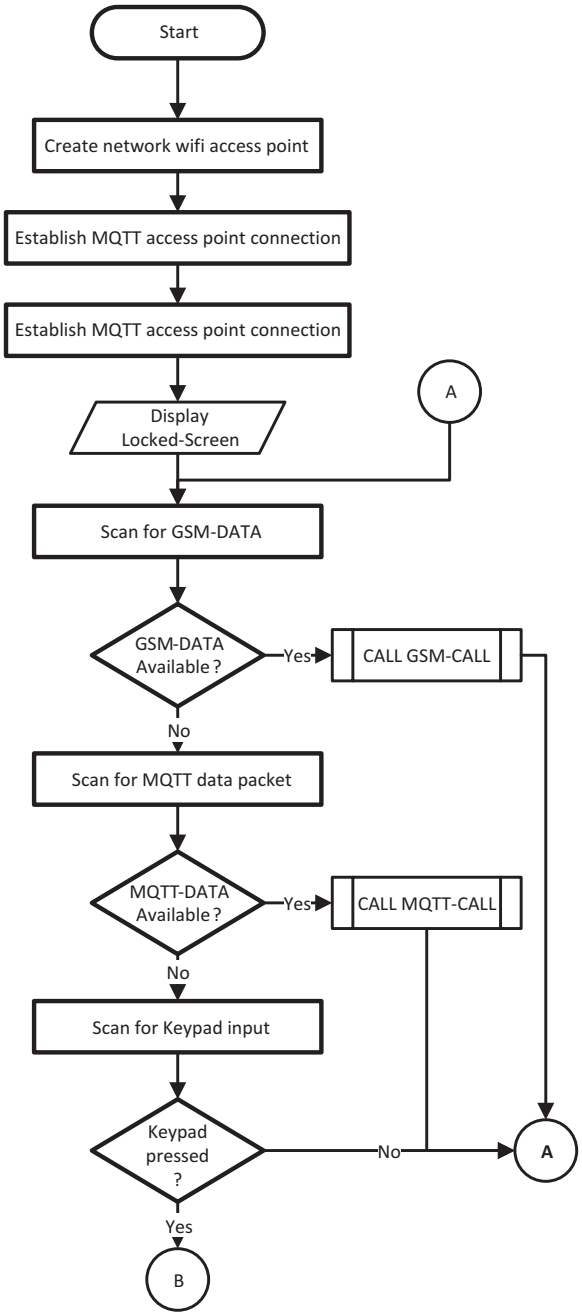


Fig. 3.6 (**a**, **b**) System locked flowchart

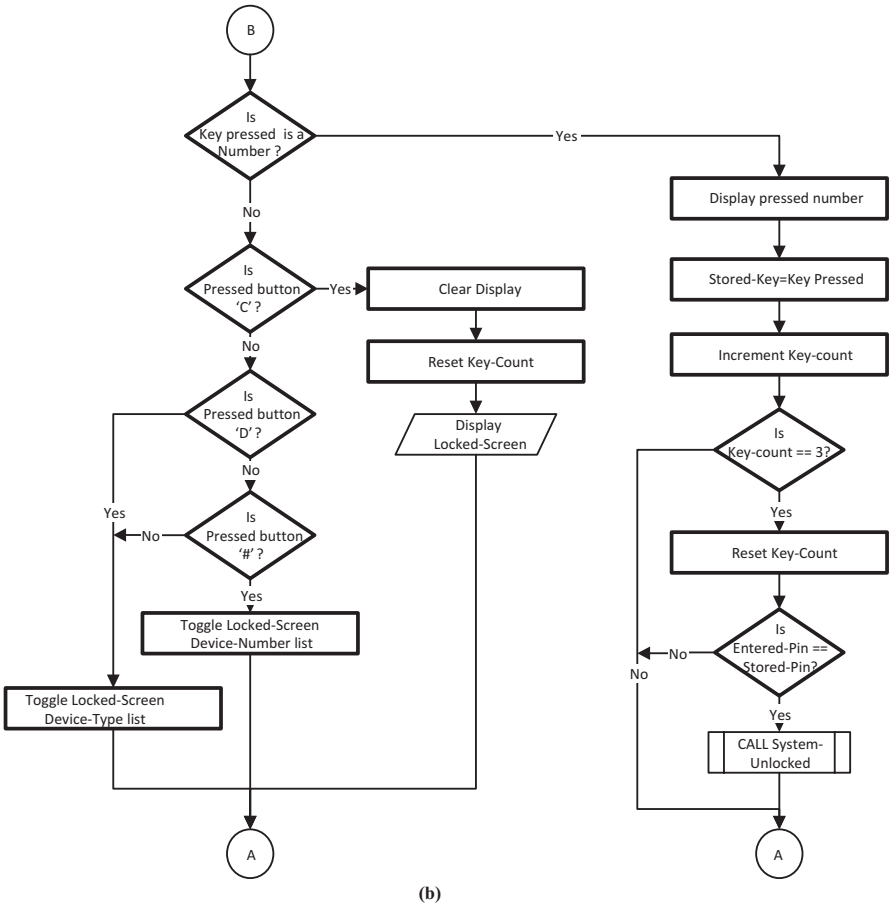
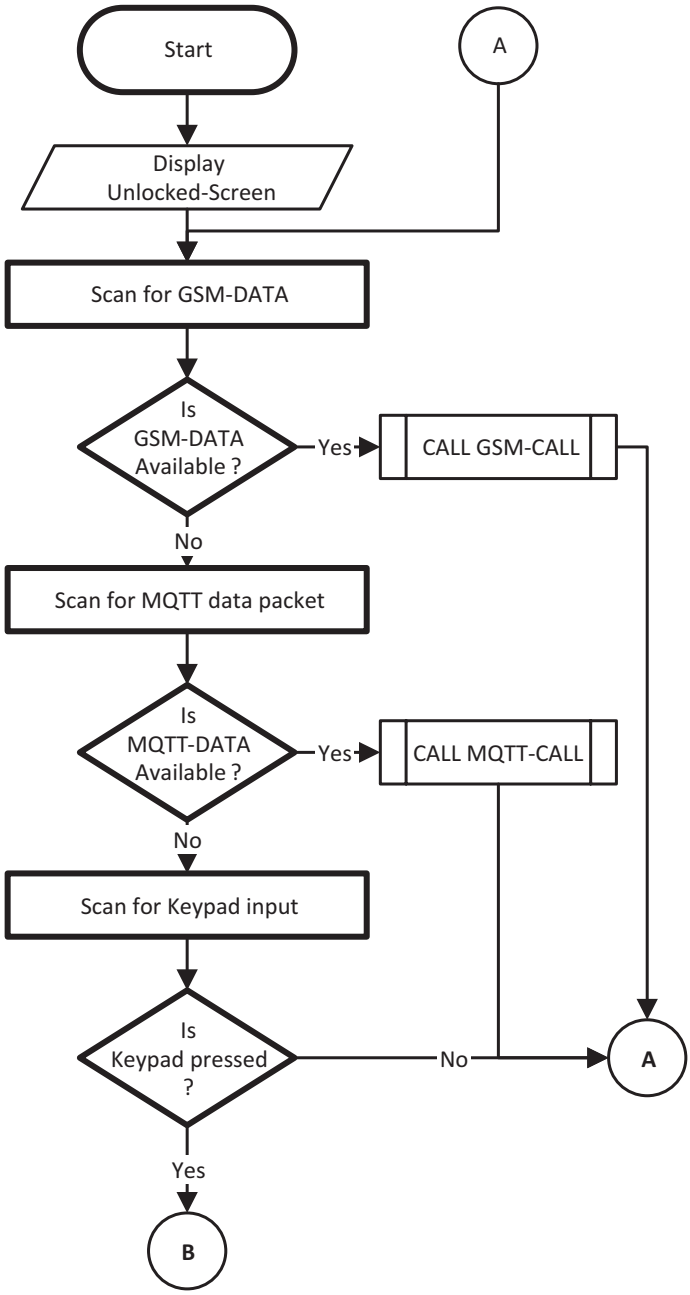


Fig. 3.6 (continued)

home switches/sockets, a user can also modify their states. This is accomplished by selecting an available smart switch or smart socket by entering the number associated with the device. Having selected the device, the Target-Device-Edit procedure is called, and the user can then modify the power state or lock state of the selected device. If the user is satisfied with the changes made and does not desire to make further changes, they can press on the keypad and go back to the system locked state. The flowchart in Fig. 3.7 shows the action.

Procedure 3: Target-Device-Edit *Description*: This is the procedure called upon by the system-unlocked procedure when a user wants to change the power state or lock state of smart home socket or smart home switch device (Smart Devices) power state or lock state. The procedure performs this by displaying a blinking cursor over the state (power state or lock state) that is being modified. Interacting with the keypad, it allows a user to toggle a blinking cursor between the two states (power state



(a)

Fig. 3.7 (**a**, **b**) System unlocked flowchart

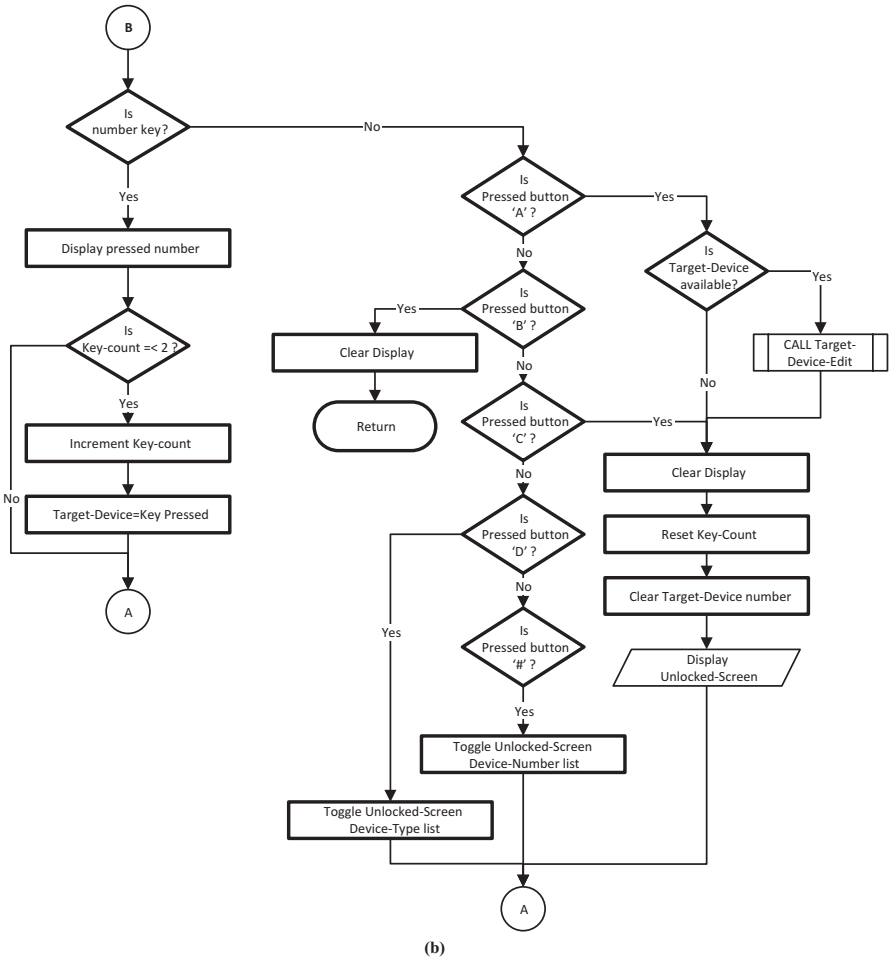


Fig. 3.7 (continued)

or lock state), toggle the value of the states (on/off or unlocked/locked) and return to the system-unlocked procedure. The flowchart in Fig. 3.8 shows the procedures.

Procedure 4: MQTT-Call *Description*: The MQTT-Call is a procedure that is needed whenever the smart hub receives an MQTT data packet. The procedure extracts information from the data packet and executes the instructions contained in the packet before returning to the function that called it. The flowchart in Fig. 3.9 shows the procedures.

Procedure 5: GSM-Call Description: The GSM-Call is a procedure that is called whenever the GSM module receives an SMS data packet. The procedure extracts information from the SMS and executes the instructions contained in the packet

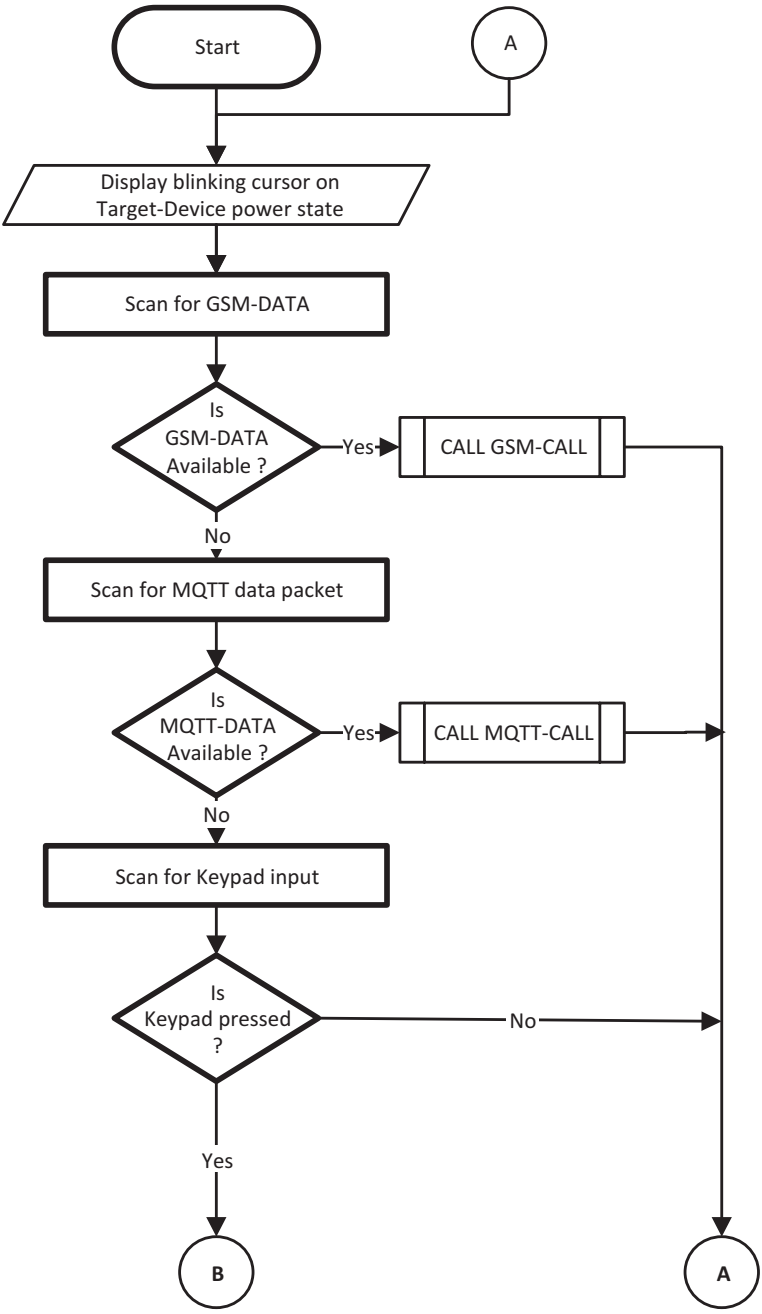
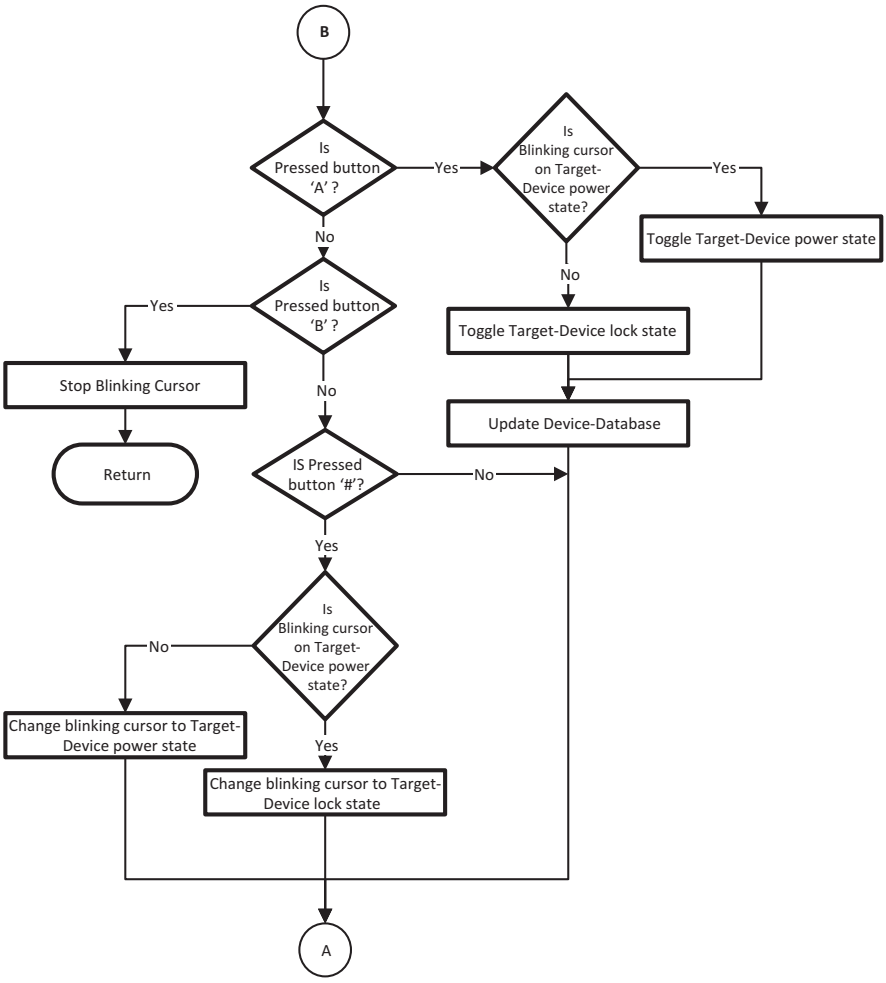


Fig. 3.8 (**a**, **b**) Target-device-edit flowchart



(b)

Fig. 3.8 (continued)

before returning to the function that called it. The flowchart in Fig. 3.10 shows the procedures.

3.3.2.7 Smart Hub Test

The smart hub is connected to a 5V lab power supply through the USB power port located on the Raspberry Pi. The smart switch, smart socket, and smart switchboard are connected, as explained in their tests. Once the smart hub is powered on, it establishes a Wi-Fi access point that the smart devices connect to. Table 3.1 shows the results of the smart hub tests.

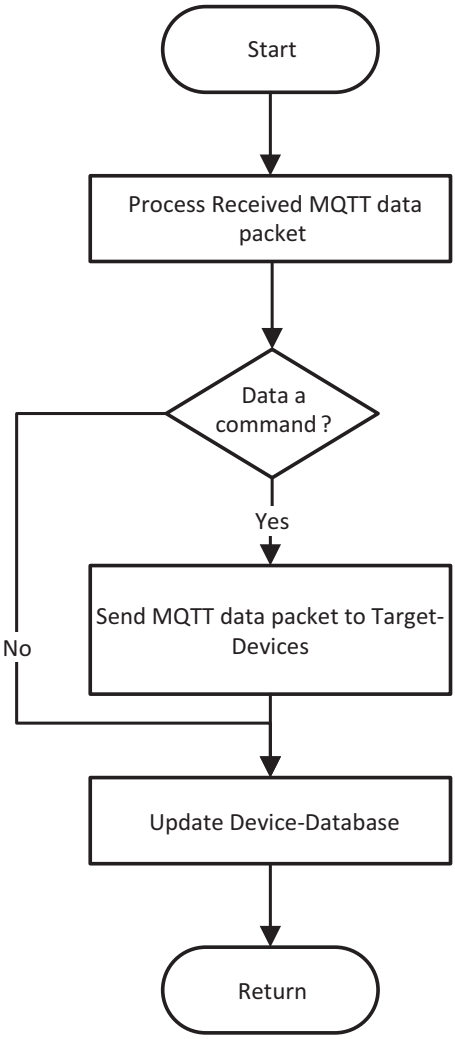


Fig. 3.9 MQTT call flowchart

3.3.3 Smart Switch And Smart Socket Design

3.3.3.1 Smart Switch Functions

The role of the smart switch is to act as a remote switch that can be turned on or off either by commands wirelessly sent from the smart hub or physically by a user. It has two modes of security; these are unlocked and locked.

The requirements of the smart switch are as follows:

1. To switch an appliance on or off based on commands from the smart hub

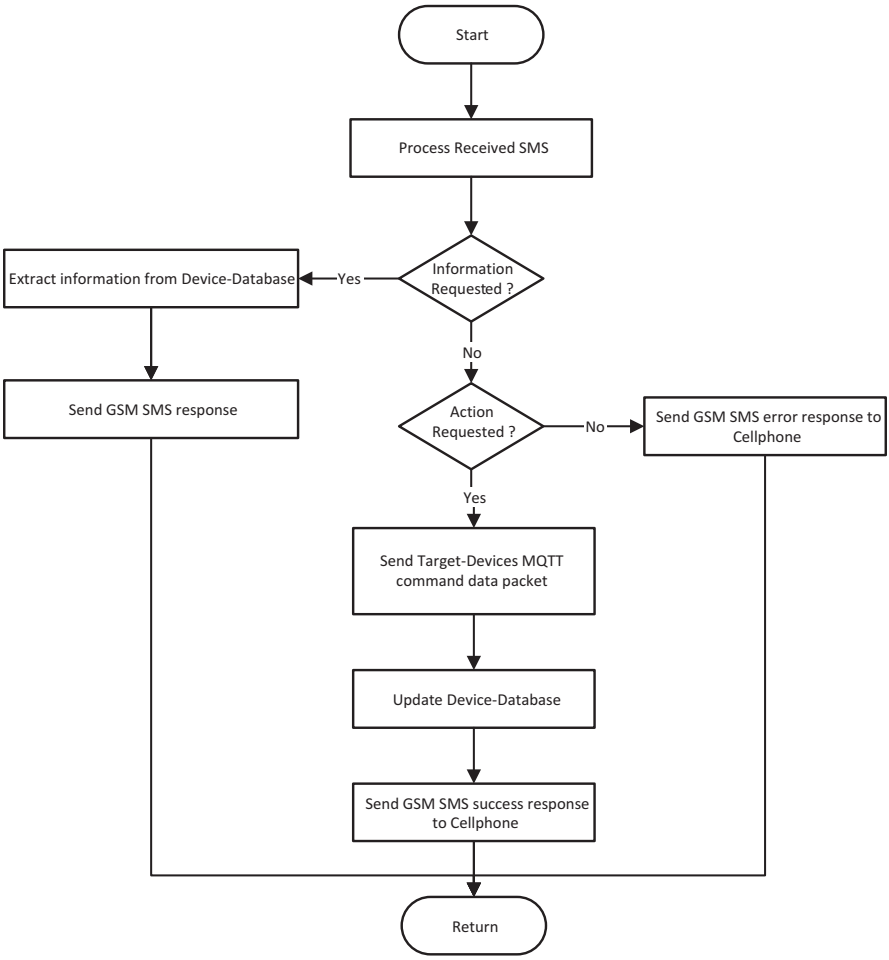


Fig. 3.10 GSM call flowchart

2. or the action of a user manually pressing on the inbuilt switch to change between the two power states (ON/OFF);
3. Send the state of the switch (ON/OFF) to the smart home hub;
4. Lock the state of the switch (on/off) so that it cannot be manually turned ON/OFF when commanded by the smart hub.

3.3.3.2 Smart Socket Functions

The role of the smart socket is to act as a remote socket that can be turned on or off either by commands wirelessly sent from the smart hub or physically by a user. It has two modes of security; these are unlocked and locked.

Table 3.1 Smart hub test results

Test	Current load state	Expected load state	Test results
Turn ON smart socket	OFF	ON	Positive
Turn OFF smart socket	ON	OFF	Positive
Turn ON smart switch	OFF	ON	Positive
Turn OFF smart switch	ON	OFF	Positive
Turn ON smart switchboard Relay 1-4	OFF	ON	Positive
Turn OFF smart switchboard Relay 1-4	ON	OFF	Positive
Lock smart socket	Unlocked	Locked	Positive
Unlock smart socket	Locked	Unlocked	Positive
Lock smart switch	Unlocked	Locked	Positive
Unlock smart switch	Locked	Unlocked	Positive
Receive SMS Command	N/A	N/A	Positive
Execute SMS command	N/A	N/A	Positive
Send SMS reply	N/A	N/A	Positive

The requirements of the smart socket are as follows:

1. To switch the circuit, it controls on or off based on commands from the smart hub or the action of a user manually pressing on the inbuilt switch to change between the two power states (On/Off);
2. Send the state of its switch (on/off) to the smart hub;
3. Lock the state of its switch (on/off) so that it cannot be manually turned on/off if commanded by the smart hub.

3.3.3.3 Smart Switch/Socket Hardware Selection

Processing Unit

Based on the requirements of the smart socket and smart switch, the Adafruit Huzzah32 breakout board based on the Espressif Systems ESP WROOM32 microcontroller [3] is selected as a suitable processing unit for the smart devices. The

advantages of an ESP32 based microcontroller over existing microcontroller technologies include its built-in wireless communication module capable of both Wi-Fi and Bluetooth standards of communication, a wide range of peripheral interfaces, and relatively fast CPU core that is useful for tasks such as data encryption.

Switching Unit
An electromagnetic relay is used to control the on or off status of electrical appliances. The dc coil is rated 5V and contacts rated for 230V at 15A, since the coil voltage rating is greater than the selected microcontroller's output voltage. The relay is paired with a switching transistor circuit for controlling its state.

3.3.3.4 Smart Switch/Socket Block Diagram

Figure 3.11 shows the block diagram of the smart socket and smart switch. The smart switch and smart socket both have a wireless point to point connection with the smart hub through the selected microcontroller's built-in Wi-Fi module. The smart socket and smart switch connect to an external switch through which a user can use to turn the smart socket or smart switch on/off. A relay module enables the control of ac mains electricity based on signals from the microcontroller.

3.3.3.5 Smart Switch/Socket Circuit Schematic Design

Figure 3.12 shows the schematic diagram of the smart switch and smart socket made using circuit design and simulation software Proteus. The main part of the circuit is the relay and transistor combination that is used to control the high voltage AC (230V) used by appliances in a building through a digital signal from the ESP32 microcontroller's general-purpose pin. When turning on an appliance, the ESP32 breakout board will output 3.3V of which will bias the base of the NPN transistor. When activated, the transistor will allow current to flow from the collector to the emitter and hence turn on the relay which will in turn switch on the AC load. To detect the press of the switch present on the smart socket and switch, the switch is connected to the

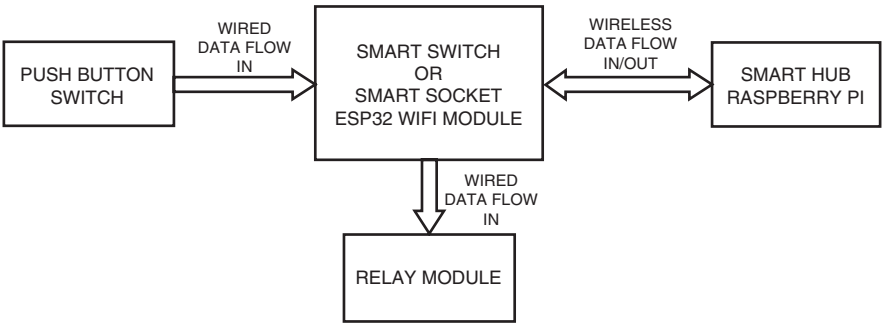


Fig. 3.11 Data flow between the smart switch and smart socket components

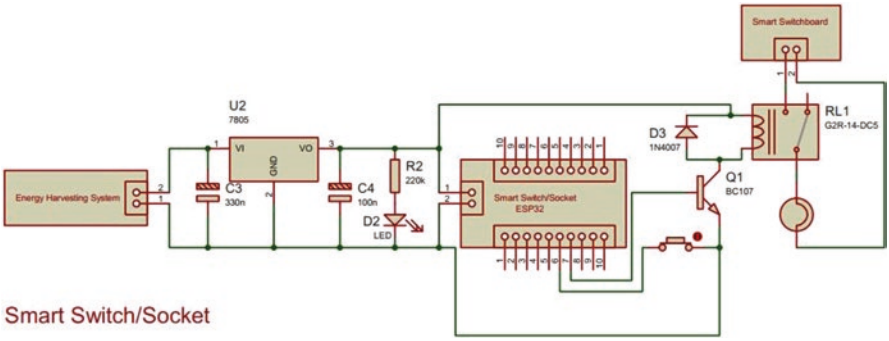


Fig. 3.12 Smart switch and socket circuit diagram

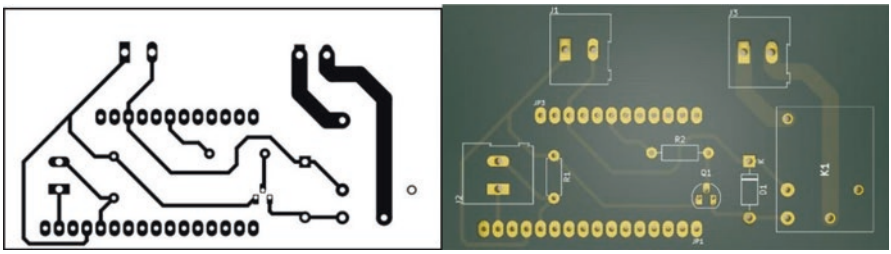


Fig. 3.13 Smart socket/switch PCB design

microcontroller general-purpose pin and ground. The powering of the switch is carried out through the microcontroller's internal pull-up resistor. The switch debouncing is carried out in software.

3.3.3.6 Smart Switch/Socket PCB Designs

Figure 3.13 shows the PCB board designs of the smart switch and smart socket designed using PCB design software KiCAD. When designing the PCB board, wide PCB tracks are used for the tracks between the relay module, AC live wire terminal, and appliance line terminal. To improve safety, enough spacing/clearance above 2 mm is left between the AC copper tracks and the microcontroller signal tracks. To connect the microcontroller board, 2.54 mm Female pin headers are used.

3.3.3.7 Smart Socket And Smart Switch Software Design

Procedure, pseudocode, and flowchart 6 describe the operation of the smart switches and smart sockets. It is used in system software design. The procedure is used to aid in the software implementation of the smart switch/socket of which is written in C++ script language.

Procedure 6: Smart Socket/Switch-Main-Loop *Description*: This is a procedure that is used by the smart sockets and smart switches when they receive an MQTT data packet command from the smart hub or when a user has pressed the local switch on the socket or switch. When a command is received, it is processed to check whether it affects its power state or the lock state of the device. If the command is to change the lock state, it then deactivates any commands from the local switch until the switch is unlocked. If the command is to change the power state, it toggles the current power state of the switch or socket. Apart from a command from the smart hub, if it receives a command from the local switch, it is first checked whether the switch is locked or not. If the switch is locked, the pulses from the switch are ignored, but if it is not locked, it toggles the power state of the smart switch or socket and then updates the smart hub of the change in the state by sending an MQTT message. The flowchart in Fig. 3.14 shows the procedures.

```
Pseudo Code 6:
1.    START
2.    Connect to Smart-Hub Wi-Fi network
3.    Establish MQTT connection with Smart-Hub
4.    WHILE forever
5.        Scan for MQTT commands
6.        IF MQTT-Data available
7.            THEN
8.            Power-State = MQTT-Data * B(01)
9.            Lock-State = MQTT-Data * B(10)
10.      ENDIF
11.      Scan Push-Button
12.      IF Push-Button pressed
13.          THEN
14.          IF Lock-State = 0
15.              THEN
16.              Toggle power state
17.              Send MQTT data packet to Update Smart-Hub
18.          ENDIF
19.      ENDIF
20.  ENDWHILE
```

3.3.3.8 Smart Switch and Smart Socket Test

The smart switch is connected to a 5V lab power supply through the DC input power terminal block. A 230V 50 Hz live wire from an AC power source is connected to the smart switch through the AC input power terminal, and an AC bulb is connected as a load on the AC output terminal end. A push-button switch is connected to the switch input terminal block. Since both the smart switch and smart

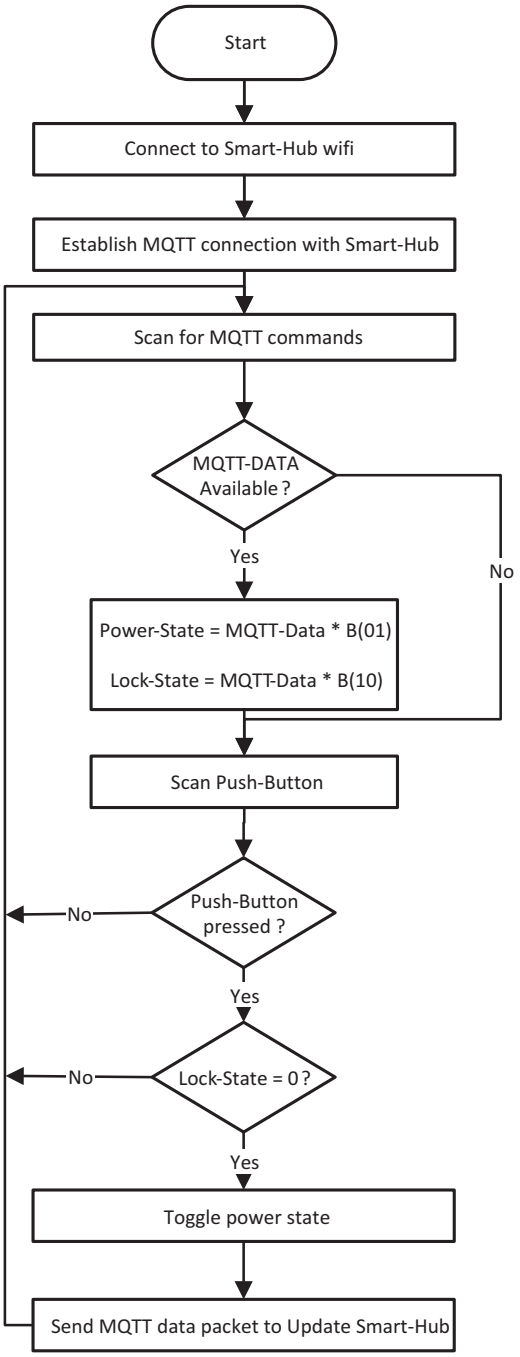


Fig. 3.14 Smart socket and smart switch flowchart

Table 3.2 Smart socket and smart switch test results

Lock state	Test	Current load state	Expected load state	Test results
Unlocked	Turn ON Load	OFF	ON	Positive
	Turn OFF Load	ON	OFF	Positive
Locked	Turn ON Load	OFF	OFF	Positive
	Turn OFF Load	ON	ON	Positive

socket are of the same design, the tests and connections made on the smart switch are the same as those of the smart socket. Table 3.2 shows the results.

3.3.4 Smart Switchboard Design

3.3.4.1 Smart Switchboard Functions

The role of the smart switchboard is to act as a remote switchboard capable of turning on or off the distribution lines in a building after the electrical panels, circuit breakers, and safety switches installed in the building's main switchboard. By doing so, users have control over buildings mains lines such as the lights mains and power sockets mains can be turned on and off.

The requirement of the smart switchboard is as follows:

(a) To switch the main distribution lines of a building on or off based on commands from the smart hub.

3.3.4.2 Smart Switchboard Hardware Selection

Apart from the absence of an input switch, the hardware selection for the smart switchboard is the same as that of the smart socket and the smart switch. It uses the same main components, such as the microcontroller and relay modules.

3.3.4.3 Smart Switchboard Block Diagram

Figure 3.15 shows the block diagram of the smart switchboard. Comparable to the smart switch and smart socket, the smart switchboard has a wireless point to point connection with the smart hub through the built-in Wi-Fi module. The relay modules enable the smart switchboard to control the state of the distribution lines from a building's switchboard.

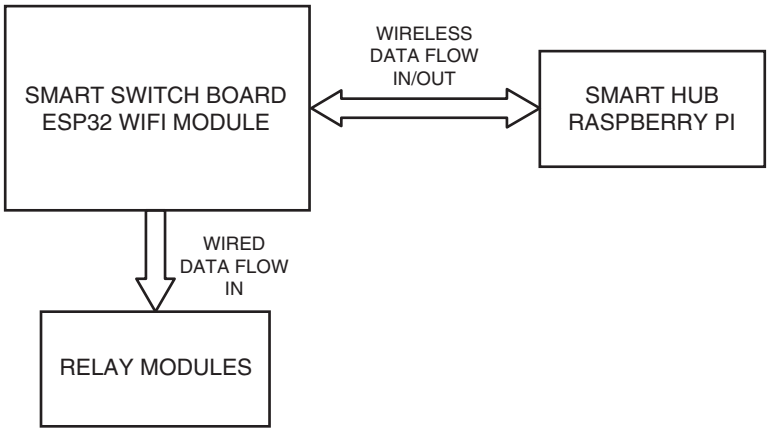


Fig. 3.15 Data flow between smart switchboard components

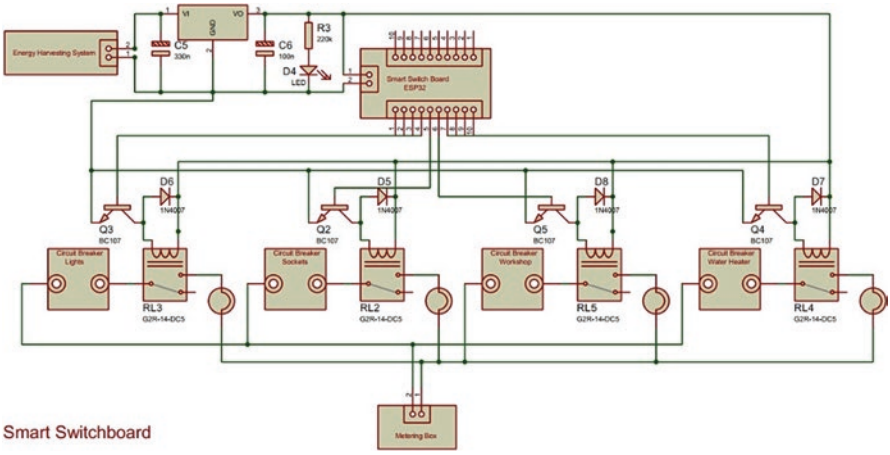


Fig. 3.16 Smart switchboard circuit diagram

3.3.4.4 Smart Switchboard Circuit Schematic Design

Figure 3.16 shows the schematic diagram of the smart switchboard made using circuit design and simulation software Proteus. Alike the smart socket and smart switch circuit, the main part of the smart switchboard circuit is the relay and transistor combination that is used to control the high voltage AC (230V) through a digital signal from the ESP32 microcontroller's GPIO pin. The design is repeated four times, therefore giving the smart switchboard capability of controlling up to four AC lines.

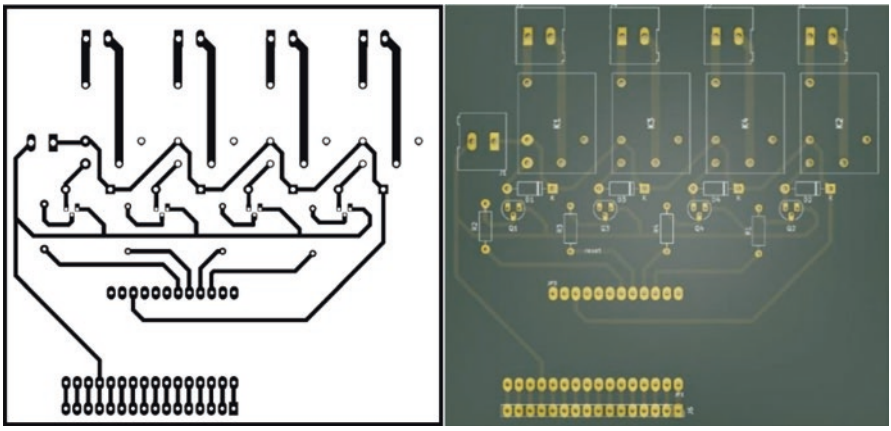


Fig. 3.17 Smart Switchboard PCB design

3.3.4.5 Smart Switchboard PCB Design

Figure 3.17 shows the PCB board design of the smart switchboard designed using PCB design software KiCAD. Comparable to the smart socket and smart switch PCB design, wide PCB tracks are used for the tracks between the relay modules, AC live wire terminals, and appliance line terminals. Safety is also observed by ensuring enough spacing/clearance above 2 mm between the AC copper tracks and the microcontroller signal tracks. To connect the microcontroller board, 2.54 mm Female pin headers are used.

3.3.4.6 Smart Switchboard Software Design

Procedure, pseudocode, and flowchart 7 describe the operation of the smart switchboard. It is used in system software design. The procedure is used to aid in the software implementation of the smart switchboard of which is written in C++ script language.

Procedure 7: Smart switchboard *Description*: This is a procedure that is used by the smart switchboard when it receives an MQTT data packet command from the smart hub. When a command is received, it is processed to extract the main appliance line that is to be turned off or on. Having obtained the appliance line, the power state of the line is set according to the command from the smart hub. The flowchart in Fig. 3.18 shows the procedures.

3.3.4.7 Smart Switchboard Test

The smart switchboard is connected to a 5V lab power supply through the DC input power terminal block. Live wires from 230V 50 Hz AC power source are connected to the smart switchboard at AC input power terminals, and AC bulbs are connected

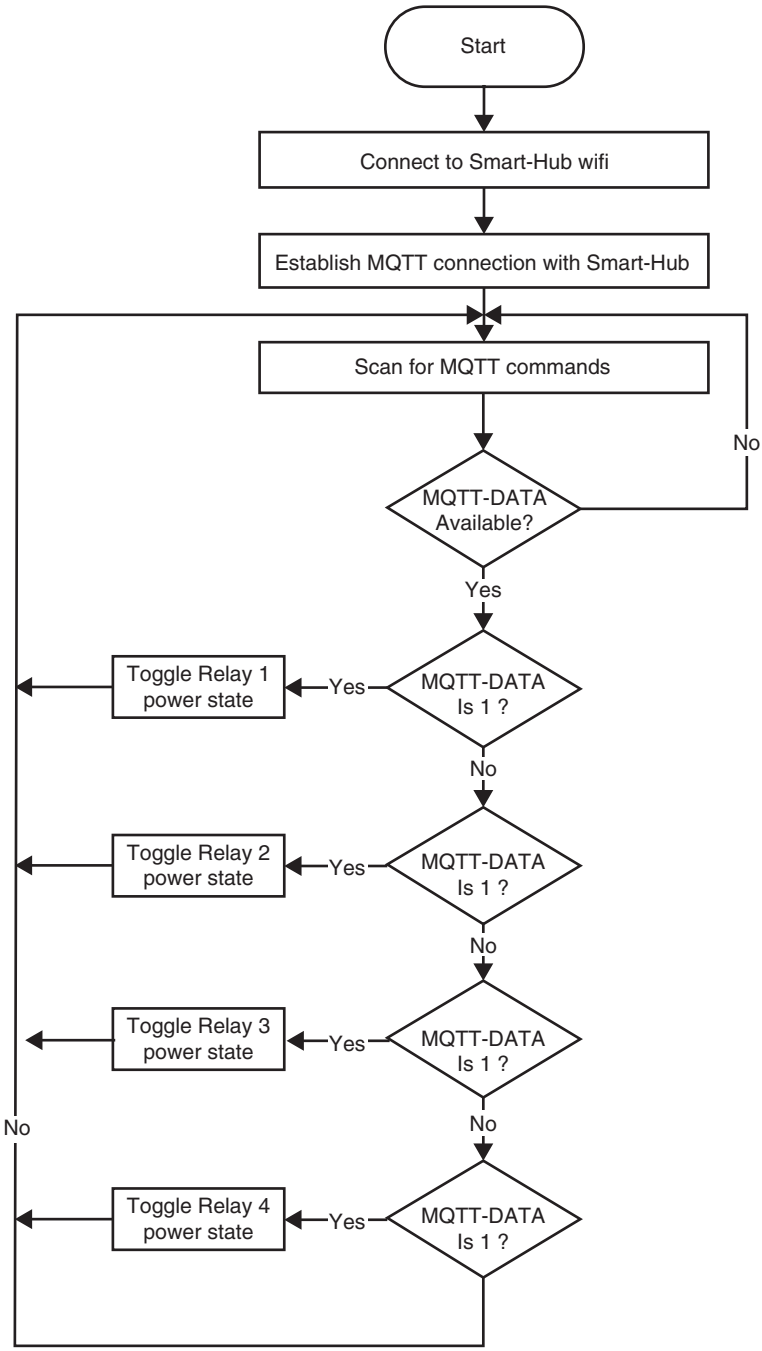


Fig. 3.18 Smart switchboard flowchart

Table 3.3 Smart switchboard test results

Test	Current load state	Expected load state	Test results
Turn ON Relay 1–4	OFF	ON	Positive
Turn OFF Relay 1–4	ON	OFF	Positive

as loads on the AC output terminals end. The smart switchboard has no input switches and is controlled by commands from the smart hub. However, temporary switches were used in the testing of the smart switchboard. Table 3.3 shows the results.

```
Pseudo Code 7:
1.    START
2.    Connect to Smart-Hub Wi-fi network
3.    Establish MQTT connection with Smart-Hub
4.    WHILE forever
5.        Scan for MQTT commands
6.        IF MQTT-Data available
7.            THEN
8.            IF MQTT-Data = 1
9.                Toggle relay 1 power state
10.       ELSE IF MQTT-Data = 2
11.                Toggle relay 2 power state
12.       ELSE IF MQTT-Data = 3
13.               Toggle relay 3 power state
14.       ELSE IF MQTT-Data = 4
15              Toggle relay 4 power state
16.      ENDIF
17.  ENDWHILE
```

3.4 Implementation of the Designed Solution

This section presents the results of the system implementation. Once having designed and tested the individual sub-systems of the secured smart home switching system, the sub-systems are integrated to create a complete Smart IoT system implementation. The implementation is carried out in two parts. First, it is implemented in a scaled-down version of a house, shown in Fig. 3.19 and then implemented in an actual sized room, shown in Figs. 3.20, 3.21, 3.22, and 3.23.

Figure 3.19 is a picture of the implementation of the designed Smart IoT system as a scaled-down version of a house.

Figures 3.20, 3.21, 3.22, and 3.23 are pictures of the implementation of the designed smart IoT system implemented in an actual sized room.

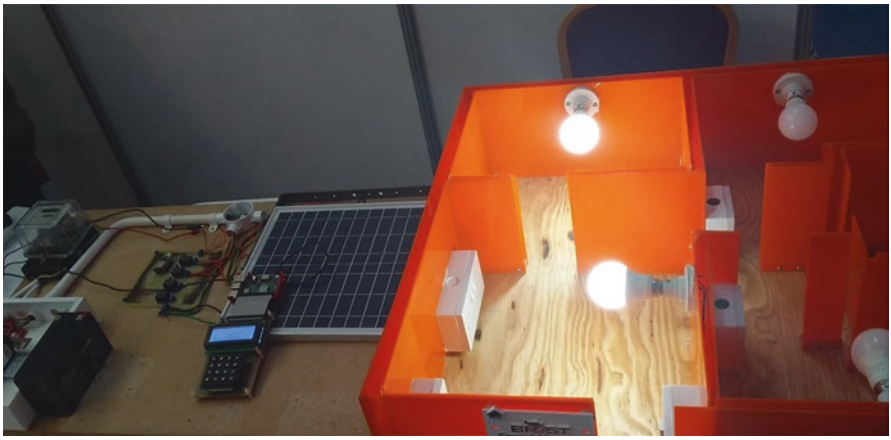


Fig. 3.19 Picture of Prototype 1 smart switch and smart sockets

Fig. 3.20 BIUST Off-Grid secured smart house front side picture

3.5 Summary

In this chapter, we report on the design and implementation of smart IoT devices and applications through the solving of an example problem of adding security and access control to a building's mains electricity. Through this chapter, one should now be familiar with the design process of making smart IoT devices. We first

Fig. 3.21 BIUST Off-Grid secured smart house inside-picture

Fig. 3.22 Secured smart home switching system smart switch implementation

described the said problem through problem definition, aim, and objectives. We then covered the methods and procedures undertaken to design the smart IoT devices used to solve the given example problem through the system structure, hardware design, and software design. Finally ended with showing the results of the system implementations.

Fig. 3.23 Secured smart home switching system smart hub and smart switchboard implementation

References

1. A.M. Zungeru, J. Gaboitaolelwe, B. Diarra, J.M. Chuma, L. Ang, L. Kolobe, M. David, I. Zibani, A secured smart home switching system based on wireless communications and self-energy harvesting. IEEE Access **7**, 25063–25085 (2019)
2. Raspberry Pi, Raspberry Pi 3 Model B+ product brief. Raspberry Pi, [Online]. Available: https://static.raspberrypi.org/files/product-briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf.
3. Espressif Systems, ESP32-WROOM-32 datasheet V2.9. Espressif Systems (2019), [Online]. Available: https://www.espressif.com/en/support/download/documents?keys=esp32. Accessed 3 Mar 2020

Chapter 4
Design of Photovoltaic System for IoT Devices

Adamu Murtala Zungeru, Joseph M. Chuma, Dauda Duncan, Bakary Diarra, Modisa Mosalaosi, Bokani Mtengi, and Jwaone Gaboitaolelwe

4.1 Introduction

In Chap. 3, we discussed the design and implementation of smart IoT devices through an example problem. This was carried out through the solving of a specific application example concerning the problem of securing a building's main power supply against unauthorized use. However, the smart IoT devices use most of the supply energy; hence, the need to self-power smart homes. In this chapter, we report on the design procedure and implementation of a photovoltaic system for the IoT devices.

The increasing world population growth and improving standards of living in developing nations add to the escalated energy demands. Technological advancements have resulted in increased use of products that require energy, leading to significant efforts made by countries to meet the demand by burning more fossils for energy production. However, fossils are finite and cause more harm to the environment than renewable energy sources. Even with environmental concerns and unreliable fossil costs, industries around the world are still dependent on fossils for better power quality and efficiency. For developing countries, the challenge remains the ability to supply quality power to remote and off-grid areas. Renewable energy sources, especially solar, become a solution of choice. Technological advancements offer capabilities for the generation of renewable energy, storage, and off-grid design for localized usage. Off-grid photovoltaic system is a standalone source and alternative to conventional sources of power supply. Recent surveys show that the off-grid photovoltaic system delivers reduced routine cost options compared to the usual conventional power generator [1–5]. An extension of the work presented in this chapter is published in [10]. However, the photovoltaic system has an initial higher cost of installation, while the conventional power generator has a higher running cost [6–9]. Even with the initial high costs of installation, off-grid systems offer long term economic benefits and

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_4

energy access to regions where renewable energy sources are abundant. Off-grid photovoltaic system is a scheme that processes ambient solar irradiance and converts it into functional electrical energy for electrical loads. In designing an energy harvesting system for a load of 2000 W, sequentially technical specifications of the energy transducer, the power conditioner, and energy bank are determined. The photovoltaic transducer is one of the most common energy transducers which harvests its energy from abundant solar irradiance in the ambient. The efficiency of the transducer depends mainly on ambient irradiance and temperature, relying on the time of the day and season of the year as well as its characteristics. The goal of the power conditioner is to maximize the output of the energy transducer by optimally matching the impedances of the transducer and the load. Also, it has the capability to adapt to varying inputs and consequentially variations of current and voltage levels at the load. In order to improve the power conditioning efficiency, Maximum Power Point Tracking (MPPT) is incorporated to achieve maximum power transfer [10–12]. The MPPT keeps on adjusting the duty cycle of a power switch of the DC–DC converter, accordingly, as the ambient temperature and solar irradiance fluctuates. Although, in Africa context, particularly in Botswana, it receives more than 2000 KW/m^2 of irradiance in an hour [13], which is enough to exploit.

This chapter intends to report on the design of a solar array system to be used as a sole energy driver for a standalone photovoltaic system used in powering smart devices in the smart homes. The design is preceded by determining the number of solar panels required to optimize the available space on the rooftop of a portacabin to be used as a self-sufficient energy house. In general, the design of a solar system is informed by the load intended to be driven. The load (usage) is determined either by performing calculations of the amount of energy required by the local appliances using nameplate specifications provided by the manufacturer, or by consulting the utility energy usage report. In this case, however, a different design process was followed due to certain constraints. Specifically, the load that the system can support is sized, which is the opposite of the traditional approach. This is particularly relevant to consumers with limited/fixed spaces. Therefore, this is a bottom-up strategy (sizing the load as opposed to sizing the solar array system). The significance of this approach is that, generally, the space is readily available in the form of home, factory, or office roofing. The traditional approach (top-down strategy where the solar system is sized based on the target load), however, requires flexibility in space available to meet the target load, a luxury that may not always be afforded to consumers especially in congested cities. As such, the design process proposed herein allows the consumer to make an informed and realistic decision on which loads may be shed off the grid. The design process presented here is inherently scalable dependent on the available roof area, solar panel output, and footprint; therefore, it is suitable for homes, factories, and office spaces.

4.2 Technical Specifications of the Design

The chapter reports on the design of a solar array system to be used as a sole energy driver for the departmental green Internet of Things office devices. The design is preceded by determining the number of solar panels required to optimize the available space on the rooftop of a portacabin to be used as a self-sufficient energy house. Details of the design reports and results are presented in [10]. The load (usage) is determined either by performing calculations of the amount of energy required by the local appliances using nameplate specifications provided by the manufacturer or by consulting the utility energy usage report. In this case, however, a different design process will be followed due to certain constraints. Specifically, the load that the system can support will be sized, which is the opposite of the traditional approach. The limitations of this system design, as mentioned earlier, are as follows:

- Spatial constraints: the spatial limitation due to the fixed size of the target roof.
- Solar panel size: the number of panels in an array is determined by the size of individual panels. The choice will be limited by the available panels.
- Single panel output: different solar panels have different power outputs. The more is the power, the better it will be over a given area of space. In this design, there will be a choice between 150 W (at 12 V), 100 W (at 18 V), and 80 W output panels. Space manipulation will play a role since, in general, the more power, the bigger the size of the panel (for a specific manufacturer design).
- Array output: it is determined by the size of the array, which is determined by the output of individual units as well as the array configuration.

The rest of the design in terms of battery size, charge controllers, and inverters will depend on the size of the array (output power). The charge controllers at our disposal may not be sufficiently specified for the array application since they are specified for individual solar panels (input voltage 14–30 V at open circuit (12 V nominal voltage) and 6 A/10 A current output). The physical layout of the solar system array atop the house is shown in Fig. 4.1. There is a choice of using the 150 W or 100 W rated solar panels. Moreover, the actual layout may be in a landscape or portrait manner. The 150 W rated solar panels have dimensions 124 × 1.4 × 84 cm, while the 100 W panels are 120 × 3 × 54 cm in dimensions. These physical dimensions are used to determine the possible grid layout scenarios shown in Fig. 4.1. The output characteristics of each grid layout will be assessed in terms of output power, voltage, and current ratings. The ratings will not be determined until the wiring configuration (series, parallel, or hybrid connection) is taken into consideration. Figures 4.2 and 4.3 show the series and parallel configurations for the grid assembly, and a hybrid may also be considered.

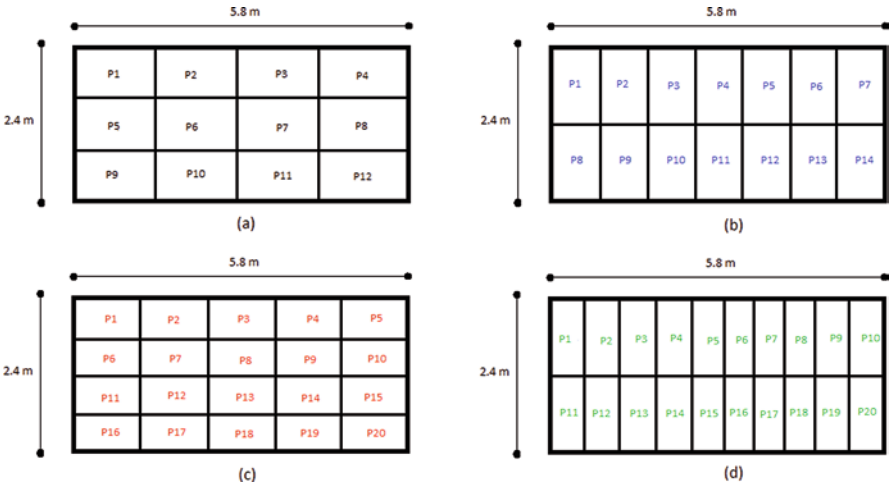


Fig. 4.1 Solar panel array layout (**a**) using 150 W panel units in a (4 × 3) grid (**b**) using 150 W panel units in a (7 × 2) grid (**c**) using 100 W panel units in a (5 × 4) grid, and (**d**) using 100 W panel units in a (10 × 2) grid

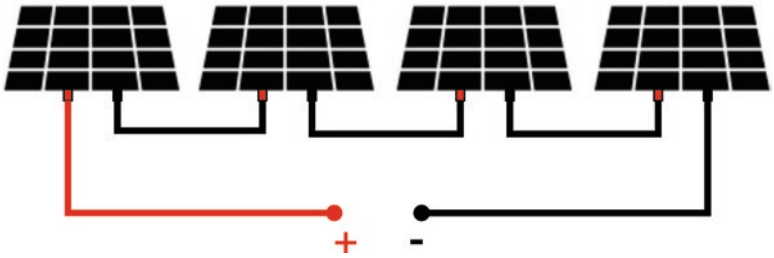


Fig. 4.2 Series array configuration using four (4) solar panels

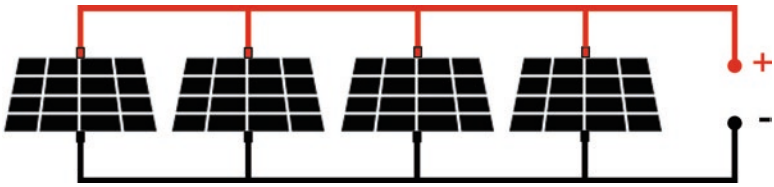


Fig. 4.3 Parallel array configuration using four (4) solar panels

The configuration chosen for the grid layout determines the charge controller, battery bank, and the inverter size. The behavior of the final grid configuration is governed by basic Kirchhoff's laws. The capacity of the solar system is determined by the number and rating of individual solar panels. The grid layout of Fig. 4.1c will have a capacity of (100 × (5 × 4)) W = 2000 W, while that of Fig. 4.1b has a capacity

of (150 × (7 × 2)) W = 2100 W. Based on the available space and size of a single 100 W panel unit, both the portrait and landscape array layouts (Fig. 4.1c, d) yield the same capacity of 2000 W. Once the configuration is chosen, the voltage and current rating of the solar array will be determined. The system size must match up with the inverter size, which generally comes in numerous sizes.

In the series configuration of Fig. 4.2, we apply Kirchhoff's voltage law (KVL) in the loop. In such a circuit, the voltage of all the components combines in a summative manner, while the current remains the same. That is, the output current rating of the array will be the same as that rate of a single unit. However, the voltage rating of the array will be the sum of all the voltages rated across each solar panel. Alternatively, the voltage rating of the array is n (the number of solar panels in the array) times the voltage rating of a single solar panel since all the panels are identical. The advantage of this configuration is that the output current of the array is minimized, which enables the design to employ smaller sized wires. In terms of wire sizing, it is common knowledge that the wire size is proportional to the amount of current flowing through the wire. Therefore, each wire is rated for a specific amount of current flowing through it. The smaller is current, and the smaller is the size of the wire.

The alternative solar array configuration is the parallel type, which is shown in Fig. 4.3. In this configuration, Kirchhoff's current law (KCL) is applied at any of the two nodes. The current output of the solar array is equal to the sum of an individual rated output current of the panels, which is equivalent to n (the number of solar panels in the array) times the current rating of a single solar panel. Since the panels are in parallel, the same voltage that appears across each individual solar panel will similarly appear across the output of the array. That is, the solar array system will have the same voltage rating as the individual panels that constitute it. In comparison with the series configuration, one quickly notices the stark difference in the output current of the respective arrays. The parallel configuration will need much thicker wires out of the panels in order to withstand the output current as compared to the series configuration. Moreover, the requirements of the charge controller will be higher (usually translates into hefty prices) than that of the series configuration. However, the series configuration is prone to the effects of shading. When one of the panels is shaded, the entire array system is driven off. This is typical of series circuits when the loop is opened at any part of the circuit, and the current ceases to flow to the output.

On the contrary, in the parallel configuration, if one of the panels is shaded, only its contribution will be missed. The issues narrated above are not just peculiar to shading; the same will apply if a panel malfunctions or ceases to operate for whatever reason. Using Fig. 4.3 as an example, if one-panel malfunctions, the system will operate at 75% capacity as opposed to not be available. The technical specifications of the design considerations are quantified in Table 4.1. As can be observed from Table 4.1, the amount of amperage that is produced by the parallel configurations is much higher than that of the series configurations. In translation, the charge controller requirements will be more costly for the parallel configuration. The series configuration, however, has a very high voltage rating while producing very little

Table 4.1 Technical specifications for four (4) possible design configurations

Array layout	Configuration	Power rating (W)	Voltage rating (V)	Current rating (A)
Design 1	Series Parallel	1800 1800	144 12	12.5 150
Design 2	Series Parallel	2100 2100	168 12	12.5 175
Design 3	Series Parallel	2000 2000	360 18	5.6 111
Design 4	Series Parallel	2000 2000	360 18	5.6 111

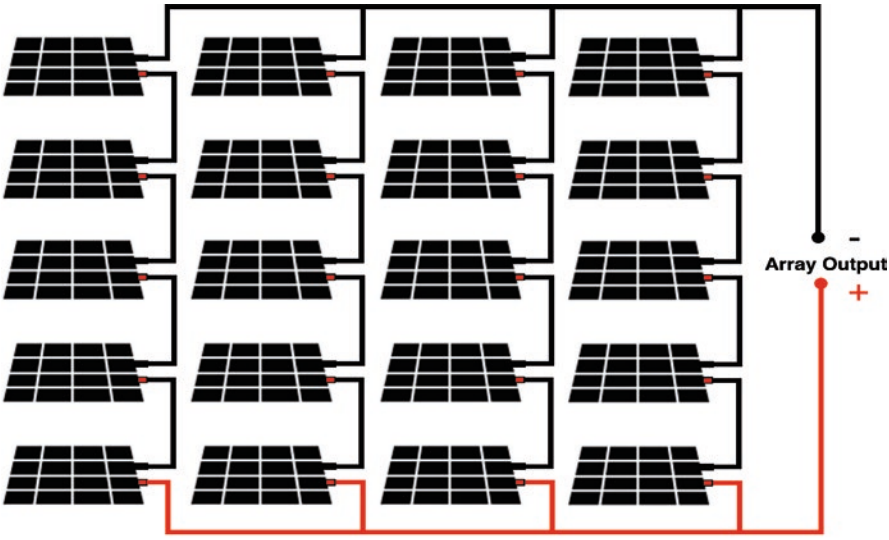


Fig. 4.4 Hybrid wiring configuration of Fig. 4.1a

current. It might not be feasible to realize such a high voltage system that is capable of driving light/few loads due to the current limitation. This, therefore, requires that other alternatives be explored in order to obtain a realistic and cost-effective system. A hybrid configuration is proposed as a solution.

The proposed system is equally subjected to the same spatial constraints, available solar panels, and individual solar panel output. The expectation is that the final system will benefit from the properties of both the series and parallel configurations, albeit in moderation. The wiring configuration of such a design is shown in Fig. 4.4, which is a hybrid implementation of Fig. 4.1a. The same approach is adopted to implement the designs of Fig. 4.1b–d. The numerical results of these four hybrids wiring configurations are depicted in Table 4.2.

It can be clearly seen from Table 4.2 that the hybrid wiring configuration moderates both the current and voltage specifications across all design configurations. While this optimization is achieved, the power rating of the system remains

Table 4.2 Technical specifications for four (4) possible hybrid wiring configurations

Array layout	Configuration	Power rating (W)	Voltage rating (V)	Current rating (A)
Design 1	Hybrid 1	1800	48	37.5
Design 2	Hybrid 2	2100	84	25
Design 3	Hybrid 3	2000	90	22.2
Design 4	Hybrid 4	2000	180	11.1

Table 4.3 Battery bank system voltage recommendations

System power rating (W)	Battery bank voltage rating (V)
Up to 500	12
500–2000	24
2000–10,000	48
Above 10,000	96

unchanged as before, as it can be observed in the third column of Table 4.2. Generally, as it can also be observed from Table 4.2, arranging more solar panels in series than in parallel will bias the system towards a higher voltage and lower current and vice versa. This is only true, provided the solar panels are rated the same (assuming they also have equal spatial footprint). In Table 4.2, Hybrid 3 and Hybrid 4 wiring configurations are derived from solar panels of the same specifications (including size), the same is true for Hybrid 1 and Hybrid 2.

Since the objective of this project is to accommodate energy storage to be used during times when there is no sunshine, the next logical step is to consider a storage system. Batteries are usually used for this purpose. Based on the specifications of this solar system (approx. 2000 W), matching battery size is required. However, the sizing of the battery depends on the operating hours. We will assume an average daily solar reception of 4 h (typical). Therefore, the energy produced by the system will be calculated as $2000 \times 4 = 8000$ Wh. The battery bank size may now be determined. The most popular batteries used for this purpose currently comes with 12 V, 100/120 Ah capacity with a maximum depth of discharge (DOD) of 70%. This is the maximum capacity of the battery, which could be used before recharging.

When it comes to system voltage capacity, there is generally a rule of thumb [14] that makes recommendations by providing operating ranges, depending on the solar system power (this need not be rigid). Such guidelines are presented in Table 4.3.

Based on Tables 4.1, and our solar panel system has a capacity of 2000 W; therefore, the battery bank system voltage should be rated at 24 V. We will also consider a 12 V, 120 Ah battery for the battery bank. The charge capacity of the battery bank = Energy required/System voltage = 8000/24 = 1000/3 Ah. With the charge capacity of the battery bank, we are now able to determine the required number of batteries to generate such charge.

$$\text{Number of batteries required in parallel} = (1000/3) \div (120 * 0.7) = 3.97$$

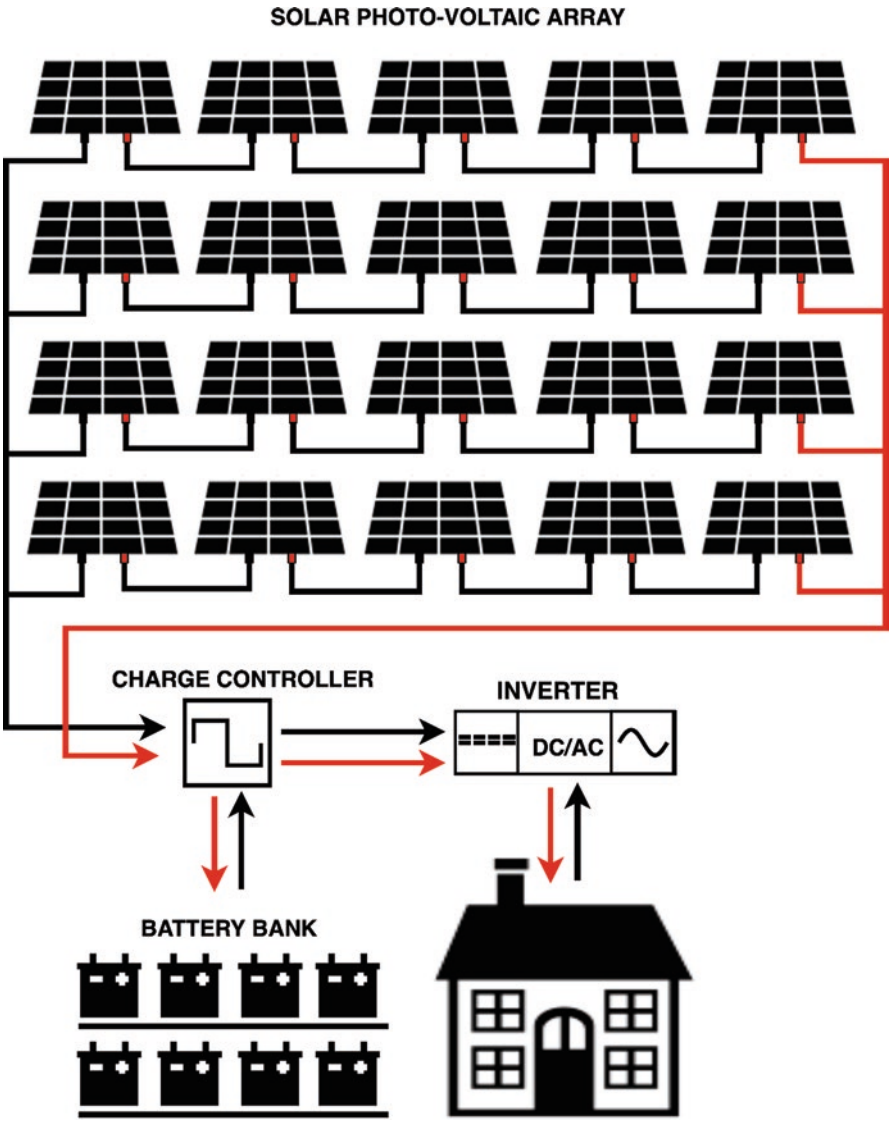


Fig. 4.5 Schematic diagram of the off-grid PV system

$$\text{Number of batteries required in series} = \text{system voltage considered} / \text{voltage of battery} = 24 \div 12 = 2$$

Thus, the battery bank should consist of two batteries in series and four batteries in parallel.

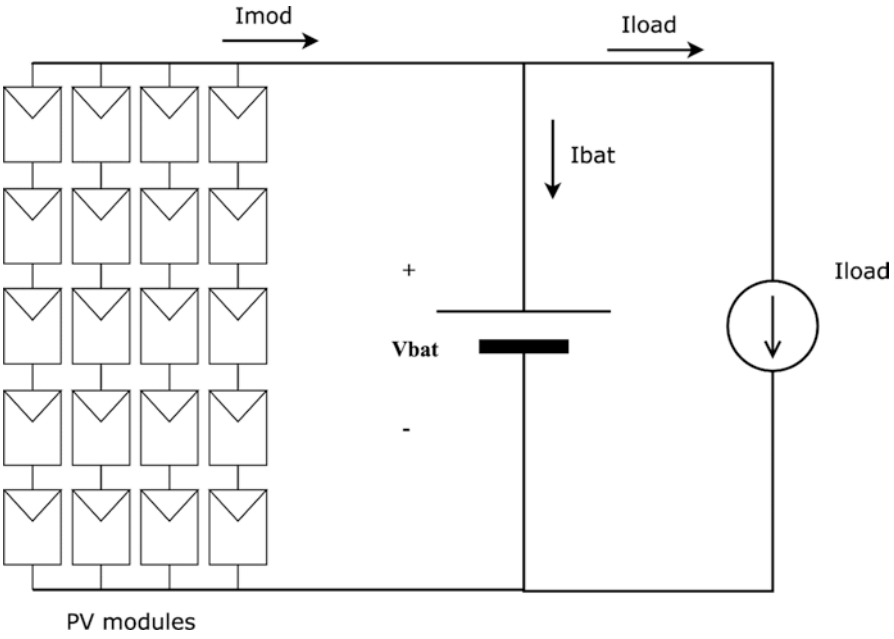


Fig. 4.6 Circuit representation of the off-grid PV system

Furthermore, we verified the specifications following the method adopted in [14]. We started by looking at the general layout of our design as well as the circuit representation of the system.

The designed off-grid connected photovoltaic system consists of 20 solar panels in a hybrid wiring configuration, eight lead-acid batteries, charge controller, inverter, and the portacabin building. Figure 4.5 shows the schematic diagram of the off-grid PV system, while Fig. 4.6 is the circuit representation of the schematic diagram.

In this work, the off-grid system provides enough energy to the secured smart system and the entire building (portacabin). The configuration of our design is as shown in Fig. 4.6, where the PV modules, also termed as generators [14], are connected to the batteries via the charge controller. The batteries then power the entire system via the inverter circuitry.

4.2.1 Equivalent Peak Solar Radiation Hours and Photovoltaic Array Sizing

In our analysis, we introduce the concept of peak solar hours, similar to [14], for the cell temperature of 25 °C, equivalent to the average temperature of the Palapye-Botswana region. If we consider the actual irradiance profile for a particular day as $D(t)$ for which the irradiance of the equivalent day is 1 kW/m^2 during a time of peak

solar radiation hours (PSRH), we can then write the relationship of the total daily radiation in the actual day, similar to its equivalent day as:

$$\int_{\text{day}} D(t)\,dt = 1.\text{PSRH} \tag{4.1}$$

With the solar cell used in the design with electrical ratings of maximum points of current and voltage of 5.56 A and 18 V, respectively, for which the sun peak hours was found to be 3.25 h, the energy generated per day as simulated using PSIM [15] is given by:

$$\int_{\text{day}} P_{\max}(t)\,dt = \int_{\text{day}} I_{mM}(t) V_{mM}(t)\,dt = 344.2\,\text{Wh/day} \tag{4.2}$$

$$150\,\text{W} \times \text{PSRH} = 100\,\text{W} \times 3.25\,\text{h} = 325\,\text{Wh\_day} \tag{4.3}$$

However, it was noted that when using the peak solar radiation hours to size and estimate the energy generated per day, one will underestimate the energy delivered by the photovoltaic modules at the days to be estimated, whereas the use of Eq. (4.2) will give better results in terms of energy delivered by the same PV modules for the days been estimated. It is worth to note that we consider the worst-case scenario, whereby the temperature is low to about 15 °C with low irradiance values too. In order to balance the energy generated by the off-grid system, it is necessary to size the PV array system. The energy balancing technique utilized the PSRH concept in writing the energy balance relationship as:

$$P_{\text{G-Nom}}\text{PSRH} = \text{L}_E \tag{4.4}$$

where $P_{\text{G - Nom}}$ is the nominal output power of the photovoltaic system for a standard condition [14], L_E the consumed energy over the average day by the load in the worst case. We can re-write Eq. (4.4) for two cases (worst-case design and average design) as:

1. Worst-case design of the system:

$$P_{\text{G-Norm}}\left(\text{PSRH}\right)_{\min} = \text{L}_E \tag{4.5}$$

where $(\text{PSRH})_{\min}$ represents the value of PSRH in the worst month of solar radiation.

2. Average-case design of the system:

$$P_{\text{G-Norm}}\left(\overline{\text{PSRH}}\right) = \text{L}_E \tag{4.6}$$

where $\overline{\text{PSRH}}$ represents the average value of the 12 monthly PSRH values. However, throughout our design, the average design was utilized. Substituting for

current and voltage in the nominal maximum power of the photovoltaic system, Eq. (4.6) becomes

$$V_{\mathrm{GN}} I_{\mathrm{GN}} \left(\overline{\mathrm{PSRH}}\right) = \mathrm{L}_E \tag{4.7}$$

However, in most applications of a photovoltaic system, a series of parallel connections of PV modules are most common. In this, the photovoltaic system is composed of $N_{\mathrm{S\text{-}PV}}$ series string of $N_{\mathrm{P\text{-}PV}}$ identical (parallel) modules and Eq. (4.7) becomes

$$N_{\mathrm{S\text{-}PV}} V_{\mathrm{GM}} N_{\mathrm{P\text{-}PV}} I_{\mathrm{GM}} \left(\overline{\mathrm{PSRH}}\right) = \mathrm{L}_E \tag{4.8}$$

where V_{GM} and I_{GM} represents the voltage and current, respectively, of the maximum power point of the photovoltaic system for the 1Sun AM1.5 standard [14]. This then means that the hybrid connections of the system are determined by the $N_{\mathrm{S\text{-}PV}} N_{\mathrm{P\text{-}PV}}$ as:

$$N_{\mathrm{S\text{-}PV}} N_{\mathrm{P\text{-}PV}} \frac{\mathrm{L}_E}{V_{\mathrm{GM}} I_{\mathrm{GM}} \left(\overline{\mathrm{PSRH}}\right)} \tag{4.9}$$

However, for an off-grid photovoltaic system, the loads are connected to either a battery or supercapacitor in the form of $\mathrm{V_{CC}}$ (DC Voltage). The energy consumed by the loads in the building over a day can now be represented as:

$$\mathrm{L}_E = 24\,\mathrm{V_{CC}}\, I_{\mathrm{DC\text{-}eq}} \tag{4.10}$$

where $I_{\mathrm{DC\text{-}eq}}$ represents the equivalent DC current drawn by the loads over the 24 h of the day. Considering Eqs. (4.8) and (4.10), we have

$$N_{\mathrm{S\text{-}PV}} V_{\mathrm{GM}} N_{\mathrm{P\text{-}PV}} I_{\mathrm{GM}} \left(\overline{\mathrm{PSRH}}\right) = 24\mathrm{V_{cc}} I_{\mathrm{DC\text{-}eq}} \tag{4.11}$$

However, Eqs. (4.8) and (4.10) as equated might not be exactly equal due to the underestimates of peak solar hour concept or overestimates of the photovoltaic energy generation system. Also, other factors might be due to the efficiency of the panels, the charge controller, the MPPT, and the inverting system (DC/AC converter). Hence, it is expected that a factor (safety factor) to augment inequality is added to the system so that Eq. (4.11) becomes

$$N_{\mathrm{S\text{-}PV}} V_{\mathrm{GM}} N_{\mathrm{P\text{-}PV}} I_{\mathrm{GM}} \left(\overline{\mathrm{PSRH}}\right) = \left(C_{\mathrm{SF}}\right) 24\mathrm{V_{cc}} I_{\mathrm{DC\text{-}eq}} \tag{4.12}$$

Therefore, to size the PV modules in parallel or series, we can arrive at Eqs. (4.13) and (4.14), respectively, as:

$$N_{\text{S-PV}} = \left(\text{VC}_{\text{SF}}\right)\frac{V_{\text{cc}}}{V_{\text{GM}}} \tag{4.13}$$

$$N_{\text{P-PV}} = \left(\text{IC}_{\text{SF}}\right)\frac{24I_{\text{DC-eq}}}{I_{\text{GM}}\left(\overline{\text{PSRH}}\right)} \tag{4.14}$$

where VC_{SF} and IC_{SF} are voltage and current safety factors, respectively.

In our design, we consider the whole building of load consumption of 12,500 Wh per day at 24 V. Each of the solar panels has the following ratings:

$$P_{\text{SP}} = 100\text{W}; V_{\text{OC}} = 22.10\text{V}, \quad I_{\text{SC}} = 6.1\text{A}, \quad I_{\text{MP}} = I_{\text{GM}} = 5.56\text{A}, \quad V_{\text{MP}} = V_{\text{GM}} = 18\text{V}$$

We then compute the required current drawn by the load over the entire day as:

$$I_{\text{eq}} = \frac{\text{L}_E}{24\text{V}_{\text{cc}}} = \frac{12,500}{24 * 24} = 21.7\text{A} \tag{4.15}$$

where $\text{L}_E = \dfrac{12,500\text{Wh}}{\text{day}}$

4.2.2 Equivalent Battery Sizing in the Photovoltaic Systems

In [14], a simplified equation governing the sizing of batteries in a standalone system is given by:

$$E_{\text{b}} = \left\{MAX\left[\left(E_{\text{bal}}\right)_{\text{max}} + E_{\text{backup}}, E_{\text{cycle}}\left(\frac{1}{x}\right)\right]\right\}\frac{1}{y\eta_{\text{cd}}} \tag{4.16}$$

and

$$E_{\text{bal}} = \left(E_{\text{ph}}\right)_{\text{month}} - \left(E_{\text{cons}}\right)_{\text{month}} = N_{\text{S-PV}} \times N_{\text{P-PV}} \times P_{\text{maxGr}} \times \left(\text{PSRH}\right)_{\text{month}} - n_i\left(E_{\text{cons}}\right)_{\text{day}} \tag{4.17}$$

where E_{backup} represents the amount of energy stored to guaranty system operation for a certain number of days, $(E_{\text{bal}})_{\text{max}}$ stands for the maximum seasonal deficit accounting for the operation of a full year, E_{cycle} the energy deficit due to power mismatch, x the battery daily cycle factor, y the discharge factor maximum depth, and η_{cd} represents the efficiency of the battery charge and discharge. Also, E_{ph} represents the energy generated by the photovoltaic system, E_{cons} the energy consumed by the load for some time, and $n_i$the number of days in a particular month. Equation (4.16) is measured in Watt-hour (Wh), but it is advisable to measure batteries in Ampere-hour (Ah) as the standard units for batteries, such that Eq. (4.16) becomes

$$E_{\text{b\_standard}} = \frac{E_{\text{b}}}{V_{\text{cc}}} \tag{4.18}$$

For our design, we consider a total load of 2000 W, for which we have arrived at 5 × 4 solar panels arrays. We assumed batteries' daily cycling of less than 15% of maximum discharge depth at 75% with charge-discharge efficiency of 90%. Hence,

$$E_{\text{bal}} = 39{,}004\text{Wh} \tag{4.19}$$

With backup storage for 7 days,

$$\begin{aligned} E_{\text{backup}} &= 7 \times 2000 = 14{,}000\text{Wh} E_{\text{b}} \\ &= \left\{ MAX\left[39{,}004 + 14{,}000{,}2000\left(\frac{1}{0.15} \right) \right] \right\} \frac{1}{0.75 * 0.90} = 78{,}519\text{Wh} \end{aligned} \tag{4.20}$$

and

$$E_{b\_\text{standard}} = \frac{78{,}519}{24} = 3272\text{Ah}$$

$$C_{\text{standard}} = \frac{E_{\text{battery}}}{L_{\text{E}}} = \frac{78{,}519}{2000} = 39.3\,\text{days}$$

From our battery sizing, it is possible to store our harvested energy for use up to 40 days.

Final Component Specification

1. Solar inverter: rated at 2000 W
2. Charge controller (MPPT): 24 V delivering 30–40 A
3. Batteries for battery bank: 8 × 120 Ah rated batteries
4. Solar panels: 20

4.3 Design of Photovoltaic System and MPPTs

A solar panel is a transducer that houses several photovoltaic (PV) cells connected in parallel or series depending on the application intended. To design and simulate the PV system, it is necessary to model the PV cell and study the characteristics of I–V and P–V plots because they define the PV behaviors. Figure 4.7 illustrates an equivalent circuit of a PV cell consisting of a current source connected in parallel with a forward-biased PN junction diode and a shunt resistance R_{SH} all in series with resistance R_{S}. While the current and voltage relationship is expressed in Eq. (4.22).

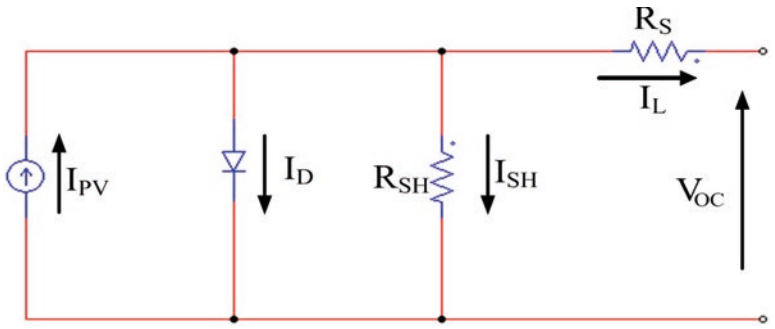


Fig. 4.7 Equivalent photovoltaic cell circuit, where I_{PV} = Current generated from irradiance incident on the solar cell, I_D = Diode Current, R_S = Parasitic Series Resistance, R_{SH} = Parallel Shunt Resistance, V_{OC} = Open-Circuit Voltage

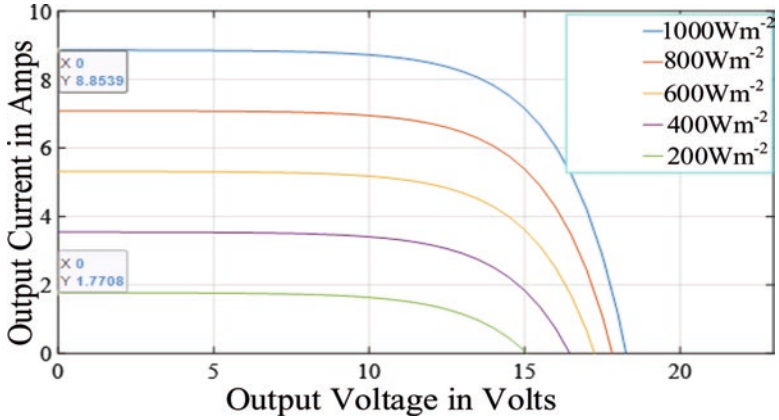


Fig. 4.8 An I–V plot of irradiance variations at 25 °C for solar panel

Figures 4.8 and 4.9 illustrate a PV cell showing current and voltage relationships under the irradiance variations at 25 °C provided by a model at Eq. (4.22). The fixed temperature and varying irradiance influence the performance of a PV cell [16], as illustrated in Figs. 4.8 and 4.9.

Concerning the current source I_{PV} feeding the diode and a voltage across the resistances in Fig. 4.7, the Kirchoff's Current Law expresses the current, I_L as [17]:

$$I_L = I_{PV} - I_D - I_{SH} \tag{4.21}$$

The current and voltage relationship in the PV cell is expressed as [17]:

$$I_L = I_{PV} - I_s \left\{ \exp\left[\frac{q\left(V + IR_S\right)}{KT_C A} \right] - 1 \right\} - \frac{V + IR_S}{R_{SH}} \tag{4.22}$$

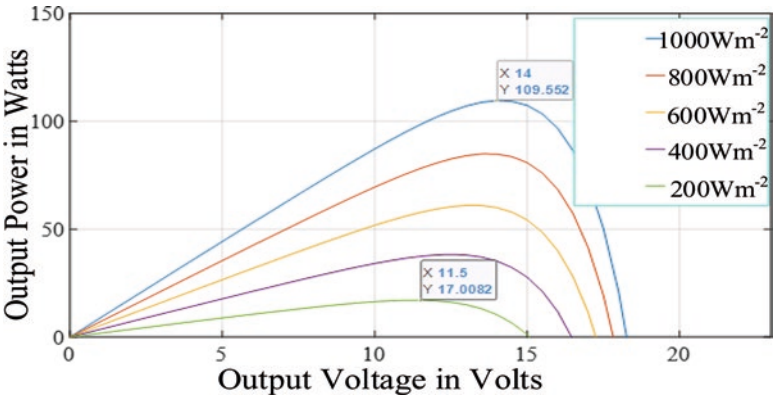


Fig. 4.9 A P–V plot of irradiance variations at 25 °C for solar panel

The current flowing through the diode is defined as

$$I_{\mathrm{D}} = I_{\mathrm{s}} \left\{ \exp\left[\frac{V + \mathrm{IR_S}}{\mathrm{KT_C}A} \right] - 1 \right\} \tag{4.23}$$

The shunt current is defined as

$$I_P = \frac{V + \mathrm{IR_S}}{R_{\mathrm{SH}}} \tag{4.24}$$

The power generated, $P = \mathrm{IV}$, where $I = I_{\mathrm{L}}$, $V = V_{\mathrm{OC}}$,

$$P = V \left\{ I_{\mathrm{PV}} - I_{\mathrm{S}} \left[\exp\left(\frac{q\left(V + \mathrm{IR_S}\right)}{\mathrm{KT_C}A} \right) - 1 \right] - \frac{V + \mathrm{IR_S}}{R_{\mathrm{SH}}} \right\} \tag{4.25}$$

where I_{PV} = Generated Current, I_{L} = Load Current, I_{S} = Reserved saturated Current of the Diode Parallel resistance, I_{P} = Current flowing through, V = Voltage across the PV Cell, T_{C} = Operating Temperature, q = Charge of an electron (1.6×10^{-19}C), K = Boltzmann's constant (1.38×10^{-23}J/K).

A typical voltage across PV cell terminals is between 0.4 V and 0.5 V dependent on the operating ambient solar irradiance and temperature [18]. In an optimal design, when a load is connected to the cell, the voltage could be 0.25 V or 0.35 V [18]. The influence of ambient irradiance variations on the relationships among voltage, current, and power is demonstrated in Figs. 4.8 and 4.9.

4.3.1 *Influence of Ambient Solar Irradiance on PV Cell*

The simulated plots in Figs. 4.8 and 4.9 demonstrated that the performance of the solar panel strongly depended on ambient solar irradiance. The plots showed that the maximum power varies to change in ambient solar irradiance. In the simulated 100 W, 8.95(I_{SC}) and 22.8(V_{OC}) solar panel, the highest power and current generated were 109.552 W and 8.8539 A, respectively, from 1000 Wm^{-2} irradiance, while the lowest power and current were 17.0082 W and 1.7708 A, respectively. The output voltage of a PV depends on the ambient irradiance slightly and in a logarithmic trend [19–21].

4.3.2 *Influence of Ambient Temperature on PV Cell*

Solar panel changes with variations of ambient temperature based on Figs. 4.10 and 4.11. The variations of a temperature inversely affect the voltage, while the current increases slightly with an increase in temperature [22, 23]. Figure 4.11 demonstrates that the PV output power increased with decreased ambient temperature.

4.3.3 *Maximum Power Point Tracking*

The installation of a solar panel is costly compared to conventional utilities on an interim basis, but in the long term, it is cheap and eco-friendly if the constraints of PV cells are optimized. Thus optimization will reduce the output cost and optimize the inputs to maximize power output despite irregular ambient irradiance and temperature levels. However, the relationship between current, voltage, and power plotted in Figs. 4.12 and 4.13 demonstrated that they have non-linear behaviors, and the tracking of the maximum power will be tricky.

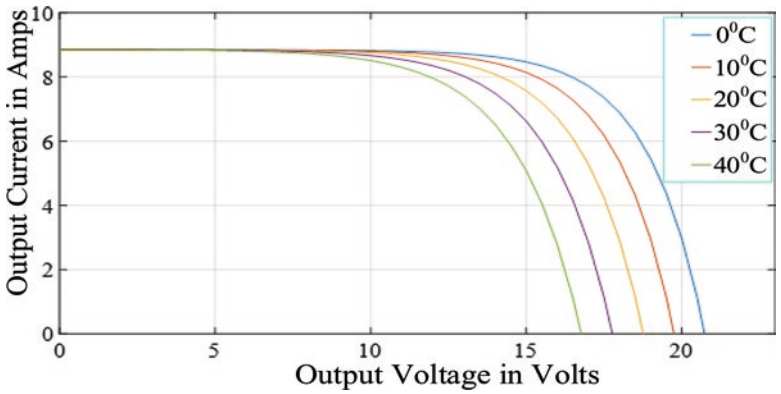


Fig. 4.10 I–V plot of temperature variations at fixed irradiance for 150 W solar panel

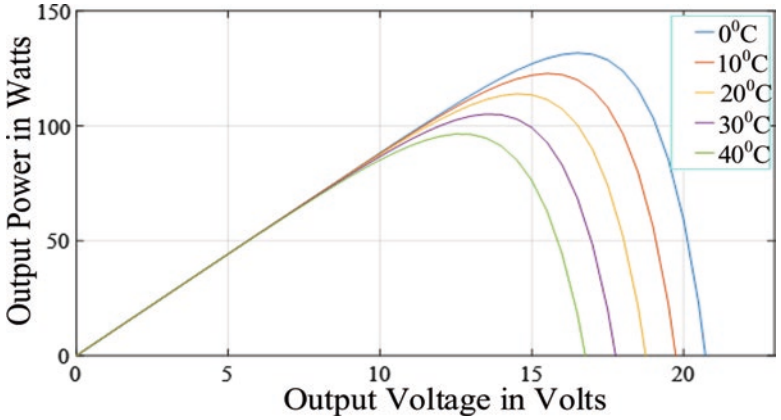


Fig. 4.11 A P–V plot of temperature variations at fixed irradiance for 150 W solar panel

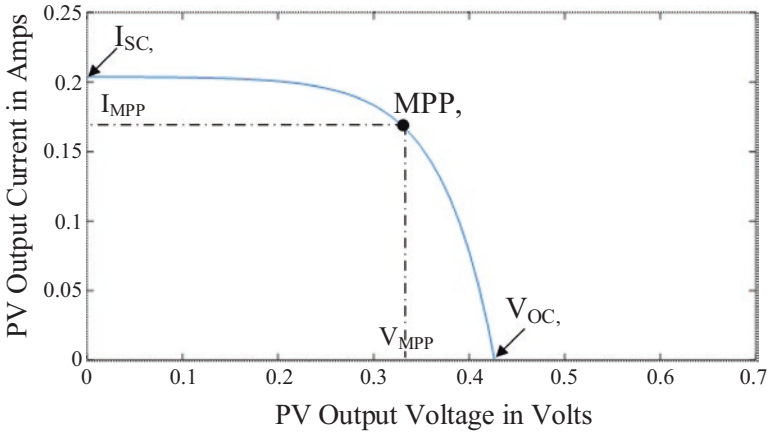


Fig. 4.12 I–V relationship of PV cell

The curves illustrating PV generated current running from 0 A to the short-circuit current, I_{SC} varying in respect to PV terminal voltages running from 0 V to open-circuit voltage, V_{OC}. Also, the curves revealed the tracking point of generated current I_{MPP} and voltage $V_{MPP,}$ which matches the maximum power P_{MPP}. Optimized operations of PV cells need to be performed to achieve maximum power transfer. However, to maintain maximum power transfer, an MPPT consisting of an electronic circuit and algorithm is integrated, as shown in Fig. 4.14. It is an intelligent sub-system of the photovoltaic system which controls the non-linear behavior of the solar panel under irregular ambient irradiance and temperature levels.

Several MPPT algorithms exist in literature such as perturb and observe (P&O), incremental conductance, fuzzy logic, neural network, which aimed at differentiating results depending on sensed quantities and experimental conditions [24]. In this

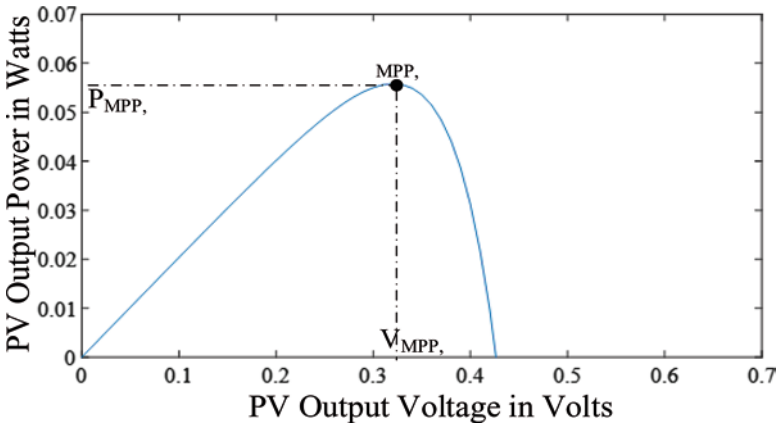


Fig. 4.13 P–V relationship of PV cell

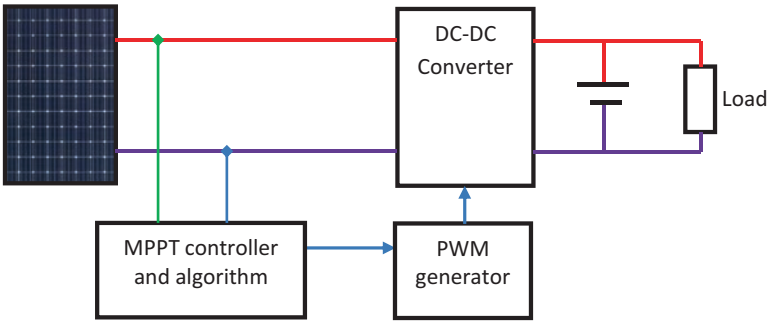


Fig. 4.14 P–V relationship of PV cell

work, we chose the P&O method as it is accepted to be the most efficient technique in most cases. A simplified flowchart of the algorithm is given in Fig. 4.15.

The algorithm is implemented in PSIM when the solar panel follows the average irradiance measured in the location of the project. The MPPT uses a buck converter designed to fulfill the relationships between the input and output quantities, as shown in Fig. 4.16. In this case, the solar arrays have an output voltage of 90 V, and the output battery needs 24V. A smoothing inductor is connected to the battery to limit its current ripple to 0.2 A. The 24 V battery bank should supply both DC and AC loads in the house, and for this purpose, a single-phase inverter combined to a 24 V–230 V transformer is used to provide AC loads. The complete system given in Fig. 4.17 is terminated by a fourth-order Chebyshev low pass filter, which cancels the high-frequency harmonics from the square signal given by the inverter. The ideal transformer has a transformation ratio of 325/24 to guarantee suitable voltage for most AC appliances in the house.

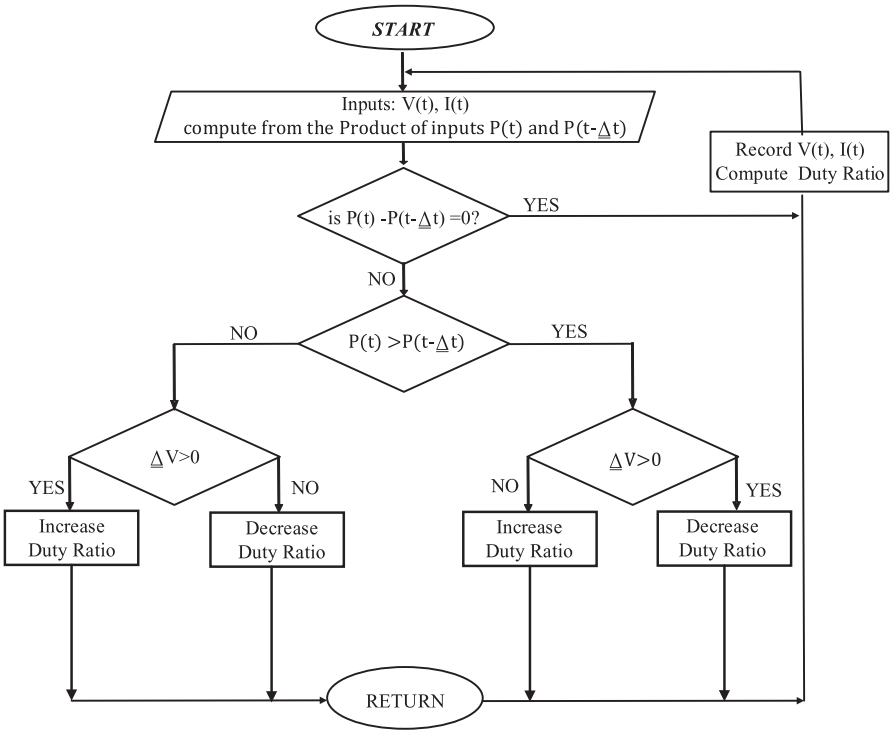


Fig. 4.15 Basic Perturb and Observe algorithm

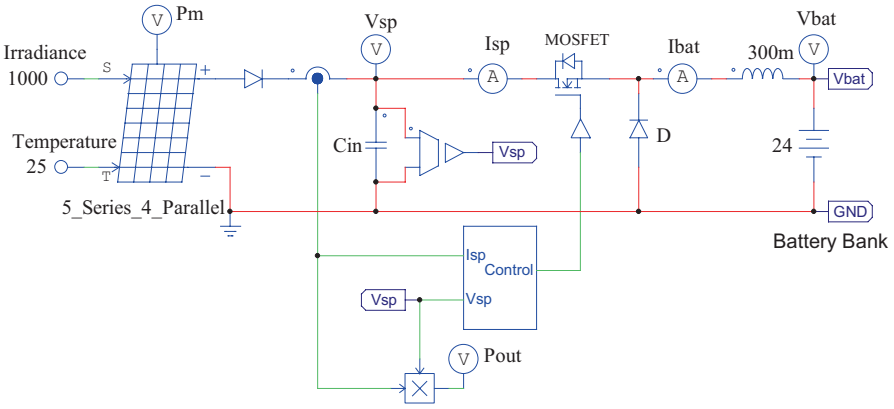


Fig. 4.16 Implementation of Perturb and Observe algorithm by simulation

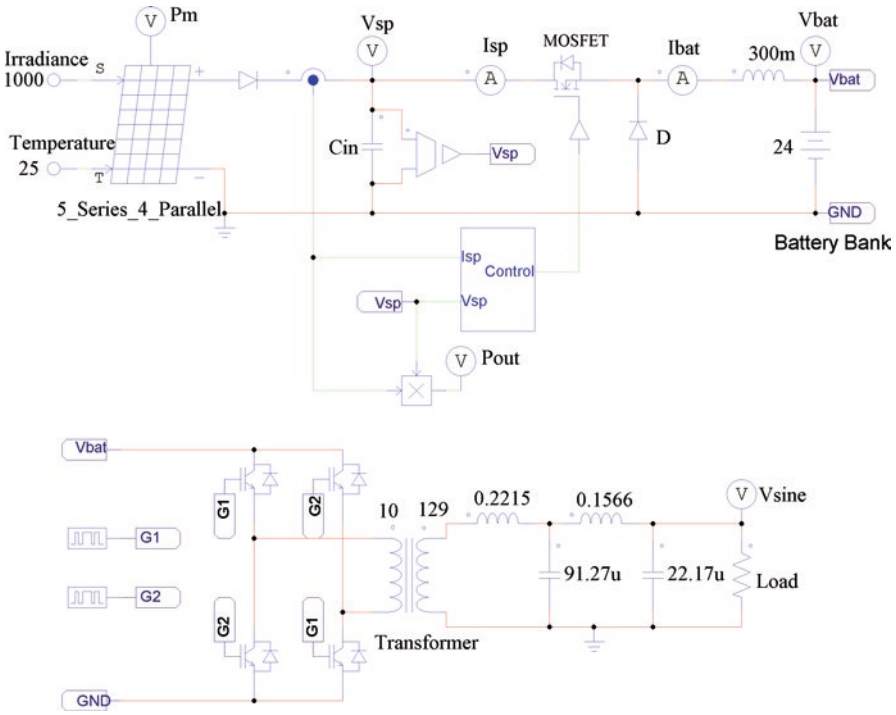


Fig. 4.17 24 VDC and 230 VAC outputs from the solar system

4.4 Photovoltaic Modules Connected to a Load via Battery

4.4.1 Lead Acid Battery

Currently, charging and discharging of electrical energy in energy bank are the straight forward achievement with batteries and lead-acid battery dominates among its kind [25]. For a standalone PV system, the battery is commonly used as an energy bank and typically has a lifecycle of 2–3 years. Conventionally, the off-grid system powered by solar panels lead-acid battery is the main energy bank to regulate, store the excess electrical energy from the panel, and deliver when needed. The lead-acid battery consists of about six cells of about 2 VDC, each connected in series to make about 12 VDC. The batteries encounter numerous operational constraints such as self-discharge, variation in ambient temperature levels, lifecycles, the mode of regulations of a DC–DC converter. Optimizing these constraints determine the performance of the battery and its lifespan. Therefore, it is important to note that energy bank in PV system requires to be well designed to avoid shorter lifecycles and frequent failures. It uses voltage based processes for charging, and it takes 12–16 h to charge fully, and higher charge currents reduce the charging time [26]. Additionally, the open-circuit voltage V_{OC} of the battery is directly proportional

to the electric charge Q stored in a period t. The fully charged battery with its highest aggregate of charges $Q_{H,}$ and state of charge (SoC) at time t is expressed as [27]:

$$\mathrm{SoC}(t) = \frac{Q(t)}{Q_{\mathrm{H}}(t)} \tag{4.26}$$

4.4.2 Battery Charge and Discharge System

Currently, the essence of the battery charge and discharge in the off-grid system is to replenish the depleted electrical energy and regulate the supply. Its optimal target is to reduce the electrical energy drawn from the battery to extend the lifetime of the energy bank.

From Fig. 4.18, the generated current can be expressed in Eq. (4.27). The figure has three main units: the energy harvesting source, battery, and the load. An example of battery is the lead-acid battery, and a 12 VDC lead-acid battery of 2.45 V per cell contains six cells, which produces 14.7 V across its terminal. At full charge, the battery voltage will get to 14.7 V, as shown in the charging curve of Fig. 4.19. At an initial voltage of 12.96 V, the battery gradually increased to the highest value of 14.7 V and remains nearly constant, depending on the age or self-discharging nature of the battery. In the discharging curve in Fig. 4.20, from the 14.7 V, it reduced gently equally depending on the age, self-discharge, and operating temperature of the battery.

$$I_{\mathrm{pv}} = I_{\mathrm{batt}} + I_{\mathrm{load}} \tag{4.27}$$

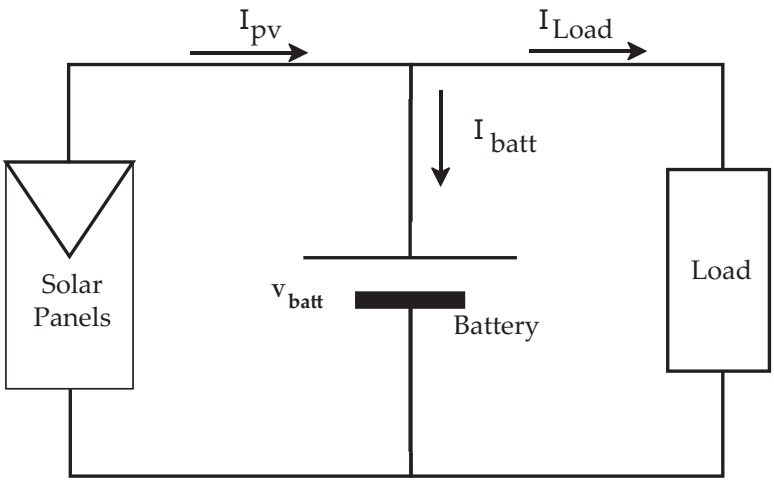


Fig. 4.18 A diagram of the off-grid system powered by solar panel

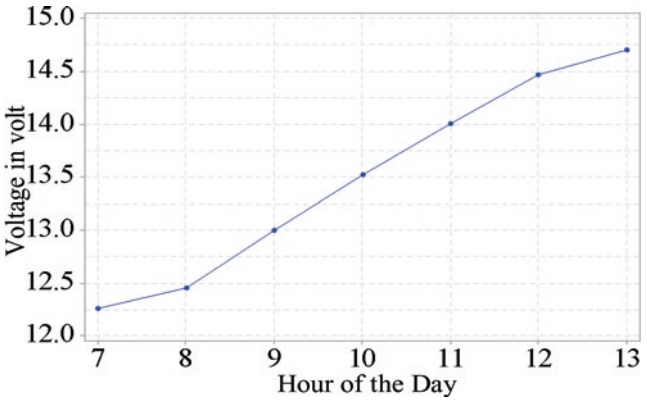


Fig. 4.19 Charging behavior of 200 Ah–12 V Lead-Acid Battery

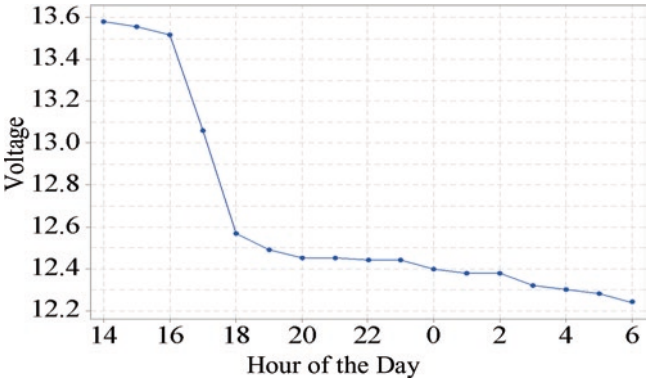


Fig. 4.20 Discharging behavior of 200 Ah–12 V Lead-Acid Battery

The lead-acid battery in operation is made up of positive electrode, lead dioxide (PbO_2), and the negative electrode, lead (Pb) in an electrolyte (H_2SO_4) [28]. The forward direction in Eq. (4.28) represents chemical reactions that occur during discharge, where electrical energy is delivered to the load. Charging occurs in the backward direction in Eq. (4.28), where electrical energy is taken from the energy source.

$$\mathrm{Pb} + \mathrm{SO_4^{2-}} \underset{\text{charge}}{\overset{\text{discharge}}{\rightleftarrows}} \mathrm{PbSO_4} + 2\mathrm{e^-} \tag{4.28}$$

$$\mathrm{PbO_2} + \mathrm{SO_4^{2-}} + 4\mathrm{H^+} + 2\mathrm{e^-} \underset{\text{charge}}{\overset{\text{discharge}}{\rightleftarrows}} \mathrm{PbSO_4} + 2\mathrm{H_2O} \tag{4.29}$$

$$PbO_2 + Pb + 2H_2SO_4 \underset{\text{charge}}{\overset{\text{discharge}}{\rightleftarrows}} 2PbSO_4 + 2H_2O \tag{4.30}$$

4.5 Simulation Results

From a typical daily solar irradiance in Botswana, the system is simulated with only the battery to see if the MPPT algorithm stuck well to the theoretical maximum power. The solar panel and the battery voltages and currents are shown in addition to the power to see the overall performances. The switches are all supposed ideal to make the study easy and to focus on the main components of the circuit. Simulation in Fig. 4.21a showed that the MPPT performs well as its output power is in the same range of the theoretical output power with a maximum deviation of 60 W at midday. The battery voltage always remains at equal value during the day.

The general performance of the MPPT algorithm is also tested in Fig. 4.21b on a random irradiance to confirm its efficiency in terms of detecting the maximum power in any situation. The PV curve showed that the maximum power point is

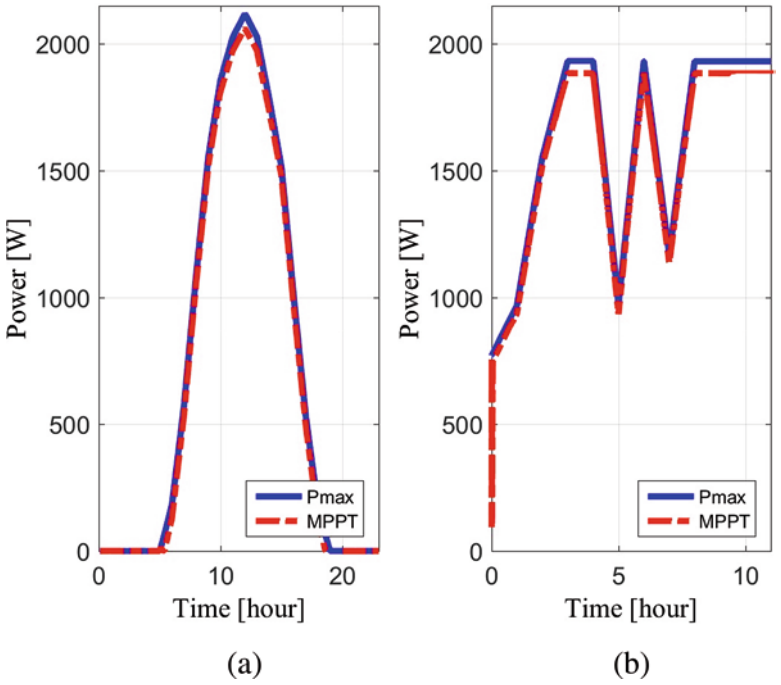


Fig. 4.21 MPPT performance on (**a**) a typical day of 24 h in Botswana and (**b**) a random irradiance

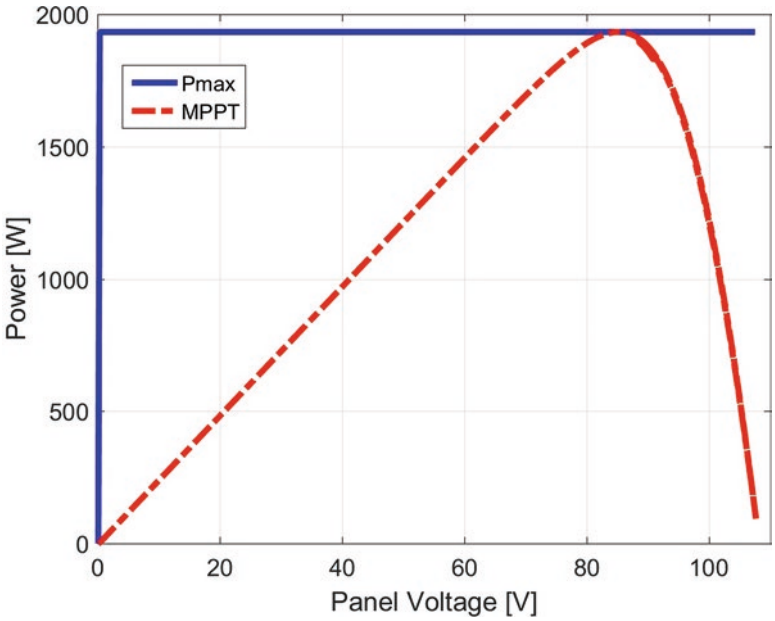


Fig. 4.22 The PV curve of the solar arrays showing the maximum power point

reached at V_{sp} = 85.19V in Fig. 4.22 with a power of 1933.17W corresponding to a current of 22.69 A.

The outputs of the system from simulation, for both DC and AC loads, led to satisfactory results with a constant 24 VDC and 230 VAC outputs without distortion in Fig. 4.23. The panel average output voltage is about 90 V. These results obtained from simplified models showed the feasibility of the green-house based on the same specifications as in the simulation. The losses caused by components will need to be included for the practical model, but the whole concept will be the same as in the simulations.

4.6 Summary

In this chapter, we report on the design and simulation of a standalone photovoltaic system to support the Internet of Things Sensor Devices, following modeling and sizing procedures. In most cases, the lack of or little knowledge of the sizing of the system's parameter causes a hazard to most of the houses that utilize renewable energy sources. Also, the efficiency of these panels that are used for the energy harvesting purpose depends on the sunlight, for which the amount of harvested energy is also influenced by ambient temperature and solar irradiation, and the entire system depends solely on Maximum Power Point Tracker (MPPT) system.

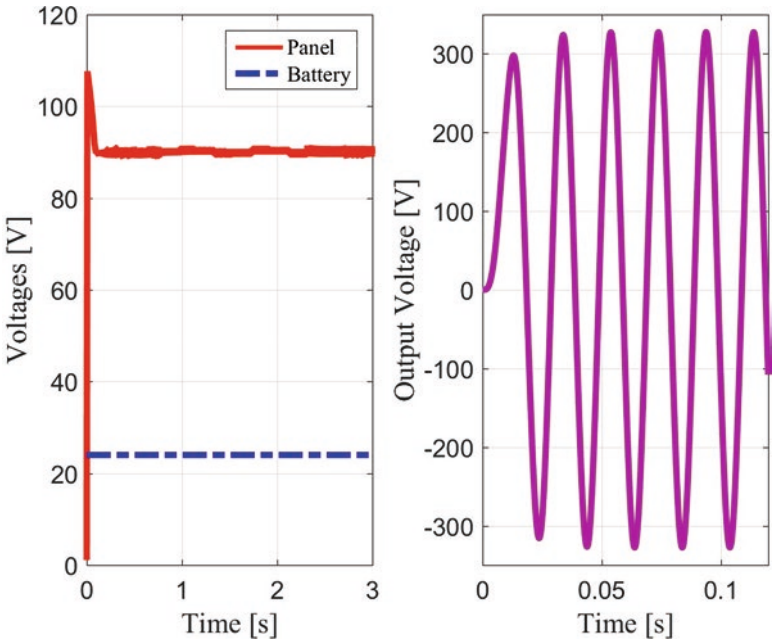


Fig. 4.23 The panel and the battery voltages and the output sinusoidal signal

Changes in solar irradiance and temperature conditions, in addition to variations in load demands, cause power imbalances. The design and consideration of MPPT help in compensation for the changes. This paper further reports on the procedures for determining the number of solar panels required to optimize the available space on the rooftop. This is generally a feasible and practical way of erecting solar systems as opposed to seeking alternative spaces to assemble the solar arrays. In general, the design of our standalone system was informed by the amount of energy the solar array system can produce under spatial constraints (available roof area), solar panel output, and footprint. As such, a PV standalone system with battery storage is designed and simulated, and the mathematical calculation for the intended 2000 W energy source is done, along with a proper layout of the design configurations. The simulation results also confirmed the mathematical models of the design.

References

1. N.O. Pearson, *Solar cheaper than diesel making India's mittal believer: Energy* (Bloomberg Technology, 2012)
2. T. Givler, P. Lilienthal, *Using HOMER software, NREL's micropower optimization model, to explore the role of gen-sets in small solar power systems case study* (National Renewable Energy Laboratory, Colo, 2005)

3. A.B. Kanase-Patil, R.P. Saini, M.P. Sharma, Sizing of an integrated renewable energysystem based on load profiles and reliability index for the state of Uttarakhandin India. Renew Energy **36**, 2809–2821 (2011)
4. D.J. Zimmerle, S.H. Kuppa, Statistical failure estimation method to size off-grid electrical systems for villages in developing countries, in *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, San Jose, 2017.
5. Africa Renewable Energy Access Program (AFREA), *Photovoltaics for community service facilities guidance for sustainability* (Africa Renewable Energy Access Program (AFREA), Washington, 2010)
6. T. Ma, H. Yang, L. Lu, Study on standalone power supply options for an isolated community. Elect. Power Energy Syst. **65**(2015), 1–11 (2015)
7. M.S. Cengiz, M. Mami, Price-efficiency relationship for photovoltaic systems on global basis. Int. J. Photoenergy vol. 2015, no. 2015, p. 12, 2015.
8. D. Gielen, *Renewable Energy Technologies Cost Analysis Series Solar Photovoltaics* (International Renewable Energy Agency, Bonn, 2012)
9. Four Peaks Technology. Solar Cell Central – Solar Electricity Costs. (Online.) Available: http://solarcellcentral.com/cost_page.html. Accessed 3 Mar 2020.
10. H.A. Sher, A.F. Murtaza, A. Noman, K.E. Addoweesh, K. Al-Haddad, New sensorless hybrid MPPT algorithm based on fractional short-circuit current measurement and P&O MPPT. IEEE Trans. Sustain. Energy **6**(4), 1426–1434 (2015)
11. L. Zhang, J. Yu, H. Ma, Y. Zhang, Design of photovoltaic power supply MPPT circuit for WSN node based on current observation. Int. J. Online Biomed. Eng. **14**(7), 45–61 (2018)
12. B. Pakkiraiah, G.D. Sukumar, Research survey on various MPPT performance issues to improve the solar PV system efficiency. J. Solar Energy (2016), 20 (2016, 2016)
13. Get Invest Mobilising Renewable Energy Investments, *Botswana Renewable Energy Potential* (Get Invest Mobilising Renewable Energy Investments, 2010)
14. L. Castaner, S. Silvestre, *Modelling photovoltaic systems using PSpice* (Wiley & Sons Ltd, Chichester, 2002)
15. POWERSIM, Powersim Technology, POWERSIM (Software for Power Electronics Simulation), 29 April 2018. [Online]. Available: https://powersimtech.com/products/psim/. Accessed 2 Mar 2020.
16. H. Sharma, A. Haque, Z.A. Jaffery, Modeling and optimisation of a solar energy harvesting system for wireless sensor network nodes. J. Sens. Actuator Netw. **7**(40), 19 (2018)
17. S. Guo, F. Ma, B. Hoex, A.G. Aberle, M. Peters, Analysing Solar cells by circuit modelling, in *PV Asia Pacific Conference 2011*, (Singapore, 2011)
18. QuantumSphere Inc., *MicroPower Step-up Low-Voltage Booster Module Enables Practical Energy Capture from Low-Power Generators* (QuantumSphere Inc., Sunnyvale, 2011)
19. M. Nasir, H.A. Khan, I. Khan, N. Hassan, N.A. Zaffar, A. Mehmood, T. Sauter, S.M. Muyeen, Grid load reduction through optimized PV power utilization in intermittent grids using a low-cost hardware platform. MDPI Energies **12**(9), 1–21 (2019)
20. A. Hu, Q. Sun, H. Liu, H. Zhou, Z. Tan, H. Zhu, A novel photovoltaic array outlier cleaning algorithm based on sliding standard deviation mutation. MDPI Energies **12**(22), 4316 (2019)
21. E. Batzelis, Non-iterative methods for the extraction of the single-diode model parameters of photovoltaic modules: A review and comparative assessment. Energies **12**(358), 26 (2019)
22. V. Jafari, M. Debghani, J.J. Fesharak, Practical implementation of the backstepping sliding mode controller MPPT for a PV-storage application. MDPI Energies **12**(18), 3539 (2019)
23. M.M. Sarafraz, M.R. Safae, A.S. Leon, I. Tlili, T.A. Alkanhal, Z. Tian, M. Goodarzi, M. Arjomandi, Experimental investigation on thermal performance of a PV/T-PCM (photovoltaic/thermal) system cooling with a PCM and nanofluid. MDPI Energies **12**(2572), 16 (2019)
24. T. Esram, P.L. Chapman, Comparison of photovoltaic array maximum power point tracking techniques. IEEE Trans. Energy Convers. **22**(2), 439–449 (2007)

25. B. Hariprakash, S.K. Martha, S. Ambalavanan, S.A. Gaffoor, A.K. Shukla, Comparative study of lead-acid batteries for photovoltaic standalone lighting systems. J. Appl. Electrochem. **38**(2008), 77–82 (2008)
26. S.K. Raheja, *Charging Information for Lead Acid Batteries* (Battery University, 2013)
27. C. Savard, E.V. Iakovleva, A suggested improvement for small autonomous energy system reliability by reducing heat and excess charges. Batteries **5**(1), 29 (2019)
28. J. Jung, L. Zhang, J. Zhang, *Lead-acid Battery Technologies: Fundamentals, Materials, and Applications* (CRC Press, Boca Raton, 2015)

Chapter 5
Security Challenges in IoT Sensor Networks

Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong

5.1 Introduction

Chapter 4 discussed the disadvantages of conventional sources of power supply and highlighted the need for an alternative photovoltaic system together with its advantages. A photovoltaic system was proposed, simulated, and designed as an off-grid and standalone 2000 W solar energy harvesting system that can be used to power the IoT devices. The powered IoT devices can be connected to sensor networks whose security can be compromised. Therefore, the goal of this chapter is to describe the security challenges at various layers (perception, network, cloud, and user interface) of a wireless IoT sensor network.

5.2 Background

IoT has promising areas of application in large sectors of the economy ranging from homes to retails and cities. IoT sensor nodes exist and operate in open and unattended environments and communicate through wireless communication mediums. IoT devices gather much sensitive information such as bank card details, names, health information, and so forth. The information is then relayed among sensor nodes and sent to the cloud where all the processing takes place. The transmission of information over IoT network without adequate security measures poses a challenge for possible unauthorized access to information. IoT devices gather much sensitive information such as personal, health, and bank card details; then, the information is transferred to the cloud where it is processed and stored. Therefore, it is essential to protect sensed information and secure transmission of the sensed data to the gateway nodes for further processing and to protect sensor data stored in the cloud. IoT sensor networks experience security and privacy issues in devices, during communication, during the processing of data and in the storage. Therefore, the

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_5

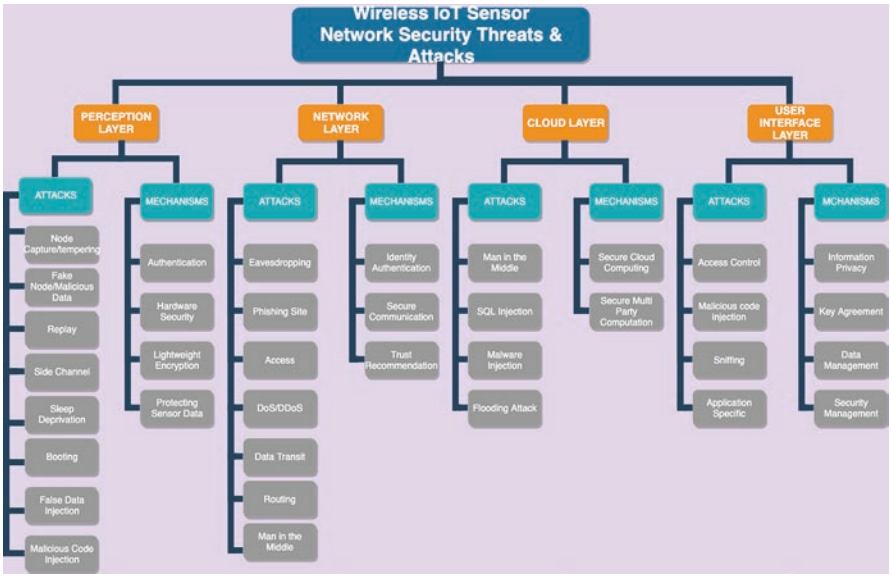


Fig. 5.1 IoT sensor networks security threats and attacks

security and privacy of IoT sensor networks remain a significant challenge that needs to be addressed [3–5]. The number of threats and vulnerabilities equally vary depending on the nature of the application domain. This section presents an overview of research issues and challenges in IoT sensor networks through the 4 tier IoT architecture. Figure 5.1 shows the threat landscape and layout of the work in this chapter.

5.3 Perception Layer

The perception layer consists of resources-constrained IoT devices such as sensors, RFID tags, Bluetooth, and Zig-Bee devices which are placed outdoors and exist in environments that expose them to natural accidents and physical attacks [6]. The nodes or devices are distributed, and attackers can physically gain access to the devices and compromise the nodes. The results of the HP survey reflected that more than 60% of the IoT devices had vulnerabilities while upgrading their software and hardware resulting from inadequate and inefficient encryption standards. The results prove that IoT devices remain vulnerable to malicious software attacks during software updates or hardware upgrades.

5.3.1 Node Capture/Tempering

The attacker gains physical access to sensor nodes, then manipulates the node by replacing it or altering a piece of hardware or extracting sensitive information. Sensitive information in a sensor node that may be compromised may be information on cryptographic keys or routing table information, hence endangering the security of the entire network [7]. Securing IoT based wireless sensor networks is more challenging for mobile sensor nodes in the network; therefore, mobility-based detection is an emerging area that can be considered for IoT wireless sensor networks.

5.3.2 Fake Node and Malicious Data

An attacker may launch malicious data through a fake node into the existing IoT sensor networks through which they can circulate malicious codes and information throughout the network [7]. A malicious node is physically injected between two or more nodes in the network. The malicious node then modifies data and passes wrong or compromised information to other nodes. The adversary launches a man-in-the-middle attack. As a result, constraining and consuming energy of low-power devices and compromise the security of the whole system. For example, the attacker inserts a replica of Node A and another malicious node (Node A1), which works together to execute the attack causing a collision at the victim node [8].

5.3.3 Replay Attack

A replay attack, sometimes called a playback attack, occurs when an attacker detects and intercepts secure network communication or data transmission and fraudulently gets it delayed or retransmitted to access data, systems, or transactions. An attacker may not acquire the data information but can replay earlier packets received from other nodes [9]. A replay attack, also known as "man in the middle attack," is a security breach whereby information is stored without authorization and may be retransmitted back to the receiver to mislead them for unauthorized operations such as false identification or authentication or duplicate transactions. They can be used to fool financial institutions into duplicating transactions allowing attackers or hackers to draw money directly from their clients' accounts.

5.3.4 Side-Channel Attack

A side-channel attack aims at extracting secrets from a chip or system through measurement and analysis of physical parameters such as timing information, power consumption, electromagnetic leaks, or even sounds. This attack is the most difficult

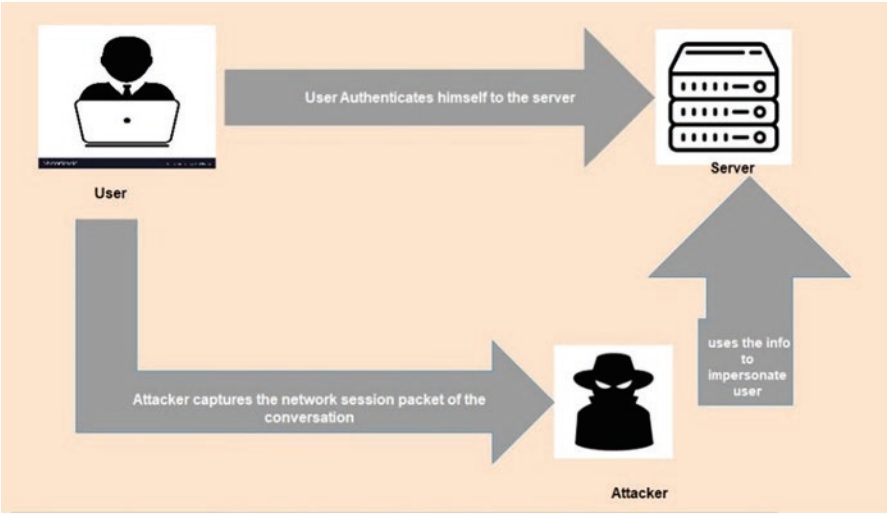


Fig. 5.2 Side-channel attack

to handle, as it is difficult to detect. A side-channel attack is implemented based on information gained from a computer system rather than the weakness of an implemented algorithm itself. Henceforth, side-channel attacks require technical knowledge of the internal operating system, and the most powerful side-channel attacks are based on statistical methods. In practice, cryptosystems are implemented on physical devices, and during cryptographic computations, physical devices reveal some information in terms of energy dissipation and time computation (side-channel information). The attacker uses this information to determine secret keys and to break the cryptosystem. Figure 5.2 presents the side-channel attack.

5.3.5 DDoS/DoS

Denial of Service (DoS) attack is a common attack in IoT that shuts down the entire system or network by preventing authorized users from accessing it. This attack is achieved by overwhelming the system or network by causing traffic with large amounts of spam requests, hence overloading the system and hindering it from performing at its optimum best. The services provided by the system or network become unavailable to the intended users temporarily or indefinitely disrupting services of a host connected to the Internet. In a distributed denial of service (DDoS), multiple sources are directing a lot of traffic to flood a target system or service, making it difficult to counterfeit the cyber-attack [9]. For example, in IoT sensor networks with online consumer devices such as IP cameras and home routers, remotely controlled bots can be launched into the network and used as part of a botnet in large-scale network attacks

5.3.6 Sleep Deprivation

Sensor nodes in an IoT sensor network are battery powered and attempt to remain in low-power sleep mode without adversely affecting the performance and applications of the node. Sensors have a limited battery life and follow a sleep pattern to maximize battery life. An attacker interacts with a victim node in a manner that makes it seem like a legitimate node with the aim to interact with it, consequently keeping the victim node out of its power conversation sleep mode [10]. The intruder node cases the victim node to stay woke, resulting in the depletion of the battery life of the victim node. Consequently, the intruder dramatically maximizes the power consumption of the victim node while minimizing its lifespan and makes it very difficult to detect and immensely cause drainage of energy on legitimate nodes, ultimately leading to the death of these nodes.

5.3.7 Booting Attack

Edge devices are typically low powered and often go through the sleep–wake cycles, making them vulnerable to booting attacks during the booting process. Security processes of these edge devices are not usually enabled during the booting process, and attackers usually take advantage of this vulnerability and try to attack these nodes when they are restarted [11].

5.3.8 False Data Injection

False Data Injection Attacks (FDIA) are major cyber threats in power systems. FDIA's compromise meter measurements by matching alterations in the results of state estimation without being detected by traditional bad data detector. The attack severely threatens the security of the smart grid. The intruder aims to destroy the stability of power grids and gain illegal profit by tampering with data of power equipment [12]. The intruder falsifies the network topology, which misleads the control center and ultimately disrupts state estimation.

5.3.9 Malicious Code Injection

An intruder introduces malicious code into a vulnerable computer program and modifies the mode of execution. This attack is achieved by an attacker launching an executable input to a program and trick the software into running that input. Code injection can be used for a variety of purposes, such as stealing data, taking full control of the system, and propagate worms. The common types of attacks include

shell injection and HTML script, which can both control and compromise the privacy of the user and completely shut the system down [13].

5.4 Network Layer

The IoT hardware or sensors that collect data using built-in sensors communicate via machine-to-machine communication to transmit data using device modules through network services. Attacks on the network layer affect the coordination of work and information sharing among devices in an IoT sensor network. The network layer is vulnerable to information privacy and denial of service attacks.

5.4.1 Eavesdropping

Attacks in eavesdropping involve a weakened connection between a client and a server that allows an intruder to redirect traffic to itself. An eavesdropping attack is performed by an attacker installing a network monitoring software (sniffer) on a computer and intercepting data during transmission. A typical example of an eavesdropping attack is a product attached with an RFID tag that passively communicates with RFID readers. In this scenario, confidential communication between an RFID reader and the RFID tag can be easily wiretapped since it is difficult to apply encryption mechanisms to the RFID reader and tag due to their limited computational capability and energy constraint [14]. Henceforth, lightweight security mechanisms need to be implemented for similar IoT devices.

5.4.2 Phishing Site

This is a cyber phishing attack whereby an attacker masquerades as a trusted or reputable entity. The attacker uses phishing emails and websites to distribute malicious links or attachments to perform a variety of actions such as extraction of login credentials or account information from victims. Phishers use social engineering ad public sources of information such as social networks like Facebook, Twitter, or LinkedIn to gather background information on the victim's data such as work history, interests, and activities. The attackers then use this information to craft a deceptive email containing malicious links or attachments to gain access to confidential data by spoofing the authentication credentials of users [15].

5.4.3 Access Attack

An access attack defines an attempt to access a network device or user account through improper means. If proper security measures are not in place, the network may be left vulnerable to a variety of intrusions. In an access attack, the attacker bypasses some authentication process through password attack, trust exploitation, and man-in-the-middle attack. The intruder gains access to a system or network and explores vulnerabilities in network authentication, FTP, and web services to gain access to web accounts and confidential information.

5.4.4 DoS/DDoS

The network layer is susceptible to the denial of service (DoS) attack, which makes the device, resource, or network unavailable to authorized users by temporarily or indefinitely disrupting services of a host connected to the Internet. DoS typically uses one computer and one Internet connection to flood a target system, whereas DDoS uses multiple computer and Internet connections to flood the target node or source [1]. In 2016 a distributed denial of service attack was launched targeted at Dyn Company's Domain Name System (DNS) provider through a botnet that contained vulnerable IoT devices such as IP cameras and printers that were infected with Mirai malware.

5.4.5 Data Transit

IoT applications have a high rate of data movement through sensors, actuators, and different IoT applications. Data disseminated by IoT devices and applications is valuable, moves from one area to another, and is always targeted by hackers and adversaries. Data stored in the cloud poses security risks, and data in transit is equally vulnerable to cyber-attacks [11]. Eavesdropping is an example of an attack of attack whereby an attacker uses a program device or sniffer to capture data on transit.

5.4.6 Routing Attack

Routing is a crucial process in any type of network responsible for transporting a data packet from source to destination through an optimal path. Routing attacks have been viewed from the perspective of an attack on the availability of Internet applications. The attackers may capture the node to extract all the cryptographic information and utilize it to perform illegal work in the network. Attackers may implement illegal, malicious code to break some routing rules and modify the routing table [2]. An intruder may also hijack Internet traffic towards a victim application server and

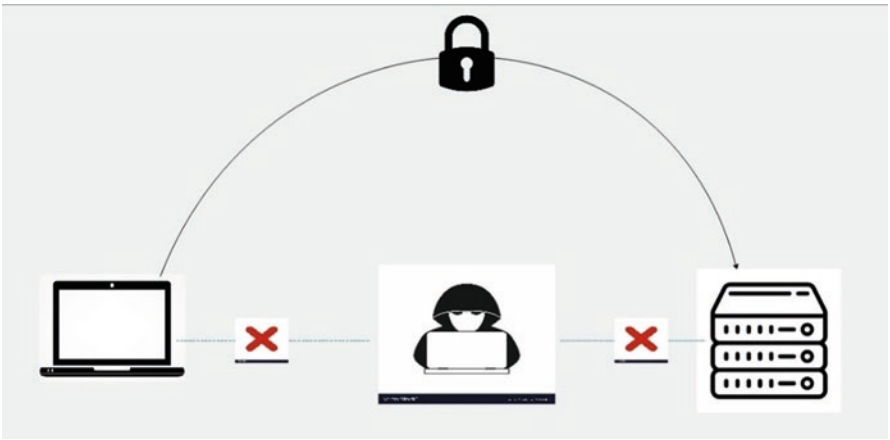


Fig. 5.3 Man-in-the-middle attack

cause the unavailability of the services rendered by the application [3]. Several routing protocols have been proposed to govern communication in a network; unfortunately, most of the routing protocols were not designed with security in mind.

5.4.7 Man-in-the-Middle Attack

An intruder injects themselves into a conversation between two entities and secretly relays and alters the communication between two parties who believe they are directly communicating with each other. For example, an attacker may place himself between two interconnected devices. Therefore, network traffic flowing between the devices will flow through the attacker's machine, allowing him to intercept, read, and modify contents. An example of man-in-the-middle (MITM) attack is eavesdropping, which occurs when a malicious node injects itself in a communication session between two devices in a network exploiting the data transmission process and processing of real-time transactions [4]. For example, devices equipped with wireless cards may try to connect to an access point with the strongest signal. Hence, intruders may set up a wireless access point and deceive devices to join their network and gain access to a victim's network traffic. Figure 5.3 presents a man-in-the-middle attack.

5.5 Cloud Layer

The cloud layer is responsible for resources, applications, and service management. The cloud layer makes use of technologies such as multi-tenancy and virtualization to make efficient use and management of resources and applications. One

server, data center, or operating system may be used to host several users on the cloud through virtualization. Despite the several advantages brought about by cloud computing, several practical problems need to be addressed mainly related to service-level agreements (SLA's), security privacy, and energy efficiency [5]. However, data in the cloud is susceptible to security threats and vulnerabilities such as man-in-the-middle attack, SQL injection, malware injection, and flooding attack.

5.5.1 Man-in-the-Cloud Attack

Cloud storage has over the years become the best method for sharing, backing up, and remotely access data from anywhere and anytime. However, cloud security is a prevalent issue that has called for attention in the area of IoT Security. Man-in-the-cloud (MITC) attacks rely on file synchronization processes of file-sharing tools such as Dropbox, Google Drive, and OneDrive as their infrastructure for command and control (C&C) and remote access [6]. These cloud-based tools use OAuth tokens to validate users. However, intruders may phish users and take OAuth tokens from users' machines. The intruder may place the token on their devices, and the file-sharing tool will synchronize shared data to their device as well, making it possible for the intruder to synchronize and maintain victims' accounts from anywhere. It is difficult to detect this threat as no malicious code can be detected or abnormal outbound traffic. Therefore, enough measures for detecting or militating against this type of attack need further investigation. Figure 5.4 presents a man-in-the-cloud attack.

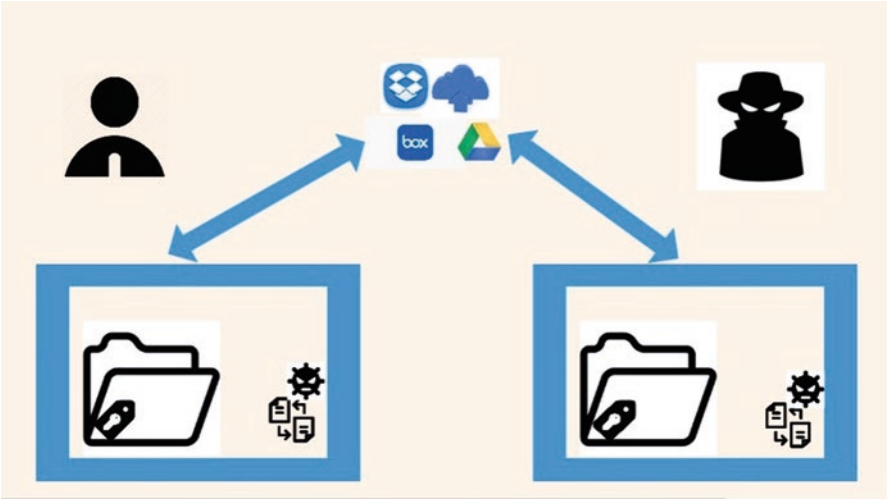


Fig. 5.4 Man-in-the-cloud attack

5.5.2 SQL Injection

SQL injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements to control a database server running behind a web application. Attackers bypass application security measures using SQL injection vulnerabilities. Attackers manipulate authorization and authentication scripts for a web page or application, impersonate authorized users to a specific database, and retrieve the content of the entire SQL database. SQL injection allows unauthorized access by the intruder to personal data and to add, modify, or delete records in a database server or the internal network through an operating system hosted on the database server. SQL inject is one of the oldest, prevalent, and most dangerous web application vulnerabilities.

5.5.3 Malware Injection

Malware, commonly known as malicious software attacks are a common man-in-the-middle attack that allows cybercriminals to create malicious software and install it on user computers without their knowledge in an attempt to gain access to personal data or to damage the computer system [7]. The malware is injected into the computer system, and it installs itself into the browser, then records data that is sent between the victim and targeted websites, and transmits it back to the attacker. Malvertising and social engineering are conventional methods used by cybercriminals to inject malware into a victim's computer systems. In Malvertising, the attacker purchases legitimate advertising space on legitimate websites and embeds malicious code within the advertisements. Social engineering is a popular malware injection method that involves the manipulation of human emotions using social media, email instant messaging, etc., to manipulate the user into downloading malware or clicking a link to a compromised website that hosts malware. Malware encompasses various forms of attack such as ransomware, spyware, command and control, and more.

5.5.4 Flooding Attack

Flooding attack is a type of denial of service (DoS) attack whereby the intruder aims to bring the network service down by flooding it with spurious messages to cause massive amounts of traffic. The intruder floods the network by initiating continuous incomplete connection requests by flooding a server or host with connections that cannot be completed [1]. The host's memory buffer is eventually filled and can no longer accept further genuine connection requests resulting in a denial of service (DoS). Figure 5.5 presents the HTTP flood attack.

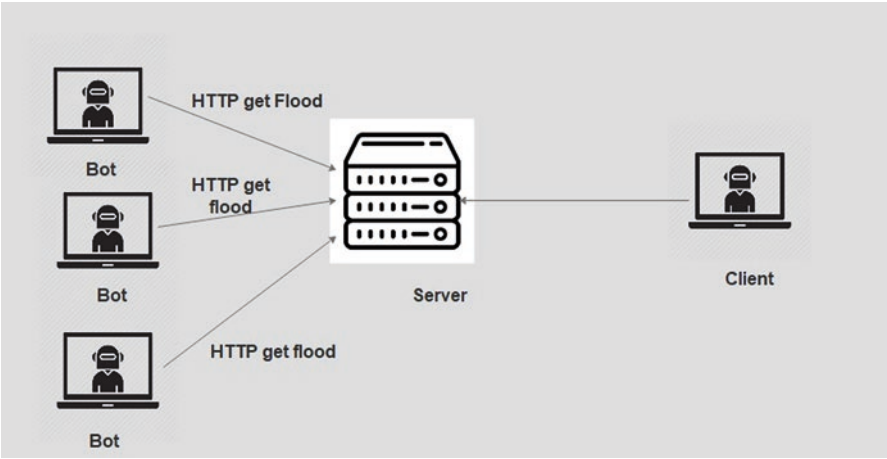


Fig. 5.5 HTTP flood attack

5.6 User Interface/Application Layer

This section presents attacks in the application layer.

5.6.1 Access Control Attacks

The access control process includes identification, authentication, and authorization processes. However, broken access control is a common web application vulnerability. An access control attack is a type of spoofing attack whereby an attacker impersonates a user to gain access to restricted resources in an attempt to steal information and perform malicious activities [6]. An attacker may impersonate the Internet Protocol (IP) address of a legitimate user to gain access to their accounts. A brute force or dictionary attack is an example of an access control attack where an attacker attacks every possible combination of numbers, letters, and characters to crack a password or PIN. Another example of an access control attack is a logic bomb that is installed by a privileged user who knows what security controls need to be circumvented to go undetected until they explode.

5.6.2 Malicious Code Injection

Malicious code injection, also known as "Remote Code Execution (RCE) attack," enables an attacker to execute malicious code through code or command injection. Code injection occurs when an application processes a code without validating its data. The code is injected into a vulnerable computer program and modifies the way

the program executes [13, 15]. The attacker capabilities depend on the server vulnerabilities, for example, PHP, Python, etc. Once the attacker can inject malicious code or commands, they may attempt to use a web shell or install malware to compromise other internal systems. Mirai malware is an example of a code injection attack that uses unsafe accessing credentials in the form of cookies, forms, HTTP protocol to enter and manipulate an operating systems shell file. The entire operating system's central command directive is manipulated and executes attacker-supplied commands in the system. The intruder may access sensitive information from a network, including email traffic (SMTP, POP, IMAP), web traffic (HTTP), FTP traffic (Telnet, FTP passwords, NFS), and many more.

5.6.3 Sniffing Attacks

Sniffing attack corresponds to theft or interception of data by capturing network traffic using sniffer [15]. A sniffer is an application aimed at monitoring, analyzing, and capturing network packets. Data packets that flow through a computer network, if the packets transmitted across a network are not encrypted; they get captured by a packet sniffer to launch the sniffing attack. Packets with weak encryption mechanisms are vulnerable to this type of attack. Various packet sniffers exist such as Wireshark, Dsniff, Etherpeek, kismet, etc., are responsible for stealing data, analyze it, or cause network crash or corrupt the data. Sniffing can also be performed using applications, hardware devices at both the host and network levels.

5.6.4 Application-Specific Vulnerabilities

Application-specific vulnerabilities are flaws in applications that could be exploited by intruders to compromise the security of an application and facilitate cybercrime activities. The cybercrime activities usually target confidentiality, integrity, or availability "CIA triad" of resources possessed by a specific application. According to Gartner security, the application layer currently contains 90% of vulnerabilities such as SQL injection, cross-site scripting, LDAP injection, etc.

5.7 Summary

In this chapter, we unveil security issues and challenges at various layers (perception, network, cloud, and the user interface) of IoT sensor networks. IoT has promising areas of application in vast sectors of the economy, including health, agriculture, oil and gas, logistics, retail, smart cities, and smart homes. Despite the tremendous benefits of IoT sensor networks, security remains a challenge in IoT. This makes

communication vulnerable to attacks and threats. Security in the Internet of Things is a fundamental issue. However, achieving the goal of a secure IoT sensor network remains a significant challenge because of the open and wireless nature of the sensor network. Also, IoT devices are resource-constrained in nature in terms of memory capacity and energy efficiency, making the goal of security challenging to achieve. The IoT sensor network security can be compromised through the devices, during communication, data processing, and storage commonly on cloud services. From the survey, the perception layer security challenges include fake node, node tempering, replay, side channel, false data, and code injection attacks. Common attacks in the network layer include eavesdropping, phishing site, access, denial of service, and distributed denial of service attack. Cloud layer attacks include a man-in-the-middle, SQL and malware injection, and lastly the user interface layer is susceptible to access control, malicious code injection, sniffing attacks and applications-specific vulnerabilities such as SQL injection and cross-site scripting which exploit the nature of the application in an IoT sensor network.

References

1. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4) (2015). https://doi.org/10.1109/COMST.2015.2444095
2. 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. Tutorials **17**(3), 1294–1312 (2015). https://doi.org/10.1109/COMST.2015.2388550
3. S. Nisha, M. Farik, RSA public key cryptography algorithm-a review. Int. J. Sci. Technol. Res. **6**(July), 7 (2017). [Online]. Available: www.ijstr.org.
4. M.A. Ferrag, L.A. Maglaras, H. Janicke, J. Jiang, L. Shu, Authentication protocols for Internet of Things: A comprehensive survey. Secur. Commun. Netw. **2017** (2017). https://doi.org/10.1155/2017/6562953
5. S. Madakam, R. Ramaswamy, S. Tripathi, Jcc_2015052516013923. J. Comput. Commun. (May), 164–173 (2015). https://doi.org/10.4236/jcc.2015.35021
6. I. Ali, S. Sabir, Z. Ullah, Internet of Things security, device authentication and access control: A review. IJCSNS **14**(8), 456–466 (2019). [Online]. Available: http://arxiv.org/abs/1901.07309.
7. A. Tewari, B.B. Gupta, Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. Futur. Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.04.027
8. J. Deogirikar, A. Vidhate, Security attacks in IoT: A survey. Proc. Int. Conf. IoT Soc. Mobile Anal. Cloud I-SMAC **2017**, 32–37 (2017). https://doi.org/10.1109/I-SMAC.2017.8058363
9. A.A.A. Ari et al., Enabling privacy and security in cloud of things: Architecture, applications, security and privacy challenges. Appl. Comput. Informatics (2019). https://doi.org/10.1016/j.aci.2019.11.005
10. M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, R. Brooks, The sleep deprivation attack in sensor networks: Analysis and methods of defense. Int. J. Distrib. Sens. Netw **2**(3), 267–287 (2006). https://doi.org/10.1080/15501320600642718
11. V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access **7**, 82721–82743 (2019). https://doi.org/10.1109/ACCESS.2019.2924045

12. B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in Internet of Things. J. Netw. Comput. Appl. **84**, 25–37 (2017). https://doi.org/10.1016/j.jnca.2017.02.009
13. K. Chen et al., Internet-of-Things security and vulnerabilities: Taxonomy, challenges, and practice. J. Hardw. Syst. Secur. **2**(2), 97–110 (2018). https://doi.org/10.1007/s41635-017-0029-7
14. X. Li, J. Xu, H.N. Dai, Q. Zhao, C.F. Cheang, Q. Wang, On modeling eavesdropping attacks in wireless networks. J. Comput. Sci. **11**, 196–204 (2015). https://doi.org/10.1016/j.jocs.2014.10.006
15. D. Sopori, T. Pawar, M. Patil, R. Ravindran, Internet of Things: Security threats. Ijarcet.Org **6**(3), 263–267 (2017). [Online]. Available: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-6-ISSUE-3-263-267.pdf .

Chapter 6
IoT Sensor Networks Security Mechanisms/Techniques

Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong

6.1 Introduction

Chapter 5 suggested a layered approach to unveil security issues and challenges at each layer of the IoT architecture. Overall in all IoT layers, vulnerabilities and security challenges are found in the perception, network, cloud, and finally, user interface layer. As suggested by authors [1–7], security must be implemented at each layer of the IoT architecture. Similarly, authors in [8–19] described and surveyed security issues in wireless systems. This section proposes security techniques and mechanisms to mitigate security issues and challenges through the layered approach. A layered approach makes fair use of intra-layer information, services, interfaces, and protocols with the aim to expose layer-specific challenges that need to be addressed for secure communication while reducing cost, energy consumption and optimize the entire network performance for robust and scalable wireless IoT sensor network applications. Each layer performs dedicated functionalities providing modularity and security for various interfaces. Experts forecast a total number of 20 billion connected things, people, processes, and data by the year 2020. This increase will pose a lot of vulnerabilities and security threats, calling for attention to various techniques for managing and protecting sensor data. In this chapter, various methods, mechanisms, and techniques for securing devices and communication of sensor data are discussed.

6.2 Perception Layer

This section presents security mechanisms for the perception layer.

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_6

6.2.1 Authentication

An increase in the number of IoT devices has led to a rise in IoT sensor networks security challenges and privacy concerns. Among these, IoT devices are prone to data breaches and attacks such as man-in-the-middle attack. Authentication of different connected devices in the network is essential to assure privacy and data confidentiality [11, 20–26]. The authentication process is vital as it helps protect the IoT sensor networks from malicious nodes that impersonate themselves to alter information and halt the entire network. Authors in [27] proposed an IoT device authentication scheme using a multifactor authentication scheme through device capability and digital signatures. The scheme has little overhead costs and successfully authenticates devices and mitigates against replay and man-in-the-middle attacks. The setback with the proposed scheme is that authentication keys are stored in local storage, making them vulnerable to attacks. Authors in [28] proposed a lightweight mutual authentication mechanism for IoT devices and dynamic session key scheme using a block cipher algorithm. The mechanism has a little overhead cost and has less risk of security attacks. Authors in [29] proposed a lightweight gait-based authentication scheme for IoT devices using a machine learning classifier. The advantage of this mechanism is that it successfully authenticates devices. The author in [30] proposed a mechanism that provides end-to-end IoT device authentication. The mechanisms integrate physical security into traditional asymmetric cryptography-based authentication schemes. The advantage of this method is that it does not impose hardware overheads. Authors in [31] proposed a multi key-based mutual authentication mechanism using secure vaults between IoT devices and the server. A summary of works on the authorization of IoT devices is presented in Table 6.1.

6.2.2 Hardware Security

IoT relies on the perception layer, which carries all the hardware devices that initiate communication and form part of the communication infrastructure. However, more work still needs to be surveyed to and implement measures to mitigate hardware vulnerabilities for IoT devices. It is, therefore, critical to ensure the security of hardware devices in IoT systems [32]. Authors have proposed several works to mitigate hardware vulnerabilities to IoT hardware [33–37]. Authors in [33] proposed a hardware device model for IoT endpoint security by detecting DoS attack through observation of electrical signals in the circuit. The solution was introduced as a resolution to difficulty in adding new software to resource-constrained IoT devices. The proposed security method detects abnormal behavior or attacks in the network by observing the physical characteristics of bus communication between the device and the controller. The method is simple with low computational complexity; however, it does not take into consideration the security of contents of data shared by

Table 6.1 Works on addressing mechanism for perception layer IoT devices

Proposed Scheme	Methodology	Comments	Reference
Improved IoT device authentication scheme using device capability and digital signatures	The scheme uses multifactor device authentication using digital signatures and device capability	Keys are stored in local storage making them vulnerable to attacks	[27]
A lightweight mutual authentication and dynamic session key scheme	Mutual authentication and dynamic session key using a block cipher algorithm	Significantly reduces computational costs and less risk of security attacks	[28]
Lightweight gait-based authentication technique for IoT using subconscious level activities	User authentication is achieved using machine learning classifiers	Successfully authenticates users using machine learning classifiers	[29]
A collaborative PHY-aided technique for end-to-end IoT device authentication	Integrate physical security into traditional asymmetric cryptography-based authentication schemes	The method does not impose hardware overheads	[30]
Authentication for IoT devices and server using secure vaults	A multi-key based mutual authentication mechanism		[30]

IoT devices in the network. Therefore, other attack detection methods based on physical characteristics should be considered. Authors in [34] proposed a low-cost mechanism for detecting the DoS attack by observing electrical signals in the circuit. Though the mechanism detects DoS attacks, other physical characteristics need to be considered other than voltage and factor into account the issue of cost when implemented in real life. Authors in [35, 36] surveyed hardware-assisted techniques for assuring device security in IoT. The techniques highlighted include machine learning, which requires a large data set for training with high overhead costs. Authors in [37] suggest the use of future generation microcontrollers or System on Chip (SoC), a component of the IoT edge node to solve hardware intrinsic security issues at the IoT perception layer.

6.2.3 Lightweight Encryption

Assuring end-to-end secure communication among devices is a major challenge in IoT sensor networks. Data encryption has been proven to be the most effective method for secure communication across wireless sensor network communication where communication channels are more prone to attacks and data breaches. According to research, using efficient encryption methods with low computational costs can play an essential role in reducing security risks and attacks in wireless IoT sensor communication [38, 39]. However, an encryption algorithm must be appropriately chosen to cater for the resource-constrained nature of IoT nodes. The critical issues in designing secure mechanisms or algorithms are to deal with the

trade-off between security, performance, and cost. Several cryptographic methods have been proposed for providing lightweight encryption to resource-constrained IoT communication devices. Encryption techniques are applied to different layers of the IoT architecture and protocols, and every complete system has several levels of encrypt, decrypt, and re-encrypt cycle. These cycles make the system vulnerable to a variety of attacks; therefore, end-to-end security is essential to prevent these attacks. Public key encryption is largely used for authentication, non-repudiation, and key exchange. The mostly used public-key encryption is Rivest, Shamir, and Adleman (RSA) algorithm is considered superlative as compared to symmetric key Advanced Encryption Standard (AES). Applications requiring data authenticity and integrity may be employed AES security mechanism through cipher block chaining (CBC) mode, which produces Message Integrity Code (MIC) or Message Authentication Code (MAC) appended to the transmitted data [2]. RC5 is a lightweight security mechanism for hardware and software implementation promoted by many researchers for protecting user's data. RC5 has been proven to be the right choice for devices with limited memory and power consumption, e.g., Vehicular ad hoc Networks (VANETs) [40]. Researchers have recommended cryptographic primitives such as ECC for IoT authentication. Although ECC requires less memory and computational power, it consumes more power and memory as compared to shared key encryption and is prone to side-channel attacks. ECC is more suitable for mobile Apps over RSA because of its ability to generate faster and smaller keys with less processing power [3]. Diffie–Hellman (DH) algorithm exchanges keys to allow the construction of a common secret key over an unreliable communication channel using discrete logarithmic problems [39]. The advantage with DH key agreement is that logarithmic operations are difficult to crack providing strong authentication to IoT end devices. Table 6.2 presents IoT cryptographic algorithms, their strengths and weaknesses and application areas.

6.2.4 Protecting Sensor Data

Experts forecast a total number of 20 billion connected things, people, processes, and data by the year 2020. This increase will pose a lot of vulnerabilities and security threats, calling for attention to various techniques for managing and protecting sensor data. Watermarking is one of the proposed lightweight security mechanisms with less computational complexity in terms of power consumption, unlike traditional security techniques for wireless sensor networks. The watermarking technique separates data elements into variable sizes in groups, and a secret watermark is generated from consecutive groups. Authors in [43] proposed a verification scheme for protecting data items based on watermarks. From the survey, conclusions were drawn that watermarking helps verify the individual data items accurately hence improving on data integrity, and attackers could not easily detect the watermark. Other methods for protecting sensor data include device fingerprinting

Table 6.2 IoT cryptographic algorithms

Cryptographic algorithm	Description	Purpose	Strength	Weakness	Application	References
AES-Advanced Encryption Algorithm	Symmetric encryption block cipher using substitution and Permutation	Confidentiality	Fast and easy to implement for both software and hardware Good security	Secret keys can be easily cracked	Internet Banking Wireless Communication	[41]
RC5	Symmetric block cipher for hardware and software implementation	Integrity	Simple, fast, less memory, low energy consumption	Limited physical security	Wireless Body Area Networks	[40]
RSA-Rivest Shamir Adelman	Asymmetric algorithm for secure data transmission using two prime numbers	Digital signatures	Difficult to crack	Slow algorithm Has many flaws in its design Least secure	Google Suite Emails HTTPS	[3]
ECC-Elliptic Curve Cryptography	Asymmetric Public key cryptography based on the algebraic structure of curves	Digital signatures Key agreement	Short key length Faster and low computation Excellent Security	Complex encoding algorithm	Crypto-currency-Bitcoin	[42]
DH- Diffie–Hellman	Asymmetric key agreement algorithm based on logarithmic operations	Key agreement	Logarithms are difficult to crack	High computation costs Lack of authentication	IPsec TLS SSL	[25]
SHA1-SHA256–Hash Algorithms	Asymmetric key algorithm	Integrity	Reliable, flexible method	Uses a lot of memory, Collision	IPsec TLS SSL	[17]

and behavioral profiling to recognize and track individuals through sensor data even without names attached to the sensor data being transmitted [11, 44].

6.3 Network layer

This section presents security mechanisms in the network layer.

6.3.1 Encryption Mechanism

The process of encrypting communication in an IoT sensor network is computationally complex regarding cost, energy, and memory for small-sized and resource-constrained IoT devices. Complex security algorithms affect the performance of devices; therefore, algorithms targeted at security and optimized resource utilization at low computation cost [38, 39]. The findings from the HP survey revealed that no encryption standard governs the transmission of data among IoT devices in the local network and the Internet. HP added that despite the existence of transport encryption systems such as SSL and TLS, cloud connections where data is stored and manipulated remain vulnerable to attacks. It is, therefore, imperative to put in place adequately configured transmission encryption standards for IoT sensor networks communication.

6.3.2 Secure Communication

Security in the Internet of Things is a fundamental issue; however, achieving the goal of a secure IoT remains a significant challenge. Internet-based applications and services are susceptible to malicious attacks. Therefore, there is a need to ensure that the security of the network is un-compromised for effective communication [45]. Nonetheless, certain conditions or requirements must be met to protect the network from attacks. Availability, confidentiality, integrity, and authentication are security requirements that protect against threats during communication in an IoT sensor network. Achieving the goal of secure routing in IoT communication presents a significant challenge owing to the requirement of maintaining the uniformity of packets routed from the source to the destination. There is a need to assure a secure end-to-end communication during routing to meet IoT secure communication needs and make significant research contributions [17, 46–48]. This involves protecting the network against threats and attacks during communication. Several security conditions must be met, such as availability, confidentiality, integrity, authenticity, and non-repudiation [49, 50]. The secure routing mechanisms can defend against various security attacks such as sinkhole, Hello flooding, wormhole, selective forwarding, and Sybil (clone ID) attacks [51–53]. Authors [50, 54–56]

Table 6.3 Routing attacks in IoT

Attack	Attack area	Impact on the network	Solutions
Sinkhole	Confidentiality and Integrity	A compromised node gets the attention of the traffic passing through the network then launches an attack which prevents the sink node from receiving the sensed information	SRPL [17]
Hello Flooding	Availability	Broadcast messages into the network and nodes believe that they are neighboring routes then update their routing tables and then the attacker gains access to the network. Puts network in a state of confusion and may permanently cause the network to be dysfunctional	RPL [56]
Wormhole	Confidentiality and Integrity	Route discovery is made difficult for nodes, and routing information cannot be distributed among nodes; therefore, hindering communication between nodes	TARF [50] MRPL [55]
Selective Forwarding	Confidentiality and Integrity	Disruption of route path and the network performance goes down	RPL [56]
Sybil and Clone ID	Confidentiality and Integrity	Compromise route path and the network performance goes down	TRSF [54]

proposed secure routing mechanisms for IoT to address these security threats. A summary of routing attacks, their impacts on the network, and proposed solutions are presented in Table 6.3. Nonetheless, developing a secure routing protocol is a challenge that needs attention for implementation in IoT secure communication.

6.3.3 Trust Recommendation

Trust needs to be established between neighboring nodes in the network. However, an attacker can join the network, masquerade, and recommend itself to other nodes in the network and attract traffic, which then makes it easy for them to forward attacks in the network. Therefore, a secure trust management system needs to be deployed to maintain a high level of trust between nodes in the network. The accuracy of trust between nodes in a network is, therefore, dependent on the number of received recommendations from nodes with a high trust value. There are different methods to ascertain trust between nodes through various recommender systems such as user-based collaborative method, content-based, and location-based filtering approach. The user-based approach refers to the people-to-people relationship where recommendations are made based on feedback ratings from users with the same preference taste with the target user. Content-based users perform matching and find perfectly matching users based on comparisons with previously rated items. Finally, the location-based filtering uses the geographic location of the users to recommend target users within their proximity [47, 57]. Authors in [54] proposed a trust-based routing protocol that utilizes seeming's theory to accomplish a lightweight, QoS,

and optimized routing framework to prevent attacks. The trust schemes are proposed based on the results of the analysis. TRSF may increase network performance if there is an assurance of security in large networks. However, the scheme is not energy efficient and does not support fault tolerance. Authors in [50] proposed a trust-aware routing framework (TARF) that implements secure multi-hop routing by misdirecting traffic routed to identify the trustworthiness of neighboring nodes. From the results of the survey, the framework was found to be effective in defending against sinkhole and wormhole attacks.

6.4 Cloud Layer

6.4.1 Secure cloud computing

Cloud computing has recently become an area of interest in recent years due to infinite and dynamic resources hosted on cloud services, e.g., Dropbox, iCloud, etc. However, security remains a prevalent issue for devices connected to the cloud and data hosted in the cloud. HP surveyed several wireless IoT devices, and the findings revealed that their cloud-based web interfaces and mobile interfaces were not secure enough as they portrayed weak passwords be guessed or cracked easily. The interfaces never required passwords of enough length and strength to log on to their applications. From the survey, it was discovered that some devices used the same passwords, which raised security questions as this made the systems vulnerable to remote access by attackers guessing the repetitive passwords used across. These systems were prone to attacks because they did no't have a limit for failed attempts of trying to log on to the system after several attempts of access. Several attempts have been made by authors to authenticate devices and servers connecting to the cloud through mechanisms such as elliptic curve cryptography [58], use of virtual based passwords using linear functions [11], and cookie-based password management systems [59]. Elliptic curve cryptographic scheme has been proven to be the best in the systematic review of literature by [59] as it requires small key sizes. Elliptic curve cryptographic scheme can be embedded in any device that is HTTP enabled and performs efficient computations in defending against threats and attacks in the IoT cloud.

6.4.2 Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) is a cryptographic primitive that allows different parties to perform a joint function or computation without disclosing their private inputs and outputs to each other. MPC uses the concept of Threshold cryptography whereby private keys are shared among parties, and the cryptographic operation of those parties can only be carried out when an authorized quorum of

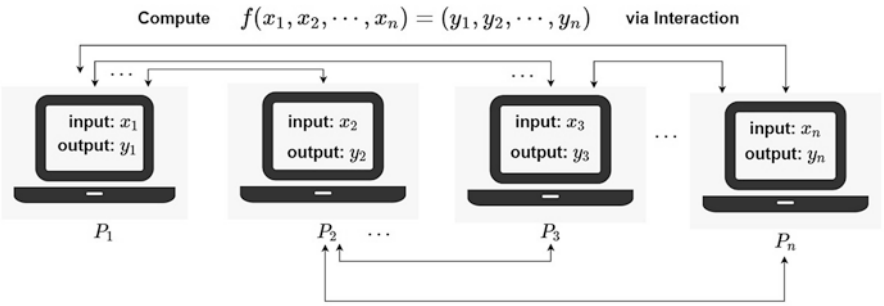


Fig. 6.1 Secure multi-party computation

those parties agrees to carry it out. SMPC addresses the problem of cooperative computation performed on a private data set by several participants in a secure manner through a distributed computing method [60]. MPC protocols exist which assure mathematical proof of security guaranteeing that despite security preaches, a malicious code will not be able to manipulate, carry out unauthorized operations, or learn anything about the secret key. This method has been proven to be efficient in encrypting data and assuring data privacy [60]. However, this method is computationally complex and may slow down the runtime of a system. Figure 6.1 shows how secure multi-party computation works.

6.5 User Interface Layer

6.5.1 Information Privacy

The user interface layer comprises of everything a user may interact with when navigating an application or a website through various controls. Failure to protect a user interface leads to multiple cybercrime activities such as eavesdropping and man-in-the-middle attack. Therefore, it is essential to invest in security solutions that include user interface protection such as API security. Web API security is concerned with protecting data that is sent over the Internet. However, several protocols exist, such as Representational State Transfer (REST), which uses HTTP or Transport layer security (TLS) to keep Internet connectivity private. REST is used within social network applications, and its responsibility is to remove ambiguity in a network by using HTTP to get, post, put, and delete methods [61]. Simple object access protocol (SOAP) uses built-in protocols known as web service security to assure confidentiality and authentication during Internet connectivity. It uses various forms of security such as XML encryption, XML signatures, and tokens to authenticate and validate communicating entities. Other forms of strengthening API security involve using an API gateway, the use of sniffers to identify vulnerabilities,

and the use of Quotas to protect against denial of service attacks and privacy assurance.

6.5.2 Key Agreement

Application layer security supports symmetric, asymmetric, and hashing algorithms. Symmetric algorithms such as DES have high cryptographic strength and high computational power and can assure confidentiality in the user interface layer. The most popular asymmetric algorithm in the user interface or application layer is the RSA, which uses digital certificates to authenticate participating parties. Cryptographic hashing, on the other hand, ensures integrity and authentication using common hash algorithms such as MD5 and SHA. Authors [62] proposed a lightweight key agreement protocol that robustly achieves mutual authentication without exposing the real identity of users in vehicle-to-grid (V2G) networks for smart grids. The key agreement mechanisms perform mutual authentication through hash functions and bitwise exclusive-OR (XOR) operations. The evaluation from the research proves that the proposed protocol can withstand different kinds of attacks in IoT. Authors in [42] performed a comparative survey to analyze the performance of Diffie–Hellman, Elliptic Curve Diffie–Hellman (ECDH), and RSA algorithm. The analysis of algorithms assessed power consumption, performance, and area. The results of the survey proved that the ECDH algorithm performed better than others in terms of power and area. However, lightweight key management mechanisms need to be further investigated to ensure the confidentiality of data to enable trust between connected things in a sensor network.

6.5.3 Data Management

Effective data management in IoT every entity applies policies and standards to manage, protect, and ensure that data is transmitted efficiently. Service-level Agreements (SLA's) require to be identified in every service provided by an IoT system. The enforcement of these policies will enable trust in a sensor network among human users in the IoT sensor networks. Policies mainly involve access to resources in a network. However, there are no universal policies and standards to govern and control the design and implementation of algorithms, making it difficult to control the security of IoT [22]. Standards and policies for securing IoT are not enough, and there is a need for mechanisms to enforce the policies. Authors in [63] proposed an automated mechanism to derive network security policies for devices without requiring vendor cooperation or modifications to devices or their cloud infrastructure. The method is scalable and presents an effective multilayered architecture that minimizes the impact of IoT attacks.

6.6 Future Works

Although the application of IoT has been widely adopted, an extensive research effort still needs to be done in this direction. Open research challenges have been identified, such as limited computational power for resource-constrained IoT devices, hardware vulnerability, un-secure routing, ineffective addressing mechanisms, ineffective sensing, and un-secure routing between IoT devices [64–67]. This section broadens the open research areas that need to be further investigated to address the challenges in IoT sensor networks. We highlight potential future directions and futuristic applications that can help alleviate these challenges.

6.6.1 Artificial Intelligence

By the year 2025, 30 billion things will be connected, meaning that big data will only get bigger hence the need to make sense of the high-velocity data streamed from the IoT devices. Artificial intelligence (AI) provides a framework and tools for radically shaping the technological landscape and analyzing diverse IoT data that is physically impossible with man. AI powers IoT with machines with decision making and the use of algorithms. The combination of A1 with IoT helps turn data into new revenue streams and the seamless integration of intelligent systems with IoT applications for the creation of an optimized application-specific solution [68–72]. Machine learning and Robotics complement IoT in designing and developing autonomous programmable systems. Machine learning is an advancement of AI using pattern recognition, neural networks, and swarm intelligence technologies to perform independent tasks. The advantages of the fusion between the two technologies lead to increased efficiency, new business opportunities, and predictive and preventative measures against anticipated risks.

6.6.2 Blockchain for IoT Security

Blockchain technology is a solution for IoT connecting billions of connected devices and applications, which involves transactions and interactions. This technology is widely adopted in the Finance sector. Blockchain is the underlying technology behind Bitcoin to manage, control, and secure IoT devices. Blockchain allows for autonomous transactions of data or money between two devices without the third party to certify the transaction. This technology provides a secure, reliable, time-stamped contractual handshake between autonomous devices. Smart contracts can be issued among devices to establish secure message exchange by creating agreements that are only executed upon meeting specific requirements. Smart contracts allow for greater automation, scalability, low costs, and security [7, 73].

Blockchain technology can further be used for analyzing big data produced from the interconnection of IoT devices and increasing the security of the infrastructure within organizations [37, 74–77]. Blockchain platforms: IOTA was explicitly designed for IoT transaction settlement at the data transfer layer and Tangle platform which is a blockless, secure, peer-to-peer network where users verify transactions of other users. Volkswagen uses IOTA for reporting critical factors such as mileage in their report cards stored in a distributed ledger. VeChain is another blockchain platform used in medical and healthcare applications for tracking production processes for medical devices, also allowing sharing of real-time biometric information between patients and doctors for real-time monitoring.

6.6.3 Machine Learning for Data Security

Machine learning algorithms and techniques have gained popularity in recent research in the field of artificial intelligence to enhance IoT security. Machine learning involves the use of algorithms within a program to learn from collected data. Machine learning provides methods for implementing necessary security measures such as authentication, intrusion detection, and offloading and access control [71, 76, 78, 79]. Authentication techniques include comparing radio channel characteristics against characteristics identified by the transmitters at the devices' physical layer [80]. Machine learning mechanisms used for authentication include Q-learning and artificial neural networks [81].

6.6.4 Context-Aware Sensing

The second crucial future direction for the IoT would be enhancing context-aware sensing for IoT, IoP, and IoE. Context awareness defines the ability to detect and respond to situations and environments in which a computational artifact is embedded to gather and utilize information to positively affect services relevant to a person or device [82]. The need to optimize the network lifetime and energy-efficient sensing for resource-constrained devices and sensors remains a challenge in IoT communication. Authors in [78, 83] strongly recommended for the development of intelligent sensing applications that can adapt to changing environments and use context information to increase the energy efficiency of the sensor node processing and communication [84]. The author in [85] recommended opportunistic and context-aware affect sensing using facial expressions and voice to make mood inferences on the smartphone to provide information to various applications such as mental health management. Audio and video sensing for resource-constrained smartphone sensors are made feasible by the advent of low-power digital signal processing co-processor (DSP) and graphics processing unit (GPU). The RFID technology is an emerging technology for embedding sensing capabilities to everyday

objects for identification and tracking in supply chain management using RFID tags. Although context awareness is the key ingredient to create smart applications that support the users' day-to-day activities, the ability to achieve non-intrusive communication in IoT remains a challenge in mobile converged platforms and heterogeneous environments. While it is crucial to reduce interactions between people and devices, it is of utmost importance to provide users with control over the things they interact to assure privacy, security, and personalization [67, 83, 85]. In the future, more context-aware applications will be targeted towards humans (IoP) as users of the services.

6.6.5 Cloud Infrastructure

The third crucial future direction for the IoT would be to enhance storage, processing, and retrieval of information challenges in IoT. A potential solution to these problems is the IoT cloud infrastructure, which encompasses hardware and software components that support computing requirements for cloud computing. The cloud infrastructure hardware includes servers, storage, and network and virtualization software such as VMware. IoT produces large amounts of data by different devices; as a result, harvesting of sensor data in an efficient manner, and the processing of that information remains a challenge. To address these challenges, the use of intelligent information retrieval methods and machine learning techniques would serve as useful tools for data processing and analytics [33, 78, 86]. A critical challenge for IoT is the implementation of artificial intelligence, which has high computation complexity on the resource-constrained devices. Authors revealed that the implementation of machine learning algorithms on IoT devices is feasible and practical to profile performance in terms of speed, accuracy, and power consumption [38, 82, 87]. Another future work direction on enhancing the cloud involves the exploration of new machine learning methods such as deep learning and neural networks [88]. Security and privacy of data stored on the cloud is another critical concern that needs to be addressed [89].

6.6.6 Sensor Internet of People

Experts forecast a total number of 20 billion connected things, people, processes, and data by the year 2020. The goal of IoT is to develop applications that can be integrated into everyday lives. Smartphones are small yet powerful devices with the capability to support IoT evolution towards the Internet of People (IoP) [78, 90]. The smartphone must have capabilities to improve the connection between people and the Internet by supporting context-aware sensing. Context sensing means giving the smartphone the ability to learn about its owner by constructing a digital profile. The smartphone needs to transparently negotiate and propose interactions

with other devices on the Internet. The smartphone must be able to manage and update digital profiles and provide its owner's context as a service to other people and scan for services that might be of interest to its owner. In the IoP, people can be used for data collection or sensing and contributes towards collaborative and intelligent information processing [91]. Mobile-centric models such as the People as a Service allows smartphones to securely infer and share users' social profiles as a third party directly from a user's smartphone. Other applications supported by this framework are GPS, social status, health habits. However, users must be empowered to personalize their preferences and have the liberty to choose whom to interact with and whom to share information with for purposes of privacy and security. People participating in IoT must be empowered to identify, verify tag, and report unexpected behavior during communication in IoT [92–95]. An example of an IoP scenario has a smartphone that learns about one's surrounding environment, which learns where one lives, works, and the route which one takes to work and at what time. A smartphone can provide routine reports on the transport control system by giving alerts on the shortest paths, where traffic jams are possible incidents to promote smart transportation. Another application of IoP includes using a smartphone to manage applications for smart homes, for example, controlling the electric heater, air conditioning, or lights remotely. Also, IoP could be used in financial institutions to support mobile banking services for transacting and managing accounts at one's comfort and convenience. IoP eliminates the middleman during transactions giving users complete control over their data and information and enables the device to devise connection without the need of a central server. IoP uses a hash algorithm-SHA256, the same underlying security technology applied to Bitcoin [65, 76, 96, 97]. Devices with low computational power such as personal computers, Raspberry Pi, or smartphones can deploy IoP.

6.6.7 Internet of Everything

The Internet of Everything (IoE) is a network of networks with over a billion connections of things, people, processes, and data, which turns information into exceptional economic opportunities, richer experiences as well as risks for individuals, businesses, and countries [98, 99]. IoE is driven by advances in technology which connect IoT and IoP orchestrated by an increase in IP-enabled devices and global Internet availability. IoE results in the increased number of connected things that will intelligently and autonomously combine data into more useful high-level information for further evaluation, intelligent decision making, and effective control over the environment. Big data analytics will enable predictive modeling, supporting the availability of near-real-time data [84, 98]. Today, people connect to the Internet through several devices ranging from smartphones to tablets and personal computers. The government and citizens of Finland are already enjoying the benefits of IoE through the local waste management company, which added sensors to garbage cans that signal when they are full prompting for a pickup. Another IoE application

is a central building management network in South Korea that monitors and optimizes energy use reducing energy consumption by 30%. In the future people will become nodes on the Internet [100]. For example, people will be able to swallow medication that will sense and give a comprehensive report of the patient's health to a doctor over a secure network. Another example involves placing a sensor on a person's skin to gather information about personal medical history. Cities experiencing budget constraints could use IoE for generating revenues through applications such as smart parking, gas monitoring as well as smart meters for water and electricity management. IoE will assist in improving public infrastructure by providing data on citizen behaviors, transport, and logistics, in addition to also providing real-time data on weather conditions, events, and emergency responses as presented in [101–105]. Social security departments use the application of predictive modeling for fighting crime, early warning systems, smart grids, and infrastructure protection. Therefore, it is vital to investigate methods that can leverage crowdsourcing and big data in IoE to improve communication between machine-to-machine (M2M), people-to-people (P2P), and machine-to-people (M2P) while reducing costs and producing a positive environmental impact.

6.7 Summary

In this chapter, we discussed various security mechanisms for addressing security challenges in the perception, network, cloud, and user interface layer of the IoT sensor network. Security must be implemented at each layer of the IoT architecture for safe, effective, and un-compromised communication. IoT devices are resource-constrained with limited memory and energy efficiency; subsequently, there is a need to assure that the security mechanism deployed is lightweight, and resource utilization is optimized at low computation cost. These involve the optimal use of memory-constrained IoT devices and energy as they directly affect the performance of the IoT devices or the sensor network. However, the security mechanism must be appropriately chosen to cater to the resource-constrained nature of IoT nodes and the nature of the application. Security is implemented in the perception layer protecting hardware or physical devices and sensor data through mechanisms such as lightweight cryptographic algorithms such as AES, DES, ECC. Security in the network layer is implemented through identity authentication, secure communication, and trust recommendation between communicating entities. In the cloud layer, we discussed secure cloud computing and secure multi-party computation as security mechanisms for addressing cloud layer security challenges. Lastly, we discuss information privacy, key agreement, data management, and security management as mechanisms to address user interface layer security challenges. Finally, future work directions and perspectives are drawn and discussed, highlighting some futuristic applications that can help alleviate security challenges in wireless IoT sensor networks. These applications include the use of artificial intelligence, blockchain, machine-learning, context-aware sensing, cloud infrastructure, sensor Internet of

People and Internet of Everything. According to research, using efficient encryption methods or security mechanisms with low computational costs can play an essential role in reducing security risks and attacks in IoT sensor network communication. Therefore, careful consideration of energy efficiency and memory management has to be made when designing a security mechanism for IoT sensor networks.

References

1. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4) (2015). https://doi.org/10.1109/COMST.2015.2444095
2. 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. Tutorials **17**(3), 1294–1312 (2015). https://doi.org/10.1109/COMST.2015.2388550
3. S. Nisha and M. Farik, RSA public key cryptography algorithm-a review. Int. J. Sci. Technol. Res. **6**(7) (2017) [Online]. Available: www.ijstr.org.
4. M.A. Ferrag, L.A. Maglaras, H. Janicke, J. Jiang, L. Shu, Authentication protocols for Internet of things: A comprehensive survey. Secur. Commun. Networks **2017** (2017). https://doi.org/10.1155/2017/6562953
5. S. Madakam, R. Ramaswamy, S. Tripathi, Jcc_2015052516013923. J. Comput. Commun. **2015**, 164–173 (2015). https://doi.org/10.4236/jcc.2015.35021
6. I. Ali, S. Sabir, Z. Ullah, Internet of things security, device authentication and access control: A review. **14**(8), 456–466 (2019) [Online]. Available: http://arxiv.org/abs/1901.07309.
7. A. Tewari, B.B. Gupta, Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. Futur. Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.04.027
8. J. Deogirikar, A. Vidhate, Security attacks in IoT: A survey. Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC **2017**, 32–37 (2017). https://doi.org/10.1109/I-SMAC.2017.8058363
9. A.A.A. Ari et al., Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges. Appl. Comput. Informatics (2019). https://doi.org/10.1016/j.aci.2019.11.005
10. M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, R. Brooks, The sleep deprivation attack in sensor networks: Analysis and methods of defense. Int. J. Distrib. Sens. Networks **2**(3), 267–287 (2006). https://doi.org/10.1080/15501320600642718
11. V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access **7**, 82721–82743 (2019). https://doi.org/10.1109/ACCESS.2019.2924045
12. B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in Internet of Things. J. Netw. Comput. Appl. **84**, 25–37 (2017). https://doi.org/10.1016/j.jnca.2017.02.009
13. K. Chen et al., Internet-of-Things security and vulnerabilities: Taxonomy, challenges, and practice. J. Hardw. Syst. Secur. **2**(2), 97–110 (2018). https://doi.org/10.1007/s41635-017-0029-7
14. X. Li, J. Xu, H.N. Dai, Q. Zhao, C.F. Cheang, Q. Wang, On modeling eavesdropping attacks in wireless networks. J. Comput. Sci. **11**, 196–204 (2015). https://doi.org/10.1016/j.jocs.2014.10.006
15. D. Sopori, T. Pawar, M. Patil, R. Ravindran, Internet of Things: Security threats. Ijarcet.Org 6(3), 263–267 (2017) [Online]. Available: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-6-ISSUE-3-263-267.pdf.
16. K. Somasundaram, K. Selvam, IOT – attacks and challenges. Int. J. Eng. Tech. Res. **8**(9), 9–12 (2018). https://doi.org/10.31873/ijetr.8.9.67

17. G. Glissa, A. Rachedi, A. Meddeb, A secure routing protocol based on RPL for internet of things, in *2016 IEEE Glob. Commun. Conf. GLOBECOM 2016 - Proc.*, (2016), pp. 1–7. https://doi.org/10.1109/GLOCOM.2016.7841543
18. M. Bouabdellah, N. Kaabouch, F. El Bouanani, H. Ben-Azza, Network layer attacks and countermeasures in cognitive radio networks: A survey. J. Inf. Secur. Appl. **38**, 40–49 (2018). https://doi.org/10.1016/j.jisa.2017.11.010
19. I. Stojmenovic, S. Wen, The fog computing paradigm: Scenarios and security issues. Proc. 2014 Fed. Conf. Comput. Sci. Inf. Syst. **2**, 1–8 (2014). https://doi.org/10.15439/2014f503
20. S. Subashini, V. Kavitha, A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl. **34**(1), 1–11 (2011). https://doi.org/10.1016/j.jnca.2010.07.006
21. S. Al Jadaani, M. Al Maliki, W. Al Ghamdi, M. Hemalatha, Security issues in cloud computing. Int. J. Appl. Eng. Res. **11**(12), 7669–7671 (2016). https://doi.org/10.5120/6369-8736
22. R. Mahmoud, T. Yousuf, F. Aloul, I. Zualkernan, Internet of Things (IoT) Security: Current status, challenges and prospective measures, in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, (IEEE, London, 2015)
23. V. Beltran, A.F. Skarmeta, An overview on delegated authorization for CoAP: Authentication and authorization for Constrained Environments (ACE), in *2016 IEEE 3rd World Forum Internet Things, WF-IoT 2016*, (2017), pp. 706–710. https://doi.org/10.1109/WF-IoT.2016.7845482
24. P. Nespoli, M. Zago, A.H. Celdran, M.G. Perez, F.G. Marmol, F.J. Garcia Clernente, A dynamic continuous authentication framework in IoT-enabled environments, in *2018 5th Int. Conf. Internet Things Syst. Manag. Secur. IoTSMS 2018*, (2018), pp. 131–138. https://doi.org/10.1109/IoTSMS.2018.8554389
25. M. El-Hajj, M. Chamoun, A. Fadlallah, A. Serhrouchni, Analysis of authentication techniques in Internet of Things (IoT), in *2017 1st Cyber Secur. Netw. Conf. CSNet 2017*, vol. 2017, (2017), pp. 1–3. https://doi.org/10.1109/CSNET.2017.8242006
26. N. Tapas, G. Merlino, F. Longo, Blockchain-Based IoT-cloud authorization and delegation, in *Proc. – 2018 IEEE Int. Conf. Smart Comput. SMARTCOMP 2018*, (2018), pp. 411–416. https://doi.org/10.1109/SMARTCOMP.2018.00038
27. Z.A. Alizai, N.F. Tareen, I. Jadoon, Improved IoT device authentication scheme using device capability and digital signatures, in *ICAEM 2018 – 2018 Int. Conf. Appl. Eng. Math. Proc.*, (2018), pp. 115–119. https://doi.org/10.1109/ICAEM.2018.8536261
28. J.H. Han, J.N. Kim, A lightweight authentication mechanism between IoT devices, in *Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Technol. Lead. Fourth Ind. Revolution, ICTC 2017*, vol. 2017, (2017), pp. 1153–1155. https://doi.org/10.1109/ICTC.2017.8190883
29. P. Musale, D. Baek, B.J. Choi, Lightweight gait based authentication technique for IoT using subconscious level activities, in *IEEE World Forum Internet Things, WF-IoT 2018 – Proc.*, vol. 2018, (2018), pp. 564–567. https://doi.org/10.1109/WF-IoT.2018.8355210
30. P. Hao, X. Wang, W. Shen, A collaborative PHY-aided technique for end-to-end IoT device authentication. IEEE Access **6**, 42279–42293 (2018). https://doi.org/10.1109/ACCESS.2018.2859781
31. T. Shah, S. Venkatesan, Authentication of IoT device and IoT server using secure vaults, in *Proc. – 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018*, (2018), pp. 819–824. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00117
32. N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: A survey. IEEE Internet Things J. **5**(1), 450–465 (2018). https://doi.org/10.1109/JIOT.2017.2750180
33. Y. Lu, Industry 4.0: A survey on technologies, applications and open research issues. J. Ind. Inf. Integr. **6**, 1–10 (2017). https://doi.org/10.1016/j.jii.2017.04.005
34. R. Jinnai, A. Inomata, I. Arai, K. Fujikawa, Proposal of hardware device model for IoT endpoint security and its implementation, in *2017 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2017*, (2017), pp. 91–93. https://doi.org/10.1109/PERCOMW.2017.7917533

35. F. Rahman, M. Farmani, M. Tehranipoor, Y. Jin, Hardware-assisted cybersecurity for IoT devices, in *2017 18th Int. Work. Microprocess. SOC Test Verif.*, (2017), pp. 51–56. https://doi.org/10.1109/MTV.2017.16
36. F. Bruguier, P. Benoit, L. Torres, L. Bossuet, Hardware security: From concept to application, in *2016 11th Eur. Work. Microelectron. Educ. EWME 2016*, (2016), pp. 1–6. https://doi.org/10.1109/EWME.2016.7496483
37. K. Sudeendra Kumar, S. Sahoo, A. Mahapatra, A.K. Swain, K.K. Mahapatra, Security enhancements to system on chip devices for IoT perception layer, in *Proc. – 2017 IEEE Int. Symp. Nanoelectron. Inf. Syst. iNIS 2017*, vol. 2018, (2018), pp. 151–156. https://doi.org/10.1109/iNIS.2017.39
38. L. Xiao, X. Wan, X. Lu, Y. Zhang, D. Wu, IoT security techniques based on machine learning: How do iot devices use AI to enhance security? IEEE Signal Process. Mag. **35**(5), 41–49 (2018). https://doi.org/10.1109/MSP.2018.2825478
39. R. Tripathi, S. Agrawal, Comparative study of symmetric and asymmetric cryptography. Int. J. Adv. Found. Res. Comput. 1 (6), 68–76 (2014) [Online]. Available: https://pdfs.semanticscholar.org/e0e4/810c5276f9c05cc82425fcf911f206c52bef.pdf.
40. N. Alsaffar, W. Elmedany, H. Ali, Application of RC5 for IoT devices in smart transportation system, in *2019 8th Int. Conf. Model. Simul. Appl. Optim. ICMSAO 2019*, (2019), pp. 1–4. https://doi.org/10.1109/ICMSAO.2019.8880351
41. D. Mendez, I. Papapanagiotou, B. Yang, Internet of Things: Survey on Security and Privacy. pp. 1–16, 2020.
42. T.K. Goyal, V. Sahula, Lightweight security algorithm for low power IoT devices, in *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, (2016), pp. 1725–1729. https://doi.org/10.1109/ICACCI.2016.7732296
43. Y. Xiao, G. Gao, An independent individual certification scheme based on digital watermark in WSNs, in *Proc. – 2019 IEEE Int. Conf. Smart Internet Things, SmartIoT 2019*, (2019), pp. 474–478. https://doi.org/10.1109/SmartIoT.2019.00086
44. I. Ud Din et al., The Internet of Things: A review of enabled technologies and future challenges. IEEE Access **7**, 7606–7640 (2019). https://doi.org/10.1109/ACCESS.2018.2886601
45. R. van der Meulen, Gartner says 8.4 billion connected 'Things' will be in use in 2017, up 31 percent from 2016. Gartner (2017) https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016. Accessed 20 July 2019
46. J. Karlsson, L.S. Dooley, G. Pulkkis, Secure routing for MANET connected Internet of things systems, in *Proc. – 2018 IEEE 6th Int. Conf. Futur. Internet Things Cloud, FiCloud 2018*, (2018), pp. 114–119. https://doi.org/10.1109/FiCloud.2018.00024
47. A.E. Basabi, J. He, S.M. Hashemi, Secure routing in IoT with multi-objective simulated annealing, in *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 – Proc.*, (2017), pp. 2073–2076. https://doi.org/10.1109/CompComm.2016.7925065
48. V. Arun, D.L. Reddy, S. Srinivas, Encryption standards for security system in energy harvesting for IoT requirements – Review, in *Proc. Int. Conf. Intell. Sustain. Syst. ICISS 2017, no. Iciss*, (2018), pp. 1224–1227. https://doi.org/10.1109/ISS1.2017.8389380
49. K. Zhang, X. Liang, R. Lu, X. Shen, Sybil {Attacks} and {Their} {Defenses} in the {Internet} of {Things}. IEEE Internet Things J. **1**(5), 372–383 (2014). https://doi.org/10.1109/JIOT.2014.2344013
50. G. Zhan, W. Shi, J. Deng, Design and implementation of TARF: A trust-aware routing framework for WSNs. IEEE Trans. Dependable Secur. Comput. **9**(2), 184–197 (2012). https://doi.org/10.1109/TDSC.2011.58
51. A. Karaagac, J. Haxhibeqiri, I. Moerman, J. Hoebeke, Time-critical communication in 6TiSCH networks, in *2018 IEEE Wirel. Commun. Netw. Conf. Work. WCNCW 2018*, (2018), pp. 161–166. https://doi.org/10.1109/WCNCW.2018.8368987

52. P.P. Lokulwar, H.R. Deshmukh, Threat analysis and attacks modelling in routing towards IoT, in *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, (2017), pp. 721–726. https://doi.org/10.1109/I-SMAC.2017.8058273
53. A. Walid, A. Mostafa, M. Salama, MalNoD: Malicous node discovery in internet-of-things through fingerprints, in *Proc. – 2017 Eur. Conf. Electr. Eng. Comput. Sci. EECS 2017*, (2018), pp. 280–285. https://doi.org/10.1109/EECS.2017.58
54. J. Duan, D. Yang, H. Zhu, S. Zhang, J. Zhao, TSRF: A trust-aware secure routing framework in wireless sensor networks. Int. J. Distrib. Sens. Networks **2014** (2014). https://doi.org/10.1155/2014/209436
55. M.A. Lodhi, A. Rehman, Multiple path RPL for low power lossy networks, in *2015 IEEE Asia Pacific Conf. Wirel. Mob.*, (2015), pp. 279–284. https://doi.org/10.1109/APWiMob.2015.7374975
56. L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the RPL-based internet of things. Int. J. Distrib. Sens. Networks **2013** (2013). https://doi.org/10.1155/2013/794326
57. V. Mohammadi, A.M. Rahmani, A.M. Darwesh, A. Sahafi, Trust-based recommendation systems in Internet of Things: A systematic literature review. Human-centric Comput. Inf. Sci. **9**(1) (2019). https://doi.org/10.1186/s13673-019-0183-8
58. B. Pourghebleh, N.J. Navimipour, Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research. J. Netw. Comput. Appl. **97**, 23–34 (2017). https://doi.org/10.1016/j.jnca.2017.08.006
59. S. Kalra, S.K. Sood, Secure authentication scheme for IoT and cloud servers. Pervasive Mob. Comput. **24**, 210–223 (2015). https://doi.org/10.1016/j.pmcj.2015.08.001
60. C. Zhao et al., Secure Multi-Party Computation: Theory, practice and applications. Inf. Sci. (Ny). **476**, 357–372 (2019). https://doi.org/10.1016/j.ins.2018.10.024
61. A. Al-fuqaha, S. Member, M. Guizani, M. Mohammadi, S. Member, Internet of Things: A Survey on Enabling. IEEE Commun. Surv. Tutorials **17**(4), 2347–2376 (2015). https://doi.org/10.1109/COMST.2015.2444095
62. J. Shen, T. Zhou, F. Wei, X. Sun, Y. Xiang, Privacy-preserving and lightweight key agreement protocol for V2G in the social internet of things. IEEE Internet Things J. **5**(4), 2526–2536 (2018). https://doi.org/10.1109/JIOT.2017.2775248
63. D. Barrera, I. Molloy, and H. Huang, Standardizing IoT Network Security Policy Enforcement. 2018. doi: https://doi.org/10.14722/diss.2018.23007.
64. 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 *Proc. – IEEE 7th Int. Conf. Serv. Comput. Appl. SOCA 2014*, (2014), pp. 230–234. https://doi.org/10.1109/SOCA.2014.58
65. T. Alam, Blockchain and its role in the Internet of Things (IoT), in *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, (2019), pp. 151–157. https://doi.org/10.32628/cseit195137
66. A. Poniszewska-Maranda, D. Kaczmarek, Selected methods of artificial intelligence for Internet of Things conception, in *Proc. 2015 Fed. Conf. Comput. Sci. Inf. Syst.*, vol. 5, (2015), pp. 1343–1348. https://doi.org/10.15439/2015f161
67. O. Yurur, C.H. Liu, Z. Sheng, V.C.M. Leung, W. Moreno, K.K. Leung, Context-awareness for mobile sensing: A survey and future directions. *IEEE Commun. Surv. Tutorials* **18**(1), 68–93 (2016). https://doi.org/10.1109/COMST.2014.2381246
68. M. Song et al., In-situ AI: Towards autonomous and incremental deep learning for IoT systems, in *Proc. – Int. Symp. High-Performance Comput. Archit.*, vol. 2018, (2018), pp. 92–103. https://doi.org/10.1109/HPCA.2018.00018
69. K.E. Stansfield, F. Azmat, Developing high value IoT solutions using AI enhanced ISO 16355 for QFD integrating market drivers into the design of IoT offerings, in *Proc. 2017 Int. Conf. Commun. Comput. Digit. Syst. C-CODE 2017*, (2017), pp. 412–416. https://doi.org/10.1109/C-CODE.2017.7918967
70. J. Knickerbocker et al., Heterogeneous integration technology demonstrations for future healthcare, IoT, and AI computing solutions, in *Proc. – Electron. Components Technol. Conf.*, vol. 2018, (2018), pp. 1519–1528. https://doi.org/10.1109/ECTC.2018.00231

71. S.B. Calo, M. Touna, D.C. Verma, A. Cullen, Edge computing architecture for applying AI to IoT, in *Proc. – 2017 IEEE Int. Conf. Big Data, Big Data 2017*, vol. 2018, (2018), pp. 3012–3016. https://doi.org/10.1109/BigData.2017.8258272
72. S.S. Gill et al., Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges. Internet of Things (2019). https://doi.org/10.1016/j.iot.2019.100118
73. A. Riahi Sfar, E. Natalizio, Y. Challal, Z. Chtourou, A roadmap for security challenges in the Internet of Things. Digit. Commun. Networks **4**(2), 118–137 (2018). https://doi.org/10.1016/j.dcan.2017.04.003
74. M.G. Samaila, M. Neto, D.A.B. Fernandes, M.M. Freire, P.R.M. Inácio, Challenges of securing Internet of Things devices: A survey. Secur. Priv. **1**(2), e20 (2018). https://doi.org/10.1002/spy2.20
75. A. Assiri, H. Almagwashi, IoT security and privacy issues, in *1st Int. Conf. Comput. Appl. Inf. Secur. ICCAIS 2018*, (2018), pp. 1–5. https://doi.org/10.1109/CAIS.2018.8442002
76. A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: The case study of a smart home, in *2017 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2017*, (2017), pp. 618–623. https://doi.org/10.1109/PERCOMW.2017.7917634
77. B.K. Mohanta, D. Jena, S.S. Panda, S. Sobhanayak, Blockchain technology: A survey on applications and security privacy challenges. *Internet of Things* (2019). https://doi.org/10.1016/j.iot.2019.100107
78. M.R. Palattella et al., Standardized protocol stack for the internet of (important) things. *IEEE Commun. Surv. Tutorials* **15**(3), 1389–1406 (2013). https://doi.org/10.1109/SURV.2012.111412.00158
79. E.M. Karanja, S. Masupe, M.G. Jeffrey, Analysis of internet of things malware using image texture features and machine learning techniques. *Internet of Things* **9**, 100153 (2020). https://doi.org/10.1016/j.iot.2019.100153
80. W. Shang-Ping, M. Qiao-Mei, Z. Ya-Ling, L. You-Sheng, An authentication protocol for RFID tag and its simulation. J. Networks **6**(3), 446–453 (2011). https://doi.org/10.4304/jnw.6.3.446-453
81. F. Ouakasse, S. Rakrak, From RFID tag ID to IPv6 address mapping mechanism, in *Proc. – 2015 3rd Int. Work. RFID Adapt. Wirel. Sens. Networks, RAWSN 2015 – conjunction with Int. Conf. NETworked Syst. NETYS 2015*, (2015), pp. 63–67. https://doi.org/10.1109/RAWSN.2015.7173281
82. J. Venkatesh, C. Chan, A.S. Akyurek, T.S. Rosing, A modular approach to context-aware IoT applications, in *Proc. – 2016 IEEE 1st Int. Conf. Internet-of-Things Des. Implementation, IoTDI 2016*, (2016), pp. 235–240. https://doi.org/10.1109/IoTDI.2015.13
83. H. Boyes, B. Hallaq, J. Cunningham, T. Watson, The industrial internet of things (IIoT): An analysis framework. *Comput. Ind.* **101**, 1–12 (2018). https://doi.org/10.1016/j.compind.2018.04.015
84. C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: A survey. IEEE Commun. Surv. Tutorials **16**(1), 414–454 (2014). https://doi.org/10.1109/SURV.2013.042313.00197
85. D. Mcfarlane, Industrial Internet of Things: Applying IoT in the industrial context. (Online.) Available: https://www.scribd.com/document/432087496/IIOT. Accessed 16 Mar 2020.
86. V. Nagamalla, A. Varanasi, A review of security frameworks for Internet of Things, in *2017 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2017*, (2017). https://doi.org/10.1109/ICICES.2017.8070757
87. H.N. Saha, A. Mandal, A. Sinha, Recent trends in the Internet of Things, in *2017 IEEE 7th Annu. Comput. Commun. Work. Conf. CCWC 2017*, (2017), pp. 1–4. https://doi.org/10.1109/CCWC.2017.7868439
88. M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J. Inf. Secur. Appl. (2020). https://doi.org/10.1016/j.jisa.2019.102419

89. L. Urquhart, D. McAuley, Avoiding the internet of insecure industrial things. Comput. Law Secur. Rev. **34**(3), 450–466 (2018). https://doi.org/10.1016/j.clsr.2017.12.004
90. T.J. Charity, Smart world of Internet of Things (IoT) and it ' s security concerns, in *2016 IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data*, (2016), pp. 240–245. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.64
91. T. Mick, R. Tourani, S. Misra, LASeR: Lightweight Authentication and Secured Routing for NDN IoT in Smart Cities. IEEE Internet Things J. **5**(2), 755–764 (2018). https://doi.org/10.1109/JIOT.2017.2725238
92. Y. Lu, Journal of Industrial Information Integration Industry 4.0: A survey on technologies, applications and open research issues. J. Ind. Inf. Integr. **6**, 1–10 (2017). https://doi.org/10.1016/j.jii.2017.04.005
93. L. Yang, W. Li, S. Member, M. Ghandehari, G. Fortino, S. Member, People-centric cognitive Internet of Things for the quantitative analysis of environmental exposure. IEEE Internet Things J. **5**(4), 2353–2366 (2018). https://doi.org/10.1109/JIOT.2017.2751307
94. J. Miranda, N. Mäkitalo, J. Garcia-alonso, T. Mikkonen, From the Internet of Things to the Internet of People. 2015.
95. M. Conti, A. Passarella, S.K. Das, The Internet of People (IoP): A new wave in pervasive mobile computing. Pervasive Mob. Comput. **41**, 1–27 (2017). https://doi.org/10.1016/j.pmcj.2017.07.009
96. G.S. Ramachandran, B. Krishnamachari, Blockchain for the IoT: Opportunities and challenges. May, (2018) [Online]. Available: http://arxiv.org/abs/1805.02818.
97. C. Lee, L. Nkenyereye, N. Sung, J. Song, Towards a Blockchain-enabled IoT Platform using one M2M Standards, in *2018 Int. Conf. Inf. Commun. Technol. Converg.*, (2018), pp. 97–102
98. A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of Cloud computing and Internet of Things: A survey. *Futur. Gener. Comput. Syst.* **56**(2018), 684–700 (2016). https://doi.org/10.1016/j.future.2015.09.021
99. M. Marjani et al., Big IoT data analytics: Architecture, opportunities, and open research challenges. IEEE Access **5**, 5247–5261 (2017). https://doi.org/10.1109/ACCESS.2017.2689040
100. Cisco, The Internet of Everything Global Public Sector Economic Analysis. pp. 1–13 (2013) [Online]. Available: http://www.cisco.com/c/dam/en_us/about/business-insights/docs/ioe-value-at-stake-public-sector-analysis-faq.pdf.
101. M.H. Miraz, M. Ali, P.S. Excell, R. Picking, A review on the Internet of Things (loT), Internet of Everything (IoE) and Internet of Nano Things (IoNT), in *2015 Internet Technol. Appl.*, (2015), pp. 219–224. https://doi.org/10.1109/ITechA.2015.7317398
102. R. Chandhok, The Internet of Everything, in *2014 IEEE Hot Chips 26 Symp.*, (2014), pp. 1–29. https://doi.org/10.1109/HOTCHIPS.2014.7478826
103. M. Stauffer, Connecting the Internet of Everything, in *2014 IEEE Hot Chips 26 Symp. HCS 2014*, (2016). https://doi.org/10.1109/HOTCHIPS.2014.7478802
104. T. You, Toward the future of internet architecture for IoE: Precedent research on evolving the identifier and locator separation schemes, in *2016 Int. Conf. Inf. Commun. Technol. Converg. ICTC 2016*, (2016), pp. 436–439. https://doi.org/10.1109/ICTC.2016.7763513
105. Z. Nezami, K. Zamanifar, Internet of Things/Internet of everything: Structure and ingredients. IEEE Potentials **38**(2), 12–17 (2019). https://doi.org/10.1109/MPOT.2018.2855439

Chapter 7
Future Challenges of IoT Sensor Networks

Pendukeni Phalaagae, Adamu Murtala Zungeru, Boyce Sigweni, Joseph M. Chuma, and Thabo Semong

7.1 Introduction

Chapter 6 describes the security challenges at various layers (perception, network, cloud, and the user interface) of a wireless IoT sensor network and reviews the proposed mitigation procedures for the security threats at those layers. Chapter 6 also gave a brief overview of the future wireless sensor technologies such as IoE and IoP driven by advances in technology connecting IoT. An increase in IP-enabled devices and the global availability of the Internet is a major driving factor of The IoE and IoP technology. As devices increase in number, threats become prevalent, and hence there is a need for further research to mitigate the challenges that will come with the advance in the IoT technology. This chapter aims to shed some light on the areas that need further research to come up with robust and lightweight security mechanisms for future IoT Networks. Some of these areas that need further research are highlighted in the subsequent sub-chapters.

7.2 Hardware Security

An increase in the number of IoT devices has led to an increase in the in-network security challenges and privacy concerns. However, recent research work has put emphasis on software-based security schemes leaving IoT hardware vulnerable to attacks. Security is IoT devices that are often neglected or treated as an afterthought by IoT device manufacturers. Research has proven that a non-secure hardware platform inevitably leads to a non-secure software stack. Existing security solutions are inadequate since the techniques do not offer strong security and protection against threats. Henceforth future work should work towards the design of security techniques for resource-constrained IoT hardware by considering both hardware security implementations such as cryptographic coprocessor or anti-tampering technologies

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1_7

(e.g., chip or memory protection, self-destruction, etc.) and software solutions in a hybrid manner [1, 2]. However, further research is required to design and develop security mechanisms with low overhead costs for lightweight IoT hardware.

7.3 Lack of Lightweight Cryptographic Algorithms

Lightweight security will always be one of the key future research areas in IoT because of the resource-constrained nature of IoT sensor networks. The process of encrypting communication in the IoT sensor networks is computationally complex regarding cost, energy, and memory for small-sized and resource-constrained IoT devices. Complex security algorithms affect the performance of devices. Therefore, algorithms targeted at security and optimized resource utilization at low computation cost for IoT devices should be further investigated for lightweight IoT solutions such as key management, access authentication, access control for specific requirements for specific IoT sensor networks [3].

7.4 Lack of Lightweight Trust Management Systems

Trust needs to be established between neighboring nodes in the network. However, an attacker can join the network, masquerade, and recommend itself to other nodes in the network and attract traffic, which then makes it easy for them to forward attacks in the network. Therefore, a secure trust management system needs to be deployed to maintain a high level of trust between resources-constrained nodes in the network. Similarly, future research work should look at the assessment of correlation agreement among IoT nodes and its role of autonomous and intelligent trust management among nodes at all layers of the IoT sensor networks. Precision knowledge apprehension is another security goal to be considered in future works to increase the high level of trust among nodes in a network include owing to the requirement of maintaining confidentiality, quality of service, and reliable communication. For these reasons, it is essential to appropriately enforce trust management starting from the characterization of different threats and attacks at each specific level of the IoT architecture [2].

7.5 Lack of Lightweight Secure Routing Protocols

Assuring end-to-end secure communication among devices is a major challenge in IoT sensor networks. Data encryption has been proven to be the most effective method for secure communication across wireless sensor network communication where communication channels are more prone to attacks and data breaches.

According to research, using efficient encryption methods with low computational costs can play an important role in reducing security risks and attacks in wireless IoT communication. However, an encryption algorithm must be chosen properly to cater for the resource-constrained nature of IoT nodes. The key issues in designing secure mechanisms or algorithms are to deal with the trade-off between security, performance, and cost. Therefore, a secure routing algorithm proposed should have the capacity to secure the network with optimal use of memory-constrained IoT devices to make IoT deployment sustainable.

7.6 Lack of Lightweight Anti-Malware Solutions

The widespread adoption of IoT devices has attracted attackers to abuse them by causing an increase in malicious software (malware). Several types of malware can affect IoT software applications and IoT sensor networks hardware devices. Research has demonstrated that malware is a serious threat that can destroy an IoT device or have the attacker get authority and control over the IoT system. The most common adverse malware include rootkits, ransomware, bots, logic bombs, virus, worms, and Trojans. In order to keep pace with the increased adoption of IoT devices and an increased number of malware attacks, researches need to design lightweight malware detection solutions for IoT resource-constrained environments. This opens up new research challenges that need to be addressed by implementing a physical level cryptosystem with emphasis on low-power and low-cost security mechanisms against malware [4]. For future work malware, obfuscation may be considered to increase the malware detection rate, and the efficiency of the mechanisms may be assessed for their implementation in resource-constrained IoT devices [5].

7.7 Summary

The rapid evolution of IoT sensor networks has brought along many benefits. However, IoT has also brought about ambiguity and several security concerns to IoT adopters. Security issues are the most critical challenges that need to be addressed in promoting the adoption and development of IoT systems. This work presents security and privacy issues and their solutions. The work suggests a layered approach to expose security issues and challenges at each layer of the IoT architecture and proposes techniques used to mitigate these challenges. IoT sensor networks undergo security and privacy issues such as hardware vulnerabilities, secure routing issues, and a lack of interoperable standards for heterogeneous networks. Finally, directions and perspectives are drawn and discussed for future directions in securing IoT sensor networks IoT covering evolving areas such as artificial intelligence, Blockchain technology, sensor Internet of People, context-aware sensing, cloud

infrastructure, security and privacy, and the Internet of Everything. IoT has endless opportunities and application domains. However, there are shortcomings that arise with the adoption of IoT such systems' security for resource-constrained IoT devices. In conclusion, this work aims to highlight IoT sensor networks' security challenges and to unearth research opportunities to address these challenges. Therefore, from the survey, opportunities such as Blockchain, machine learning, and the development of context-aware applications, IoP and IoE, should be further investigated to review real-life examples and analysis of their effectiveness towards enhancing privacy and security in IoT sensor networks. The convergence of IoT sensor networks with neural networks, deep learning, predictive modeling, and low-power protocols and algorithms can be investigated for enhancing security in IoT sensor networks.

References

1. F. Rahman, M. Farmani, M. Tehranipoor, Y. Jin, Hardware-assisted cybersecurity for IoT devices, in *2017 18th Int. Work. Microprocess. SOC Test Verif.*, (2017), pp. 51–56. https://doi.org/10.1109/MTV.2017.16
2. M. Frustaci, P. Pace, G. Aloi, G. Fortino, Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet Things J. **5**(4), 2483–2495 (2018). https://doi.org/10.1109/JIOT.2017.2767291
3. Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, D. Qiu, Security of the Internet of Things: Perspectives and challenges. Wirel. Netw. **20**(8), 2481–2501 (2014). https://doi.org/10.1007/s11276-014-0761-7
4. N. Sklavos, Malware in IoT software and hardware, in *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE'16)*, (Barcelona, 2017), pp. 8–11
5. Su, V. Danilo Vasconcellos, S. Prasad, S. Daniele, Y. Feng, K. Sakurai, Lightweight classification of IoT malware based on image recognition. Proc. Int. Comput. Softw. Appl. Conf. **2**, 664–669 (2018). https://doi.org/10.1109/COMPSAC.2018.10315

Index

A
Access attack, 89
Access control, 26, 27, 29, 51
Access control attacks, 93
Advanced Encryption Standard (AES), 100, 111
Alternating current (AC) electricity, 28
Amazon Echo, 12
Ambient solar irradiance, 70
Ambient temperature, 70, 71
Anti-malware solutions, 121
API security, 105
Applications
  commercial IoT, 12, 13
  consumer IoT, 11, 12
  domains, 10, 11
  IIoT, 13
Application-specific vulnerabilities, 94
Artificial intelligence (AI), 12, 107–109, 111
Attention in the area of IoT Security, 91
Authentication, 98–100, 102, 105, 106, 108, 111

B
Batteries, 61
  bank system voltage, 61, 62
  capacity, 61
  charge and discharge system, 75, 76
  lead acid, 74–76
  sizing, 66–67
  voltages and output sinusoidal signal, 79
Battery bank system voltage, 61, 62
Blockchain technology, 107, 108
Bluetooth, 9, 16
Booting attack, 87

C
Carbon emission, 2
Cellular technologies, 18
Charge controllers, 57
Cipher block chaining (CBC) mode, 100
Cloud infrastructure, 109
Cloud layer
  advantages, 91
  flooding attack, 92, 93
  malware injection, 92
  MITC attacks, 91
  secure cloud computing, 104
  SMPC, 104, 105
  SQLi, 92
Commercial IoT, 12, 13
Communication devices, 2
Communication protocols, 4, 6
Communication technologies, 4
  IoT sensor networks, 13–15
  WBANs, 15, 16
  wireless IoT network, 13
  WLAN, 17, 18
  WPANs, 16, 17
  WWAN, 18–20
Consumer IoT, 11, 12
Context-aware sensing, 108, 109
C++ script language, 43, 48
Cryptographic algorithms, 100, 101, 120
Cryptographic hashing, 106

A. Murtala Zungeru et al., *Green Internet of Things Sensor Networks*,
https://doi.org/10.1007/978-3-030-54983-1

D
Data compression, 4, 6, 7
Data encryption standard, 99
Data gathering, 25
Data management, 106
Data processing, 25
Data security, 108
Data transit, 89
Denial of service (DoS), 89
Diffie–Hellman (DH) algorithm, 100, 106
Diode, 69
Direct current (DC) electricity, 28
Display unit, 29
Distributed control system (DCS), 13
Distributed denial of service (DDoS), 86, 89
DoS attack, 86

E
Eavesdropping, 88
Electricity, 26
Elliptic curve cryptography (ECC), 100, 104, 111
Elliptic curve Diffie–Hellman (ECDH) algorithm, 106
Encryption algorithm, 99
Encryption mechanism, 102
Energy balancing technique, 64
Energy harvesting, 4–6, 26–28, 56, 75, 78
Energy production, 55
EnOcean technology, 20
Exclusive-OR (XOR) operations, 106

F
Fake node, 85
False data injection attacks (FDIA), 87
Flooding attack, 92, 93
Fourth-order Chebyshev low pass filter, 72
Future challenges (IoT sensor networks), 119–122
Future IoT networks, 119–122

G
Global System for Mobile (GSM), 27, 30, 31
Google Home, 12
Green Internet of Things, 57
Green Internet of Things Sensor Networks (GIoTSNs)
  communication protocols, 6
  communication technologies, 4
  components, 2
  data compression, 6, 7
  design technologies, 4
  energy harvesting, 4–6
  environmental hazards, 3
  leverage technologies, 4
  principles, 4
Grid configuration, 58
GSM-call, 36, 40
GSM/GPRS module, 30

H
Hardware security, 98, 99, 119, 120
Hashing algorithms, 106
Home automation, 11
HTTP flood attack, 92, 93
Hybrid wiring configuration, 60, 61

I
Industrial IoT (IIoT), 13
Industry 4.0, 13
Information privacy, 105, 106
Intelligent motion sensors, 12
Internet-based network, 1
Internet of Everything (IoE), 110, 111
Internet of Things (IoT)
  big data, 2
  GIoTSNs, 3, 4
  machine/things to machine/things, 2
  people to machine/things, 2
  people-to-people, 2
  sensor networks, 2, 3
Inverting system (DC/AC converter), 65
IoT architecture, 84, 97, 100, 111
IoT automation, 12
IoT devices, photovoltaic system, *see* Photovoltaic system
IoT sensor networks, 2, 3
  applications, 9, 11
  communication (*see* Communication technologies)
  security challenges (*see* Security challenges)
  security mechanisms (*see* Security mechanisms)
IoT technology, 119

K
Key agreement, 106
KiCAD (PCB design software), 48
Kirchhoff's current law (KCL), 59, 68

Kirchhoff's laws, 58
Kirchhoff's voltage law (KVL), 59

L
Lead acid battery, 74–76
Lightweight, 119, 120
  anti-malware solutions, 121
  cryptographic algorithms, 120
  secure routing protocols, 120, 121
  trust management systems, 120
Lightweight encryption, 99–101
LoRa, 19
6LowPAN, 16, 17
Low-Power Wide Area Network (LPWAN), 18, 21
Low-Power Wireless Personal Area Networks (Low PANs), 16

M
Machine learning, 12, 98, 99, 107, 109
Machine learning algorithms, 108
Machine-to-machine communication, 88
Malicious code injection, 87, 93, 94
Malicious data, 85
Malicious software attacks, 92
Malvertising, 92
Malware Injection, 92
Man-in-the-cloud (MITC) attacks, 91
Man-in-the-middle (MITM) attack, 90
Maximum power point tracking (MPTT), 70–72, 77–79
  duty cycle of power switch, 56
Message Authentication Code (MAC), 100
Message Integrity Code (MIC), 100
MQTT-Call, 36, 39
MQTT data packet, 48
Multi-hop communication, 6

N
Natural energy, 4
Near Field Communication (NFC), 15
Network layer
  access attack, 89
  data transit, 89
  DoS/DDoS, 89
  eavesdropping, 88
  encryption mechanism, 102
  MITM attack, 90
  phishing site, 88
  routing attack, 89, 90
  secure communication, 102, 103
  trust recommendation, 103, 104
Node capture/tempering, 85

O
Off-grid PV system
  ambient solar irradiance, 56
  battery/supercapacitor, 65
  circuit representation, 63
  cost options, 55
  hybrid wiring configuration, 63
  schematic diagram, 62, 63
  source, 55
  specifications, 63
Optical energy, 5

P
Peak solar radiation hours (PSRH), 63–66
Perception layer
  authentication, 98, 99
  booting attack, 87
  DDoS/DoS, 86
  fake node, 85
  FDIA, 87
  hardware security, 98, 99
  HP survey, 84
  lightweight encryption, 99–101
  malicious code injection, 87
  malicious data, 85
  node capture/tempering, 85
  protecting sensor data, 100, 102
  replay attack, 85
  resources-constrained IoT devices, 84
  side-channel attack, 85, 86
  sleep deprivation, 87
Perturb and observe (P&O) method, 71, 73
Phishing site, 88
Photovoltaic (PV) system
  ambient solar irradiance, 70
  ambient temperature, 70, 71
  array output, 57
  array sizing, 63–66
  battery sizing, 66–67
  charge controllers, 57
  current and voltage relationship, 68
  design and simulation, 67
  energy production, 55
  equivalent circuit, 67, 68
  hybrid configuration, 60
  hybrid wiring configuration, 60, 61

Photovoltaic (PV) system (*cont.*)
irradiance variations
I–V plot, 68, 69
P–V plot, 69
KCL, 68
maximum power point tracking, 70–72
MPPT, 71, 72
off-grid, 55, 56, 62, 63
optimized operations, 71
power conditioner, 56
power quality and efficiency, 55
PSRH, 63–66
PV behaviors, 67
solar array system (*see* Solar array system)
solar panel, 57
spatial constraints, 57
technical specifications, 59, 60
transducer, 56
Playback attack, 85
Power conditioner, 56
Power consumption, 1, 2
Processing unit, 29
Protecting sensor data, 100, 102
Public key encryption, 100
Public–private partnerships (PPP), 12
PV array sizing, 63–66
PV behaviors, 67
PV transducer, 56
Python script language, 32

R
Radio frequency identification (RFID), 9, 15, 16, 108
Raspberry Pi, 12, 38
Raspberry Pi single-board computer, 30, 31
Real-time applications, 9
Remote Code Execution (RCE) attack, 93
Remote socket, 40
Renewable energy sources, 55
Replay attack, 85
Representational State Transfer (REST), 105
Rivest, Shamir and Adleman (RSA) algorithm, 100
Routing attacks, 89, 90, 103
RSA algorithm, 106

S
Samsung Smart Things, 12
Secure cloud computing, 104, 111
Secure communication, 102, 103
Secured smart home switching system, 27, 28
Secure multi-party computation (SMPC), 104, 105
Secure routing protocols, 120, 121
Security, 26, 27, 39, 40, 51
Security attacks, 102
Security challenges
cloud layer, 90–93
network layer, 88–90
perception layer, 84–88
threats and attacks, 84
user interface, 93, 94
Security mechanisms, 119–121
AI, 107
Blockchain technology, 107, 108
cloud infrastructure, 109
cloud layer (*see* Cloud layer)
context-aware sensing, 108, 109
intra-layer information, 97
IoE, 110, 111
machine learning for data security, 108
network layer (*see* Network layer)
perception layer (*see* Perception layer)
sensor IoP, 109, 110
user interface, 105–106
Security modes, 39, 40
Security threats, 97
Self-sufficient energy house, 57
Sensor IoP, 109, 110
Sensor nodes, 1, 2, 4–7, 9, 83
Service-level Agreements (SLA's), 106
Sensors
conventional networks, 1
GIoTSNs (*see* Green Internet of Things Sensor Networks (GIoTSNs))
WSNs (*see* Wireless sensor networks (WSNs))
Short Message Service (SMS), 29
Shunt current, 69
Side-channel attack, 85, 86
Sig-Fox, 18, 19
Simple object access protocol (SOAP), 105
Simulation, 73, 77–79
Single-board (SBC) computer, 29
Single-phase inverter, 72
Sleep deprivation, 87
Smart cities, 4, 12, 13
Smart devices, 2, 7
Smart Grid, 4
Smart home, 11, 12, 27, 28
building system, 26
functionalities, 27
requirements, 29

secured smart home switching system, 27–28
smart IoT devices and applications, 27
software implementation, 32
switches/sockets, 34
switching system, 50
WSN, 26
Smart hub, 27, 28
block diagram, 30, 31
circuit diagram, 31
components, 30
functions, 28, 29
hardware selection
display unit, 29
GSM module, 30
input method, 29
processing unit, 29
PCB designs, 31, 32
software design
GSM-call, 36, 40
MQTT-Call, 36, 39
procedures and flowcharts, 32
system-locked, 32, 33
system-unlocked, 32, 34, 35
target-device-edit, 34, 36, 37
test results, 38, 41
Smart IoT devices
aim, 26
BIUST Off-Grid secured smart house, 51, 52
definition, 26
design, 25
implementation, 25
limitations, 25
objectives, 26
scaled-down version of house, 50
secured smart home switching system, 27, 28
smart hub (*see* Smart hub)
smart switch and smart socket design, 39–46
smart switchboard, 46–49
system implementation, 50, 52, 53
Smart kettles, 25
Smart light bulbs, 25
Smart refrigerators, 25
Smart sockets, 27, 28
Smart switch, 27
Smart switchboard, 27, 28
block diagram, 46–47
circuit diagram, 47–48
data flow, 47
functions, 46
hardware selection, 46
PCB designs, 48
requirement, 46
software design, 48, 49
test results, 48, 50
Smart switches, 25–29, 31, 32, 43, 44
Smart switch/socket
block diagram, 42
circuit diagram, 42–43
functions, 39–41
hardware selection
processing unit, 41, 42
switching unit, 42
PCB designs, 43
software design, 43–44
switch-main-loop, 44
test results, 44, 46
Smart vehicles, 12
Smartness, 25
Sniffing attacks, 94
Solar array system, 79
batteries, 61
design, 57
grid configuration, 58
hybrid wiring configuration, 60, 61
parallel array configuration, 57–59
physical layout, 57, 58
series array configuration, 57–59
solar panels, 57
technical specifications, 59, 60
voltage rating, 59
Solar cells, 64, 68
Solar irradiance, 77
Solar panels, 57
parallel array configuration, 58, 59
series array configuration, 58
Solar system
design, 56
specifications, 61
SQL injection (SQLi), 92
Storage system, 26
Supply energy, 55
Switching system, 26–28, 50, 52, 53
Switching unit, 42
Switch-main-loop, 44
Symmetric algorithms, 106
System-unlocked procedure, 34

T

Target-device-edit, 34, 36, 37
Transport layer security (TLS), 105
Trust-aware routing framework (TARF), 104

Trust management systems, 120
Trust recommendation, 103, 104

U
Universal Serial Bus (USB) modem, 27
User-based collaborative method, 103
User interface
  data management, 106
  information privacy, 105, 106
  key agreement, 106
User interface/application layer
  access control attacks, 93
  application-specific vulnerabilities, 94
  malicious code injection, 93, 94
  sniffing attack, 94
User interface protection, 105

V
Vehicle-to-grid (V2G) networks, 106
Vehicular ad hoc Networks (VANETs), 100

W
Wireless Body Area Networks (WBANs), 15, 16
Wireless communication, 2
Wireless Fidelity (Wi-Fi), 17, 18
Wireless LAN (WLAN), 17, 18
Wireless Personal Area Networks (WPANs), 16, 17
Wireless sensor networks (WSNs), 26, 27
  applications, 1
  power consumption, 1
  sensor nodes, 1
Wireless Wide Area Network (WWAN), 18–20
Worldwide Interoperability for Microwave Access (WiMAX), 19, 20

Z
Zig-Bee, 9, 16
Z-Wave, 16, 17
  and Zig-Bee enable communication, 11