



IntechOpen

Internet of Things

Edited by Fausto Pedro García Márquez



Internet of Things

Edited by Fausto Pedro García Márquez

Published in London, United Kingdom



IntechOpen





Supporting open minds since 2005



Internet of Things

<http://dx.doi.org/10.5772/intechopen.91605>

Edited by Fausto Pedro García Márquez

Contributors

Marek Babiuch, Jiri Postulka, J.S. Prasath, Sergii Kapshtyk, Mikhail Ilchenko, Teodor Narytnyk, Vladimir Prisyazhny, Sergey Matvienko, Ganga Dhandapani, V. Ramachandran, Ahmad Showail, Fausto Pedro García Márquez

© The Editor(s) and the Author(s) 2021

The rights of the editor(s) and the author(s) have been asserted in accordance with the Copyright, Designs and Patents Act 1988. All rights to the book as a whole are reserved by INTECHOPEN LIMITED. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECHOPEN LIMITED's written permission. Enquiries concerning the use of the book should be directed to INTECHOPEN LIMITED rights and permissions department (permissions@intechopen.com).

Violations are liable to prosecution under the governing Copyright Law.



Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 3.0 Unported License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be found at <http://www.intechopen.com/copyright-policy.html>.

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in London, United Kingdom, 2021 by IntechOpen

IntechOpen is the global imprint of INTECHOPEN LIMITED, registered in England and Wales, registration number: 11086078, 5 Princes Gate Court, London, SW7 2QJ, United Kingdom

Printed in Croatia

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Additional hard and PDF copies can be obtained from orders@intechopen.com

Internet of Things

Edited by Fausto Pedro García Márquez

p. cm.

Print ISBN 978-1-83968-849-2

Online ISBN 978-1-83968-850-8

eBook (PDF) ISBN 978-1-83968-851-5

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,400+

Open access books available

132,000+

International authors and editors

160M+

Downloads

156

Countries delivered to

Our authors are among the
Top 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Meet the editor



Fausto Pedro García Márquez has been a full professor at the University of Castilla–La Mancha (UCLM), Spain, since 2013. He is an honorary senior research fellow at Birmingham University, UK, and a lecturer at the Postgraduate European Institute. From 2013 to 2014, Dr. Márquez was a senior manager at Accenture. He obtained his European Ph.D. with a maximum distinction. He has been awarded several prizes, including the Runner (2020) and Advancement Prizes (2018) for Management Science and Engineering Management; First International Business Ideas Competition Award (2017); Runner (2015), Advancement (2013), and Silver (2012) Prizes from the International Society of Management Science and Engineering Management (ICMSEM); and Best Paper Award, *Renewable Energy* (2015). He has published more than 150 papers in reputable journals. He is the author and editor of thirty-one books and five patents. He is an editor for five international journals and a committee member of more than forty international conferences. He has been the principal investigator for four European projects, six national projects, and more than 150 projects for universities and companies. His main interests are artificial intelligence, maintenance, management, renewable energy, transport, advanced analytics, and data science. He is an expert in the European Union in AI4People (EISMD), and ESF. He is also the director of the Ingenium Research Group.

Contents

Preface	XIII
Chapter 1 Introductory Chapter: Internet of Things <i>by Fausto Pedro García Márquez</i>	1
Chapter 2 The Internet of Things Space Infrastructure. Current State and Development Prospects <i>by Mikhail Ilchenko, Teodor Narytnyk, Vladimir Prisyazhny, Segii Kapshtyk and Sergey Matvienko</i>	5
Chapter 3 Internet of Things Security and Privacy <i>by Ahmad J. Showail</i>	25
Chapter 4 An IoT Based Cloud Deployment Framework for Effective Classification of Machine Conditions <i>by Ganga Dhandapani and V. Ramachandran</i>	39
Chapter 5 Compound Cryptography for Internet of Things Based Industrial Automation <i>by J.S. Prasath</i>	63
Chapter 6 Smart Home Monitoring System Using ESP32 Microcontrollers <i>by Marek Babiuch and Jiri Postulka</i>	81

Preface

This book provides relevant theoretical frameworks and the latest empirical research findings in the Internet of Things (IoT). It is written for professionals who want to improve their understanding of the strategic role of the IoT at the global economy level, at networks and organizations, in teams and work groups, in information systems, and at the level of individuals as players in networked environments.

The IoT is a closed-loop system in which a set of sensors is connected to servers via a network. The data from sensors are stored in a database and then analysed by IoT analytics. The results are usually employed by either humans, machines, or software to make decisions to the operation of the system. The system is a general one that uses different types of sensors that monitor things such as weather conditions, images, velocity, and more.

New data science techniques have appeared in the last few years to solve the complex and robust problems generated in the IoT. The volume, variety, velocity, complexity, and so on of the data obtained by the IoT require new approaches to solve problems in which quality and computational cost are the main variables. Some problems to be addressed by the IoT are maintenance, management, optimization, planning, decision-making, operations management/research, safety, and security in fields such as transportation, energy, banking/finance, social science, media, and marketing. IoT is related to the concept of “smart” technology, such as smart cars, smart homes, smart cities, smart manufacturing, smart banking, and more.

Simulation methods are employed to determine the design of a future IoT system, therefore, an anticipated load generated by its sensors. The results can be compared with real ones, leading to conclusions.

Statistics and machine learning are methods applied to IoT analytics. These include multivariable linear regression, time series forecasting, dimensionality reduction, clustering, classification, artificial neural networks, support vector machines, and hidden Markov models. These types of methods can be used individually or in combination.

Performance evaluation and modeling are operations research techniques. They are employed mainly to study the computing facilities used in fog computing and high-layer server(s). They are also applied to the supporting IP network that provides connectivity between sensors, actuators, fog computing devices, and higher-layer servers. These techniques can also be utilized in sensors and actuators to set the capacity of any IoT system or any of its layers. Performance evaluation is also considered for the computational time employed for the IoT, mainly due to the end-to-end response time. It is applied to working IoT, but it cannot be employed to design IoT because of many unknown parameters. In this case, it is employed as a prototype or a model, the former being the most expensive and the latter being the most utilised.

The IoT system should be defined by a model based on interconnected layers. Each layer is given by a functionality. Some examples of layers are networking, IoT controllers and devices, data storage, fog computing, and data abstraction. The model is called a reference model.

Advances in all the layers are happening rapidly and thus there is a need to develop laws and standards to guarantee human rights and address ethical issues. In addition, the layers must have the required security (e.g., communication, protocols, authentication, networks).

Fausto Pedro García Márquez
University of Castile-La Mancha,
Spain

Introductory Chapter: Internet of Things

Fausto Pedro García Márquez

1. Overview of Internet of Things

Internet of Things (IoT) can be understood as a closed loop system, where a set of sensors are connected by a network to servers. The data from sensors are stored in a database. The data is then analyzed by IoT analytics, where the results are usually employed to make decisions to the operation of the system [1]. The system must be seen as a general one that monitors by using any type of sensor, e.g., weather conditions, images, velocity, etc. [2]. The decision making can be done by human, but also by a machine, software, etc.

New data science techniques are appearing in the last few years to solve the complex and robust problems generated in IoT [3]. The volume, variety, velocity, complexity, etc. of the data obtained by IoT require of new approaches to solve the problems for decision making [4], where quality and computational cost are the main variables to evaluate them. Some examples to be addressed by IoT are maintenance management, optimization, planning, decision making, operations management/research, safety, security, etc., in fields as transport, energy, bank/finance, social science, media, marketing, etc., and, nowadays, the called Smart XX and e-XX, where XX can be any term, e.g., car, home, cities, industry, manufacturing, bank, agriculture, farm, environment, water, metering, health, etc.

Simulation methods are employed to determine the design of a future IoT system, therefore, an anticipated load generated by its sensors [2]. The results can be compared with the real ones, leading to get conclusions.

Statistics and Machine Learning are also extended methods applied to IoT analytics, e.g., multivariable linear regression, time series forecasting, dimensional-ity reduction, clustering, classification, artificial neural networks, support vector machines, and hidden Markov models [5]. These types of methods are being of great interest for the researchers, where they are working developing new ones, or hybrid based on the use of two or more different methods [6].

Performance evaluation and modeling can be considered as an operations research technique. They are employed mainly to study the computing facilities used in fog computing and high-layer server(s). On the other side, they are applied to the supporting IP network that provides connectivity between sensors, actuators, fog computing devices, and higher-layer servers. Finally, it can be utilized in sensors and actuators. The performance is usually employed to set the capacity of any IoT system, or in any of the layers. The performance is also considered for the computational time employed for the IoT, mainly jitter of the end-to-end response time and end-to-end response time. The performance is applied to working IoT, but it cannot be employed to design of IoT because of many parameters are unknown yet. In this case, it is employed a prototype or a model, being the first one the most expensive and the last one the most utilized.

The IoT system should be defined by a model, which is based on interconnected layers. Each layer is given by a functionality. Some examples of layers are: networking, IoT controllers and devices, data storage, fog computing, data abstraction, etc. The model is named reference model.

The advances are going very fast in all the layers, and now the governments and main responsibilities are working to develop laws and standards to guarantee the main rights for human, and also for the ethical issues. On the other side, the layers must have the required security from different point of views, e.g., communication, protocol, authentication, network, etc.

The reduction of power together to the transmission distance between the inter-devices are issued to cover and research in IoT. In this sense, ZIGBEE (has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device), Bluetooth (short-range wireless technology standard used for exchanging data between fixed and mobile devices over short distances using ultra high frequency radio waves in the industrial, scientific and medical (bands 2.402–2.48 GHz, and building personal area networks), Wireless Sensor Network (WSN), Radio Frequency Identification (RFID, reader and transmits digital data that contain identification and other information required), Wi-Fi (wireless network protocols, based on the IEEE 802.11 family of standards), Cellular networks, IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), etc. are technologies that appear to solve it. On the other side, the demand of energy is rising. Efficient and low power consumption devices are being developed and employed, e.g., smart lighting systems are being utilized in cities to manage the public lighting, which is estimated to be about 10% of the overall energy consumed in a city, by using led, local control units that save the data in cloud and is then analyzed.


Other examples of IoT use are shown as follows: waste and garbage management; Smart homes domain; agriculture domain (e.g., sampling and mapping of soil, irrigation, fertilizer, crop disease and pest management, crop monitoring, forecasting and harvesting); Industrial Internet of Things (IIoT); Energy conservation (including smart grid, Energy Internet of Things (EIoT), Smart Metering (SM) and Advanced Metering Infrastructure); Healthcare (e.g., monitor patients distantly, Monitoring of Blood Glucose Level, Electrocardiogram Monitoring, Blood Pressure Monitoring, Body Temperature Monitoring, Monitoring of Blood Oxygen Saturation, Rehabilitation System, Wheelchair Management).

Author details

Fausto Pedro García Márquez
Ingenium Research Group, University of Castilla-La Mancha, Spain

*Address all correspondence to: faustopedro.garcia@uclm.es

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Kopetz, H. Internet of things. In *Real-time systems*, Springer: 2011; pp 307-323.
- [2] Weber, R.H.; Weber, R. *Internet of things*. Springer: 2010; Vol. 12.
- [3] Márquez, F.P.G.; Lev, B. *Data science and digital business*. Springer: 2019.
- [4] Marugan, A.P.; Márquez, F.P.G. *Decision-making management: A tutorial and applications*. 2017.
- [5] Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Computer networks* 2010, 54, 2787-2805.
- [6] Márquez, F.P.G.; Lev, B. *Advanced business analytics*. Springer: 2015.

The Internet of Things Space Infrastructure. Current State and Development Prospects

Mikhail Ilchenko, Teodor Narytnyk, Vladimir Prisyazhny, Segii Kapshtyk and Sergey Matvienko

Abstract

This chapter presents an overview of possibilities for existing Satellite Communication Systems utilization to provide Internet of Things Services. It is shown that existing Satellite Communication Systems provide traffic transmission for Internet of Things Systems with Cloud Architecture. The propositions on possibility of Fog and Edge computing implementation in Satellite Communication Systems are proposed. The ways for Low-Earth Orbit and Geostationary Orbit Satellite Communication Systems modernization for Fog and Edge computing implementation for the Internet of Things Systems are presented. To increase the efficiency of IoT data processing and the reliability of Internet of Things Data Storage, it is proposed to generate an Orbital Cloud Data Storage in Geostationary Orbit, which consists of several Geostationary Orbit Satellites - Cloud Computing Data Centers. Methods for access provision to the Orbital Cloud Data Storage using Geostationary Orbit High-Throughput Satellites and satellites from the structure of Low-Earth Orbit Satellite Communication Systems are proposed. The issues of interaction between Orbital Cloud Data Storage and ground-based Cloud Data Processing and Storage Infrastructure are briefly considered. The orbital slots in Geostationary Orbit are proposed for location of Geostationary Orbit Satellites - Cloud Computing Data Centers.

Keywords: IoT system, satellite communication system, geostationary orbit, low-earth orbit, satellite constellation, IoT smart things

1. Introduction

In the last decade, the Internet of Things has become an important component of modern info communications. According to Transforma Insights [1], the total number of active Internet of Things (IoT) devices in 2019 was 7.6 billion. The number of active IoT devices is expected to grow up to 24.1 billion by 2030, with a CAGR of 11%. For the forecasted period, the short-range technologies will remain the main type of IoT device connection: Wi-Fi, Bluetooth, Zigbee. The number of IoT device connections to cellular networks is predicted to increase from 1.2 billion in 2019 up to 4.7 billion in 2030. The need to provide a large number of connections and IoT device traffic transmission has become the main driver for the development and implementation of a new 5G mobile broadband standard [2].

The development of the Internet of Things is constrained by the limited coverage of terrestrial mobile broadband networks, which ones for commercial reasons cover areas with relatively high population densities. It is possible to expand the area of providing IoT services by using the satellite telecommunication systems resource, with specified and widespread application. The purpose of this article is to present the overview of the readiness of existing Satellite Communications Systems to provide IoT Services, and describe potential directions for the development of this communications sector in future.

2. The internet of things space infrastructure current state

Up to now, the Cloud Computing [3] technology has been the prevailing architecture for IoT systems. According to this architecture, IoT devices (hereinafter referred to as IoT Smart Things) transform monitored physical parameters into electronic signals and transmit relevant information to the Cloud Computing Data Centers for information processing in accordance with the IoT service purpose, information storage and archiving. If it is necessary to implement any actions, the Cloud Computing Data Center delivers control actions bursts to the IoT Smart Things.

Current satellite telecommunications systems are used to transmit IoT information Traffic for IoT systems based on the Cloud Computing Architecture. **Figure 1** shows the model of the IoT System built on Cloud Computing Architecture and using a satellite telecommunications system. The IoT Smart Things are located at the lowest level

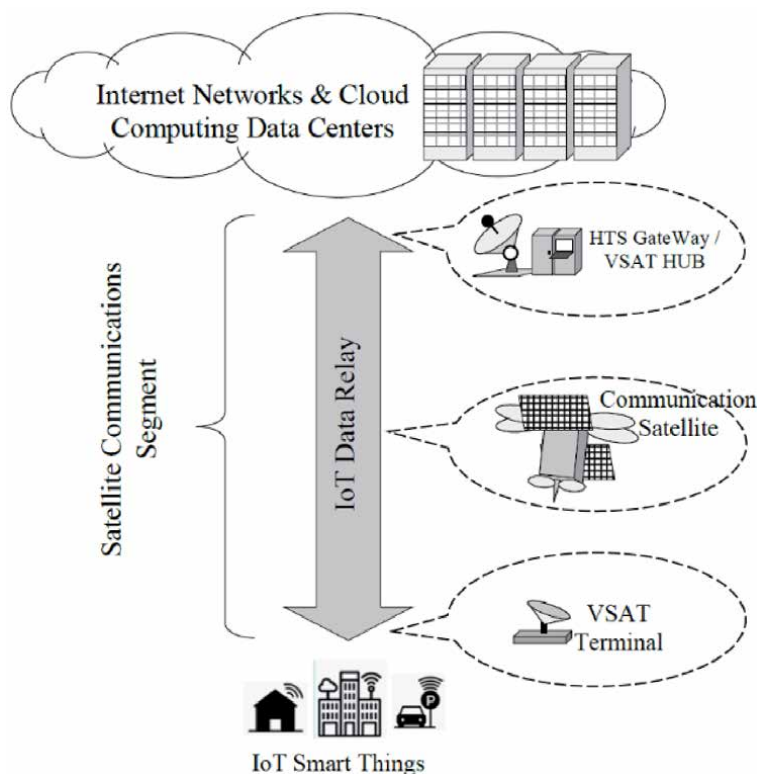


Figure 1. The IoT system model, built on the basis of the cloud computing architecture and using a satellite telecommunications system.

of the hierarchical structure of the IoT System. This group of devices includes Sensors, i.e. devices that transform physical processes into electrical signals and form IoT Information Bursts based on these electrical signals, as well as devices that implement physical actions based on received commands - Actuators. The Cloud Computing Data Center is at the top of the hierarchical level of the system.

The Satellite Communication Segment provides with IoT Data Relay Channel from IoT Smart Thing Sensor, to a Cloud Computing Data Center and vice versa, to the IoT Smart Thing Actuator. The satellite communication channel is established with the following elements utilization:

- VSAT terminal, which is located in close proximity to IoT Smart Things, sensors and actuators, and provides connection of these devices on short-range technology basis: Wi-Fi, Bluetooth, Zigbee. The VSAT interface to the local network or to the terrestrial local communication network is the system boundary for a satellite telecommunications system;
- Communication Satellite, which provides retransmission of the IoT Smart Things information. For the hierarchical model of the Cloud IoT System Architecture mission, the type of satellite payload, or repeater of a telecommunications satellite is not essential: either it will be Transparent Transponder or Regenerative Transponder [4];
- The VSAT-Network HUB or Gate Way. Generally, this facility is connected to the Internet Backbone through which IoT data transmission to the Cloud Computing Data Center is provided.

Figure 2 shows examples of utilization of various types of Satellite Communication Systems to support the operation of the IoT Systems and to provide IoT Services. The Figure 2 shows Satellite Communication Systems using two orbits types: Low Earth

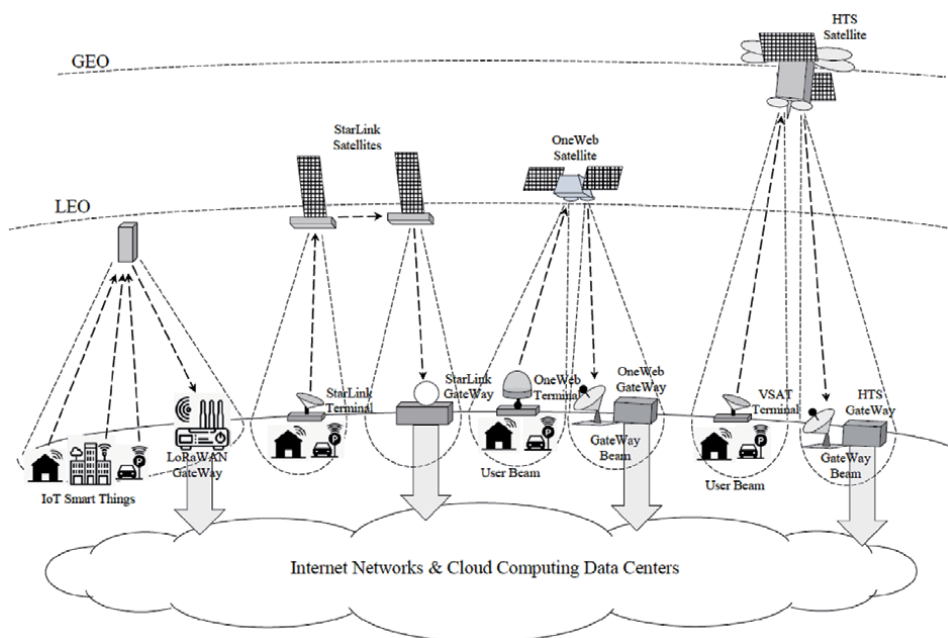


Figure 2. Satellite communication systems utilization for providing operation of IoT systems.

Orbit (LEO) and Geostationary Orbit (GEO). Considering the fact that Broadband Access Satellite Systems in LEO and in Medium Earth Orbit (MEO) do not have essential differences in their construction architecture, **Figure 2** does not show MEO Satellite Communication Systems, like O3b System.

The first application option of LEO satellite communication systems is utilization of the IoT narrow-band long range data transmission modified protocol LoRaWAN with exploitation of the CubeSat form factor spacecraft [5]. IoT Smart Things within the CubeSat coverage zone, transmit IoT information bursts by LoRaWAN protocol. CubeSat receives and retransmits the signals, which come at the LoRaWAN Gate Way. The LoRaWAN Gate Way is connected with the local communication system and provides IoT data transmission through the Internet network to the Cloud Computing Data Center. The IoT Service Control actions bursts are transmitted in the opposite direction. In this architecture, the satellite segment is used as a radio extension link, i.e. as a tool providing the transmission range increase of the LoRaWAN protocol signals.

StarLink and OneWeb are the perspective broadband access satellite systems, which are currently at the different stages of the Satellite Constellation development. These Systems are also capable to provide the IoT data transmission.

The architecture of the StarLink System has been changing several times during the system design and the spacecraft development. The StarLink Constellation with satellites located on LEO with 550 km high is currently in the stage of satellite launches realization and constellation development [6]. StarLink satellites form steerable beams that cover End User Terminals and Gate Ways Earth Station, and provide broadband access satellite service. The external interface of the StarLink end user terminal serves as the StarLink system interface for IoT systems. IoT Smart Things are being connected to a StarLink terminal via a short-range radio access network, for example WiFi or LAN. Then the IoT information packets are being transmitted in the information up-link flow to the StarLink satellite, where it is being routed towards the beam covering the gate way at a given time. It has been often mentioned in the press that StarLink satellites provide an Inter-Satellite Optical (laser) Link [7]. In this case, as shown in **Figure 2**, the IoT information packet can be transmitted over Inter-Satellite Optical Line from the satellite which covers the StarLink end user terminal with connected IoT Smart Things, to a satellite which covers a Gate Way Earth Station. The StarLink Gate Way Earth Station interface is connected to the Internet Backbone and IoT Smart Things information packets come to the Cloud Computing Data Center through this connection.

The OneWeb System Architecture includes two groups of satellite beams: the User Beams providing connection with the User Terminal and the GateWay Beams providing connection with the GateWay Earth Station. The interface of the OneWeb User Terminal to the LAN or WiFi serves as the interface of the OneWeb system and therefore the borderline of the OneWeb system to the IoT Smart Things. The OneWeb User Terminal transmits IoT information bursts of the IoT Smart Things connected to it in the general flow through the Up-Link to the OneWeb satellite, which relays the received User Beams information flow to the Gate Way beam. The OneWeb Gate Way Earth Station is connected to the Internet Backbone. The Gate Way Earth Station receives information bursts of the IoT Smart Things in the general flow, extracts them and provides routing over the Internet Network to the Cloud Computing Data Center.

Currently, GEO communication Regular Satellites and High Throughput Satellites (HTS) are capable to provide the IoT information data transmission services. Both types of GEO satellites can be equipped with a payload with Transparent Transponders or Regenerative Transponders [4]. **Figure 2** shows an example of IoT data transmission using the HTS. Allocation of the separate User Beams and

Gate Way Beams is the feature of the architecture of the HTS geostationary satellite communication systems [8]. The interface of the VSAT Terminal to the LAN or WiFi serves as an external interface of the geostationary satellite communication system to the IoT Smart Things. The VSAT Terminal multiplexes IoT information bursts into a common flow and transmits it over the Up-Link. The HTS satellite transfers the received flow from the VSAT User Terminal to the Gate Way Beam. The GateWay, or its analogue - the HUB of VSAT Network in case of geostationary Regular Satellites utilization, is connected to the Internet Backbone, through which the IoT information bursts get to the Cloud Computing Data Center. To improve the efficiency of cloud services provided with the use of satellite telecommunication systems, the Microsoft Company together with Azure Company started the project implementation on the Cloud Storage Data Centers location in close proximity with satellite teleports [9].

As can be seen from the above presented structure, Satellite Communication Systems take place of data transmission channels between IoT Smart Things and Cloud Computing Data Centers in IoT Systems. The Satellite Communication Systems have to provide two-way transmission of the entire IoT Data Traffic in the Cloud Architecture IoT System, that significantly increases the communication load for channels and systems.

3. Perspective development directions of the IoT space infrastructure

The high communication channels load with two-way traffic generated by the IoT system is not the only disadvantage of the conventional IoT cloud architecture, but it is also the delay caused by the IoT information bursts transmission over the data channel through routers and other network equipment. Besides, the propagation time of the radio signal in radio networks and of the light wave in fiber-optic communication systems are of a significant impact. Delay has a particular impact on the IoT Delay Sensitive Service [10].

The solution to the problem is utilization of Fog and Edge Computing [3] in the IoT System Architecture. In this case, some of computations related to the IoT information processing is performed at the intermediate layers of the IoT system hierarchical structure. For this, the corresponding computing capacity is located at intermediate layers. Computing capacity locates nearer to the IoT Smart Things: Sensors and Actuators. As a result, the IoT information processing time is reduced, the IoT system response time to external impact is reduced, and the communication channel load is reduced. Only the results of IoT data processing at the lower layers are being transferred to the higher layers of the hierarchical system, at the same time the value of the transmitted information increases.

3.1 Implementation of edge and fog computing in satellite systems

Satellite communication systems are flexible enough to be adapted for implementation of Fog and Edge Computing. **Figure 3** shows the IoT Satellite System Model constructed with implementation of Fog and Edge Computing Architecture.

Edge Computing is a Distributed Computing Model when computation takes place near location where data is collected and analyzed, rather than on a Centralized Server or in the Cloud [11]. As shown earlier, in most cases, the User Terminal or VSAT Terminal Interface acts as the satellite communications system/network boundary to the local area network or to the short-range radio network, for example, Wi-Fi, ZigBee. The User Terminal or VSAT Terminal is located in the immediate vicinity to the location of the IoT Smart Things - Sensors and Actuators. Implementation of Edge Computing in satellite telecommunication systems can

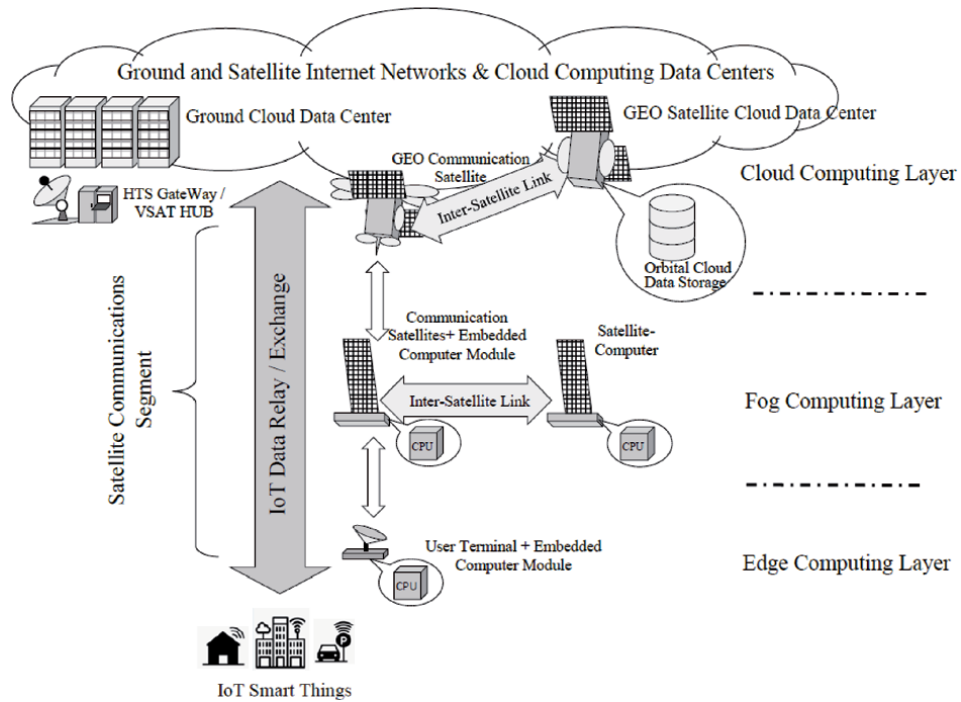


Figure 3. Hierarchical model of the internet of things satellite system architecture with implementation of fog and edge computing.

be ensured by supplementing of the User Terminal or VSAT Terminal Modem with an additional Computing Module or Single-Board Computer. Structurally, a User Terminal or VSAT Terminal is a board with modem chips installed on it. Through modernization, such a design can be supplemented with a Single-Board Computer, which will provide the implementation of Edge Computing. An alternative option is to connect a Single-Board Computer to an Ethernet-type Local Area Network with a Wi-Fi router being connected to it as well as other equipment of radio access technology for short-range IoT Smart Things. This added Computing Capacity will support the IoT Smart Things computing needs within the coverage of a short-range radio access network. In this case, only the results information about the IoT local information processing will be transmitted via a satellite communication channel.

Edge Computing is a Distributed Computing Model when computation takes place near location where data is collected and analyzed, rather than on a Centralized Server or in the Cloud [11]. As shown earlier, in most cases, the User Terminal or VSAT Terminal Interface acts as the satellite communications system/network boundary to the local area network or to the short-range radio network, for example, Wi-Fi, ZigBee. The User Terminal or VSAT Terminal is located in the immediate vicinity to the location of the IoT Smart Things - Sensors and Actuators. Implementation of Edge Computing in satellite telecommunication systems can be ensured by supplementing of the User Terminal or VSAT Terminal Modem with an additional Computing Module or Single-Board Computer. Structurally, a User Terminal or VSAT Terminal is a board with modem chips installed on it. Through modernization, such a design can be supplemented with a Single-Board Computer, which will provide the implementation of Edge Computing. An alternative option is to connect a Single-Board Computer to an Ethernet-type Local Area Network with a Wi-Fi router being connected to it as well as other equipment of radio access technology for short-range IoT Smart Things. This added Computing Capacity will

support the IoT Smart Things computing needs within the coverage of a short-range radio access network. In this case, only the results information about the IoT local information processing will be transmitted via a satellite communication channel.

Fog Computing is implemented at intermediate layers of the IoT System Hierarchical Model [12]. A Communication Satellite or a Satellite Constellation is the intermediate layer of the Hierarchical Structure of the Internet of Things System with a Satellite Communications Segment. It includes both GEO Satellites and LEO or MEO Satellite Constellations. The implementation of Fog Computing in the satellite segment of IoT Systems is possible by supplementing the orbital segment with Computing Capacity for the Fog computing implementation. In [13], the Fog computing implementation method was proposed by supplementing Micro-Constellations with separate Satellites-Computers. Considering the fact that the modernization of satellite-repeaters equipment is possible only at the stage of their manufacturing, the implementation of Fog computing in the orbital segment of the IoT Satellite Systems will take a longer period of time. This time period includes the project development of a modernized satellite, its ground tests, expectation time for an Orbital Life Time completion of the already launched satellites and a queuing time for new satellite launch.

Supplementing the Orbital Segment of Satellite Communications Systems with Computing Capacity will allow the implementation of Fog computing for processing of the IoT Information accepted from IoT Smart Things located in the service area of the Satellite. As a result, the efficiency of information processing will increase, and the Delay Time will be reduced. The IoT Information Traffic will load only the section “User Terminal - Satellite Payload” of the Satellite Communication Channel. In the direction “Satellite Payload – Gate Way/VSAT-network HUB” the result of IoT Information processing and summarizing will be transmitted only, that will significantly reduce the amount of information transmitted and increase its value.

Modern Satellite Communications Technologies as well as design and production technologies of Spacecraft for various purposes significantly expand the capabilities of Satellite Communication Systems in terms of Cloud computing implementation, which are at the highest hierarchical layer of the IoT System Architecture. Along with the traditional solution of IoT Information Transfer support to the Cloud Computing Data Center, with utilization of the GateWays or VSAT-network HUB with the Internet backbone connection, an alternative solution is possible – the special Spacecraft-Satellite Cloud Data Centers development and launching them to GEO. Currently the Space Belt project is underway already [14]. However, this project implies the use of Satellites – Data Center (or Cloud Data Storage) located in LEO. Access to Satellites Data Centers implies to be carried out through a GEO Satellite-Repeater.

An alternative solution is the development and launch of GEO Satellite, with a Cloud Data Center Module as a Payload. These Satellites will be accessed via GEO Satellite-Repeaters according with Inter-Satellite Links. To increase data storage and computing operations liability, to increase cloud computing productivity, Satellite Cloud Computing Data Centers will be connected to ground-based Cloud Computing Data Centers provided with special high-speed secure radio links.

3.2 LEO system based on LoRaWAN protocol

A LEO Communication system built to ensure the IoT Data Transmission using a modified communication long-range LoRaWAN protocol can be adapted to implement Fog computing by upgrading the System Orbital Segment Architecture by LoRaWAN GateWay equipment and Computing Capacity (see **Figure 4**).

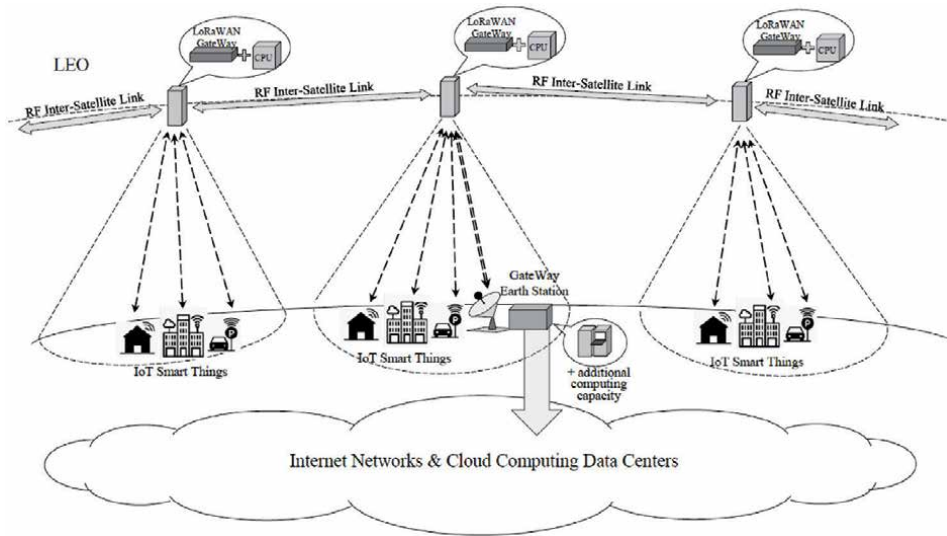


Figure 4. Architecture of the modified LEO internet of things satellite system based on the LoRaWAN protocol.

Considering the LoRaWAN Architecture, the CubeSat Payload can be supplemented with the following equipment:

- LoRaWAN Gateway, which provides data collection from Sensors – IoT Smart Things and relays the Sensor Data to the Central Server in the combined Multiplexed Stream;
- Computing Module (CPU), which will act as a LoRaWAN Server directly on-board CubeSat. The CPU installation as the part of Payload will allow the implementation of Fog computing in LEO Satellite System to provide processing of the IoT Smart Things Information Burst directly on the board of CubeSat. The CubeSat provides processing of IoT Smart Things Information located in the service area of the CubeSat.

The proposed changes could be implemented within several years. Since the CubeSats in-orbit life time is rather short and, as a rule, does not exceed 3–5 years, the proposed changes can be implemented via the launch of CubeSats next generation implying to maintain the operation of the orbital constellation of the system.

To ensure the interaction of the orbital and ground segments of the IoT LEO Communication System, constructed with utilization of the CubeSat type of spacecraft and providing IoT services based on the modified LoRaWAN algorithm, it is advisable that the System Orbital Segment/Satellite Constellation will be connected with the ground Internet network through GateWay Earth Station which should be added to the Ground Infrastructure of the IoT Satellite System. The main task of the Gate Way Earth Station is to receive the IoT data combined Multiplexed Stream, i.e. the information on the results from processing of the IoT Sensor information bursts in the Fog computing layer of IoT System, and transferring the received data flow to the Cloud Computing Data Center. The Gate Way Earth Station in a LEO Communications Satellite System provides connection with several Satellites simultaneously. Therefore, it is advisable to add the equipment of the GateWay Earth Station with a computer, which will equip the Earth Station with Computing Capacity. The Computing Capacity implemented into the Earth Station will make

it possible to realize Fog computing for generalizing of the IoT information from several CubeSats situated in the GateWay Earth Station radio visibility zone. In the Hierarchical Architecture of the Internet of Things, such a processing of generalized information corresponds to the Fog computing layer.

To improve the efficiency of LEO Satellite Communication Systems developed with utilization of small- and ultra-small satellites, including CubeSats, the Inter-Satellite Links (ISL) are included in the system architecture [15]. The ISL utilization between CubeSats in LEO Communication System allows transmitting the generalized IoT Data Flow to a neighboring CubeSat for its further relaying to the GateWay Earth Station and thus to expand the service area of the GateWay Earth Station and to reduce their number.

3.3 OneWeb LEO broadband access satellite system

The Transparent Payload utilization in satellites is a feature of the OneWeb LEO Broadband Access Satellite System Architecture. The OneWeb Satellite Payload provides transfer of the User Beam frequency band to the Gate Way Beam frequency band [16]. There is no information processing in the payload. The OneWeb System architecture does not imply Inter-Satellite Links between Satellites.

For adaptation of the OneWeb System to the peculiarities of the Internet of Things and implementation of Edge and Fog computing, the capabilities of User Terminals and Gateway can be used (see **Figure 5**). In the OneWeb System the network boundary from the End User side is the interface to the Ethernet LAN or to the Wi-Fi radio access network. The equipment of the User Terminal could be supplemented with a computing module in the form of a separate Processor Unit or a Single-Board Computer. An alternative option could be the connection of the Single-Board Computer to the Ethernet LAN. This Computing Capacity, located at the User Terminal layout in the immediate vicinity of the IoT Smart Things (Sensors and Actuators), is introduced into the System in order to implement Edge Computing. Supplementing the User Terminal with a computing facility will make possible primary processing of information packets from IoT Sensors and form control commands for Actuators at the User Terminal Layer. The Satellite Communication Channel will transmit generalized information formed on the

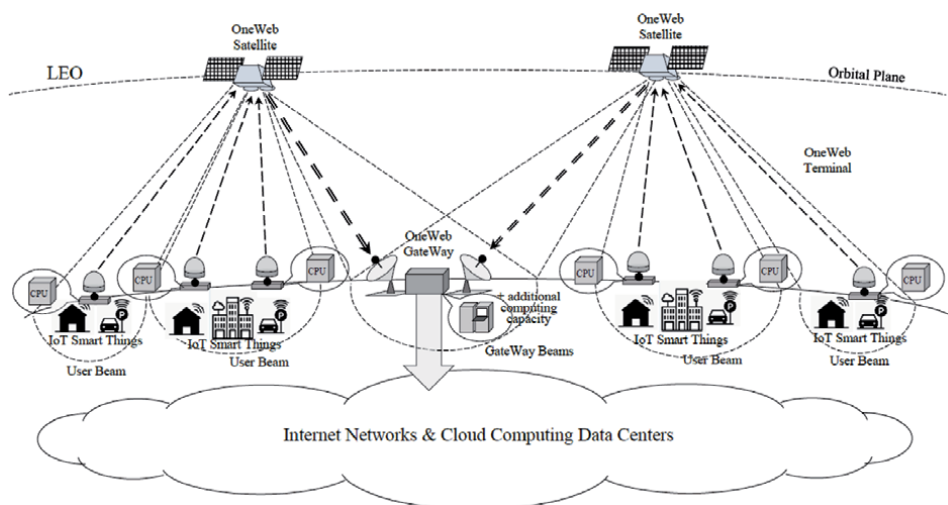


Figure 5.
Adaptation of the OneWeb satellite system for IoT systems.

results of processing information from the local group of IoT Smart Things located in the coverage area of the short-range radio access technology.

Fog computing can be implemented at the GateWay Layer. For this, the GateWay Earth Station Equipment have to be supplemented with a Computing Module - a Multiprocessor Computer Group installed in additional Rack (see **Figure 5**). This Computation equipment will provide the IoT Data processing received from the service area of the OneWeb Satellite, or from Satellites situated in the service area of the GateWay Earth Station. Control commands for special IoT Smart Things and groups of Smart Things – Actuators, will be transmitted from the GateWay Earth Station via Satellite to the User Terminal. Generalized information on the results of the IoT data processing and about the decisions and generated control commands will be transmitted via the Internet to the Cloud Computing Data Center, as to the highest Layer of the IoT System Hierarchical Architecture.

3.4 StarLink LEO broadband access satellite system

LEO Broadband Access Satellite System Starlink, like the OneWeb System, has a formed architecture focused on providing End Users with high-speed Internet Access Services. As it was above mentioned, the existing Starlink System Architecture allows only the IoT Services of a Cloud Architecture.

For adaptation of the StarLink System to the Internet of Things System peculiarities and implementation of Edge and Fog computing, methods similar to those proposed for the OneWeb system can be used, namely (see **Figure 6**):

- supplementing User Terminals with Single-Board Computers or connecting a Single-Board Computer to the WiFi Radio Network for implementation of Edge computing for the data processing of the IoT Smart Things located inside a short-range radio access network;
- supplementing of the equipment of the GateWay Earth Station with a separate multiprocessor computer group/rack for implementation of Fog computing for

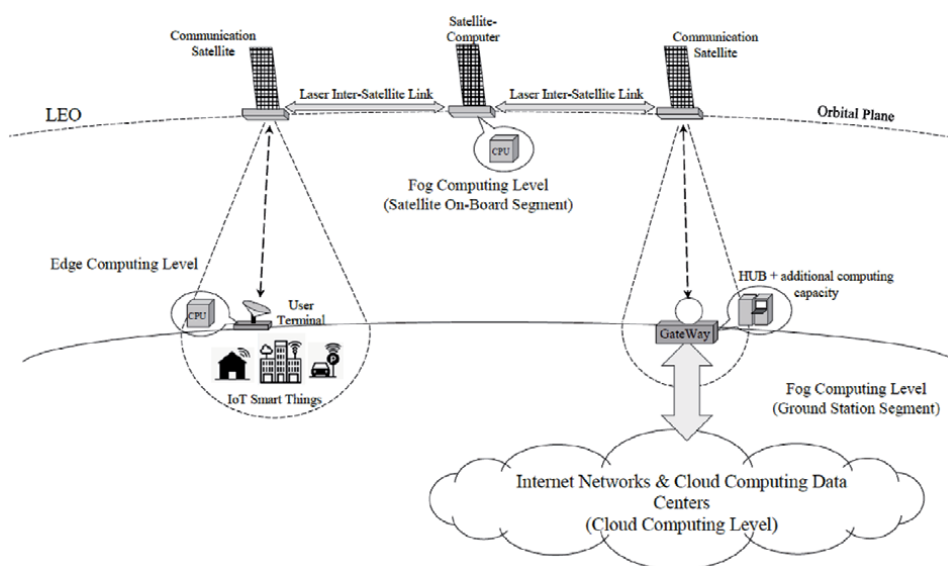


Figure 6. Adaptation of StarLink system for edge and fog computing IoT services.

the data processing for the IoT Smart Things located in the service area of all the StarLink satellites in the GateWay radio visibility zone.

At the same time, the Laser Inter-Satellite Links utilization in the StarLink System [7] makes it possible to consider the decision of supplementing the StarLink Constellation with Satellite-Computers (see **Figure 6**). Unlike StarLink Satellite-Repeaters, the Payload of the Satellite-Computer is a Computing Module - a Processor Unit and a long-term Memory Module. Like Satellite-Repeaters, the Satellite-Computer Payload comprises the Router in it. To ensure links with other satellites, the Satellite-Computer is equipped with Optical Heads for the Laser Inter-Satellite Links.

The purpose of the Satellite-Computer is to generate the Computational Capacity directly in LEO in the same Orbital Plane with the Satellite-Repeaters. In each Orbital Plane of the StarLink Constellation, several Satellite-Computers can be placed (see **Figure 7**). The IoT information will be transmitted from Satellite-Repeaters to a Satellite-Computer via Inter-Satellite Links for processing, actuator command generation and aggregation of generalized information. Placing Satellite-Computers in the Orbital Plane and retargeting optical transceivers/optical heads of Laser Inter-Satellite Links towards them will not destroy the integrity of the Orbital Plane Data Transmission Ring Network, as Satellite-Computer, like the Satellite-Receiver, is equipped with a Router that will distribute data streams assigned for further retransmission via the Ring Network of the Orbital Plane and will extract information assigned for processing in the Computing Module of the Satellite-Computer.

Supplementing of the StarLink Constellation with Satellite-Computers will make it possible to create Computing capacity directly in the orbit for the Fog computing implementation for the IoT Systems.



Figure 7.
Location of satellites-computers in one orbital plane of the StarLink constellation.

3.5 GEO high throughput satellite system. Orbital cloud data center

Geostationary Satellite Communication Systems are an important component of modern Satellite Communication Infrastructure. The growing demand for data transmission bandwidth and for provision of information services, primarily for the Internet Access, has become a driver for the HTS, a new class of GEO Satellites, to enter the market. The main advantage of these satellites is the information transmission low cost per one bit between two subscribers [8].

The architecture of GEO HTS Systems has its own characteristics, which were mentioned above. Another feature of HTS is the principle: one Transponder per one Spot Beam [17]. According to this principle, one Transponder provides amplification of signals via the entire frequency band, which can be 150 ÷ 250 MHz and more.

The architecture of GEO HTS Systems can be adapted to the peculiarities of the IoT Systems in several stages as follows (see **Figure 8**). At the first stage, it is possible to upgrade the system elements related to the Ground Communication Segment: User/VSAT Terminals and Gate Way Earth Stations (GateWay). Loading Computing Capacity on these elements will allow the implementation of Edge and Fog computing for the IoT Systems. Possible technical solutions for the Computing Capacity implementation on these elements are similar to the technical solutions discussed above for OneWeb and StarLink LEO Systems.

Currently available design and manufacture technologies for the GEO Satellites with a 15–20 years life time, and the experience accumulated in the construction and operation of Satellite Constellations, in-Orbit Satellites Interaction, makes it possible to consider the issue of creating perspective Orbital Cloud Data Storage, consisting of several Geostationary Satellites – GEO Satellite Cloud Data Centers (see **Figure 8**).

Orbital Cloud Data Storage cannot be considered the alternative to the Ground Cloud Data Processing and Storage Centers, since the Computing Capacity and Storage Capacity for Ground Cloud Data Centers are practically unlimited. Orbital

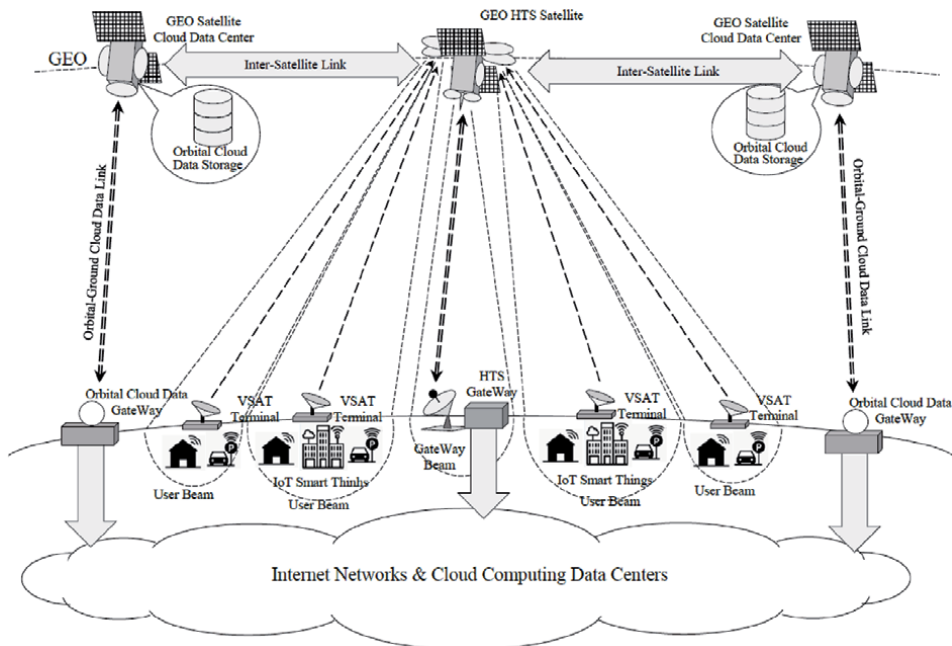


Figure 8. Adaptation of the geostationary high-throughput systems to the IoT systems peculiarities and interaction with an orbital cloud computing data center.

Cloud Data Storage is an augmentation to the Ground Cloud Infrastructure and is focused primarily on processing and storing data from IoT Satellite Systems. To improve the reliability of data storage and the Computing Capacity increase, if necessary, the Orbital Cloud Data Storage interacts with the Ground Cloud Centers infrastructure via RF data transmission channels specially dedicated.

GEO High-Throughput Satellites will provide access to GEO Satellites - Cloud Data Centers via Inter-Satellite Link, set up in the radio frequency or optical band. The possibility of the long-distance optical links utilization in space has been practically confirmed on the establishment of the Europe Data Relay System (EDRS), implemented by the European Space Agency order [18].

To route IoT Information to the Satellite - Cloud Data Center, the GEO HTS Satellite have to route IoT Traffic. Routing can be provided by the following method:

- with Regenerative Payload on board the Satellite through extracting IoT information from the Data Transport Stream transmitted by VSAT terminals and IoT Information routing towards GEO Satellite - Cloud Data Center. The Advanced Regenerative On-board Processing Satellite (AR-OBPS) technology can be used as a basic technology for this process [19];
- when separated frequency bands allocated for IoT information in the common frequency band of each user beam. The IoT Information frequency band will be switched in the Satellite Payload separately from the other frequency band and transmitted over the Inter-Satellite Link between satellites to GEO Satellite - Cloud Computing Data Center. The Intelsat Epic^{NG} Platform Digital Payload Technology [20] can be used as a basic technology for this process.

Orbital Cloud Data Storage can provide the IoT Data Processing for LEO IoT Systems (see **Figure 9**). In this case, LEO System based on the LoRaWAN protocol provides the implementation of Fog computing, as shown above, and the Orbital Cloud Data Storage provides the Cloud computing Layer (see **Figure 3**).

Interaction between LEO CubeSats and GEO Satellites – Cloud Computing Data Centers, is provided via LEO-GEO Inter-Satellite Link. To set up GEO-LEO Inter-Satellite Link the CubeSats from the LEO System could be equipped with Deployable Parabolic Dish Antennas [21]. LEO CubeSats should be designed to point Deployable Parabolic Dish Antenna towards GEO Satellite – Cloud Computing Data Center or towards GEO HTS, which in this case will be used as an IoT Data Repeater and Router.

Figure 10 shows the architecture of the Constellation of the combined LEO-GEO IoT Satellite System. CubeSats are located in LEO and provide the IoT Information/IoT Information Burst reception using modified LoRaWAN protocol directly from IoT Smart Things – Sensors and transmitting control information to IoT Smart Things – Actuators within CubeSat coverage zone. To simplify, in **Figure 10** only the Orbital Plane is shown. The LEO Component of the Constellation consists of several Orbital Planes, which number is determined according to the requirements for continuity of the Service, Power Capacity and the Life Time of IoT Devices - Sensors and Actuators, and other factors. CubeSats are equipped with Deployable Parabolic Dish Antennas of RF LEO-GEO Inter-Satellite link and provide parabolic antennas steering towards the GEO Satellite.

GEO Satellites - Cloud Computing Data Centers or GEO High-Throughput Satellites are located in the Geostationary Orbit and are equipped with GEO-GEO Inter-Satellite Link. GEO Satellite provides the reception of processed IoT Data from LEO CubeSats in the radio visibility zone. To provide a continuous radio interconnection with LEO CubeSats, three GEO Satellites - Cloud Computing Data

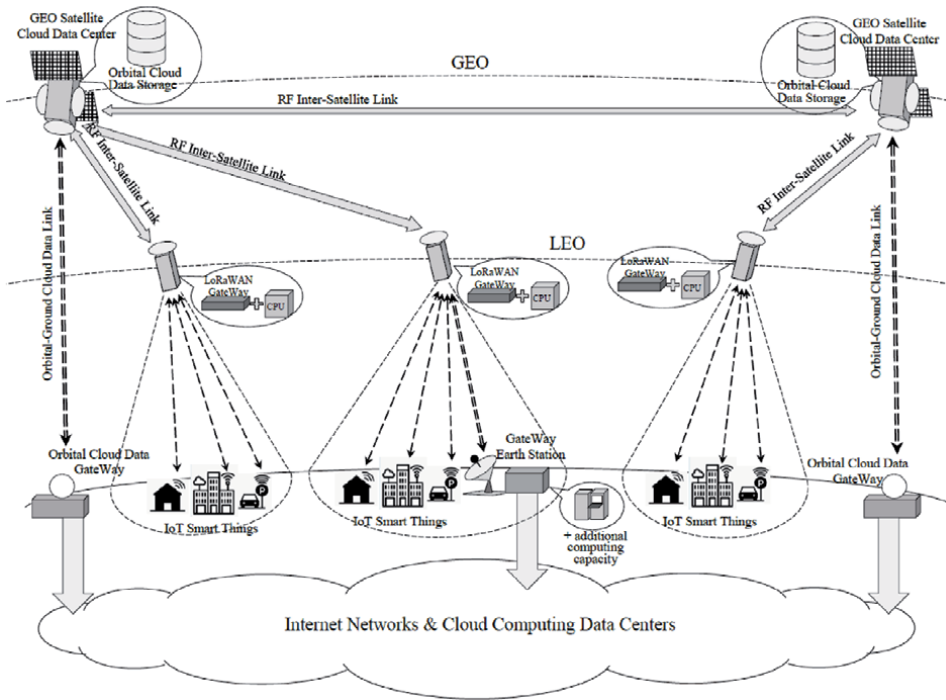


Figure 9. Interaction between the orbital cloud data storage and the LEO IoT satellite system based on the modified LoRaWAN protocol.

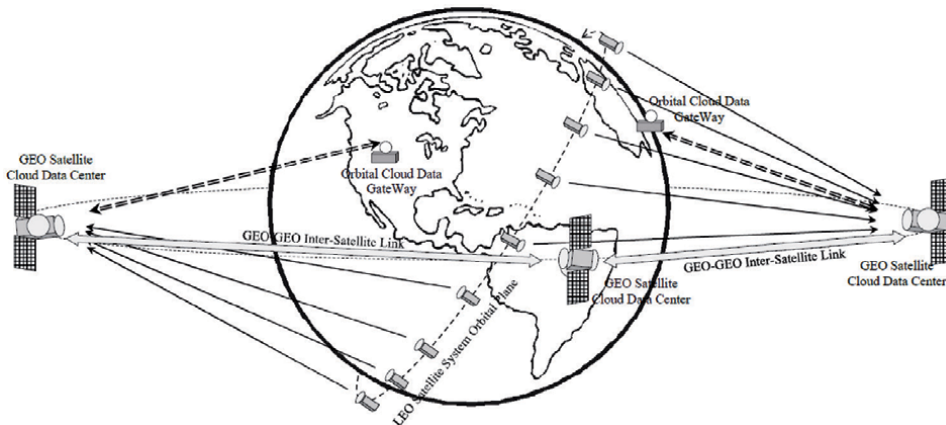


Figure 10. The combined GEO-LEO IoT satellite system constellation architecture.

Centers or High-Throughput Satellites is sufficient to be placed in GEO. The integrity of the Orbital and Ground Cloud Data Infrastructure is supported by GEO-GEO Inter-Satellite Links and GEO Satellite - Ground Cloud Computing Gateway Earth Station Links (see **Figure 10**).

Currently, Geostationary Orbit is uploaded enough with operating GEO Satellites of various missions and Satellites that have been taken out of service (inoperative). **Figure 11** shows a chart of the Geostationary Orbit upload by satellites under control [22].

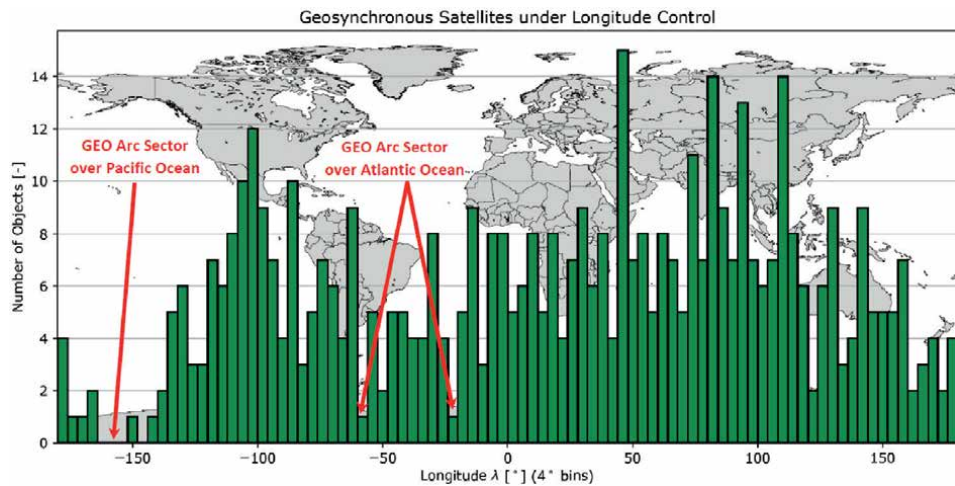


Figure 11.
 Geostationary orbit upload [22] and free orbital slots for operation of GEO satellite – Cloud data center

As can be seen from **Figure 11**, the most free GEO sectors are the ones located over the Pacific and Atlantic oceans: sectors $144^{\circ} \text{ W} \div 164^{\circ} \text{ W}$; $56^{\circ} \text{ W} \div 60^{\circ} \text{ W}$; $20^{\circ} \text{ W} \div 24^{\circ} \text{ W}$. Considering the fact that GEO-GEO Inter-Satellite Link are used to provide access to GEO Satellite - Cloud Data Centers, the GEO Satellites from the Orbital Cloud Data Storage can be placed in these GEO sectors, which are not of interest for the satellite communication services provision to end users on the Earth surface.

4. Conclusions

1. Modern Satellite Communication Systems provide data transfer of IoT Systems, based mainly on Cloud Technology. The disadvantage of the IoT Systems Cloud Architecture is the necessity to transfer the entire amount of information from IoT Smart Things to Cloud Computing Data Centers and vice versa, that leads to Satellite Communication Systems inefficient load.
2. Satellite Communication Systems can be adapted to the peculiarities of Data Transport Streams in the IoT Systems, which use Fog and Edge computing Technologies to increase their efficiency. The Satellite Communication Systems adaptation to the peculiarities of IoT Fog and Edge computing is being carried out by placing computers of various capacities as the part of User/VSAT Terminals, Satellite Payloads and Gate Ways Earth Stations or VSAT networks HUB Stations. Such an arrangement of Computing Capacities and distribution of computations allows maintaining the strong IoT System Hierarchical Architecture, reducing the processing time and the transferred data volume, and increasing the value of information transmitted to the Cloud Computing Data Center.
3. The ways for the Satellite Communication Systems transition from the IoT Systems Cloud Architecture to the Multi-Layer Architecture with the Edge and Fog computing utilization are proposed. The implementation variants of Fog computing in LEO Systems are considered: Satellite Constellation of CubeSats with the modernized LoRaWAN protocol - the CubeSat Payload update and CubeSats replacement during the Satellite Constellation planned

update; OneWeb system - the End-User Terminals and GateWay Earth Station equipment update; StarLink System – the User and Gate Way terminals update, the Constellation supplementation with Satellites - Computers.

4. In GEO High-Throughput Systems, the implementation of Edge and Fog computing is possible in two stages. At the first stage, the transition to Fog and Edge computing is possible by the User terminals and GateWay Earth Station modernization to supplement their structure with Computing Modules. At the second stage, during the planned replacement of a GEO High-Throughput Satellite, its Payload can be equipped with additional equipment for the IoT Systems Traffic allocation, traffic processing and carrying out the necessary calculations for the Fog computing system implementation in the Space Segment Structure.
5. To increase the efficiency of processing, storage and the IoT Systems access to Cloud Services, it is reasonable to create a Cloud Services Space Segment - an Orbital Cloud Data Storage, consisting of several GEO Satellites - Cloud Computing Data Centers which are connected via Inter-Satellite Links. The Orbital Cloud Data Storage can be accessed through upgraded GEO High-Throughput Satellites and through LEO CubeSats equipped with a LEO-GEO Inter-Satellite Link.

List of Acronyms

AR-OBPS	Advanced Regenerative On-board Processing Satellite
EDRS	Europe Data Relay System
GEO	Geostationary Orbit
HTS	High Throughput Satellites
IoT	Internet of Things
ISL	Inter-Satellite Link
LAN	Local Area Network
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
VSAT	Very Small Aperture Terminal

Author details

Mikhail Ilchenko¹, Teodor Narytnyk¹, Vladimir Prisyazhny², Segii Kapshtyk^{2*}
and Sergey Matvienko³

1 National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kiev, Ukraine

2 National Space Facility Control and Test Center, Kiev, Ukraine

3 Scientific and Production Complex “Kurs”, Kiev, Ukraine

*Address all correspondence to: sergii.kapshtyk@gmail.com

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Global IoT market to grow to \$1.5trn annual revenue by 2030. Available from: <https://www.iot-now.com/2020/05/20/102937-global-iot-market-to-grow-to-1-5trn-annual-revenue-by-2030/>
- [2] 5G Sub 6 GHz Technologies and Trends. Available from: <https://www.microwavejournal.com/articles/34295-g-sub-6-ghz-technologies-and-trends>
- [3] Chernyshev A.I., Demidenko I. O.Voruev., A.V., Mikhnevich S.Yu. Programmiruemo upravlenie dostupom k seti s adaptivnoy nastrojkoj fizicheskikh interfejsov / Programmable network access control with adaptive physical interface configuration. In: Izvestiya Gomelskogo gosudarstvennogo universiteta imeny F. Skoriny; № 6 (111), 2018.p.p.55-62.
- [4] Gerard Maral, Michel Bousquet. Satellite communications systems - 5th ed. © 2009 John Wiley & Sons Ltd.
- [5] Satellite 2020 – Lacuna Space explains LoRaWAN satellite success. Available from: <https://www.spaceitbridge.com/satellite-2020-lacuna-space-explains-lorawan-satellite-success.htm>
- [6] STARLINK ENCYCLOPEDIA. Available from: <https://www.comnews.ru/content/209438/2020-10-07/2020-w41/enciklopediya-starlink>
- [7] Mark Handley, Delay is Not an Option: Low Latency Routing in Space. University College London. Available from: https://www.researchgate.net/publication/328891593_Delay_is_Not_an_Option_Low_Latency_Routing_in_Space
- [8] SATELLITE COMMUNICATIONS & BROADCASTING MARKETS SURVEY. FORECASTS TO 2025. 23rd Edition. September 2016. Copyright © 2016 Euroconsult
- [9] Azure Space partners bring deep expertise to new venture. Available from: <https://news.microsoft.com/transform/azure-space-partners-bring-deep-expertise-to-new-venture/>
- [10] Ye Chen, Wei Liu, Tian Wang, Qingyong Deng, Anfeng Liu, Houbing Song. An adaptive retransmit mechanism for delay differentiated services in industrial WSNs In: EURASIP Journal on Wireless Communications and Networking (2019) 2019. Article number: 258
- [11] Edge Computing for Dummies®, Stratus Special Edition. John Wiley & Sons, Inc. Copyright © 2020 by John Wiley & Sons, Inc., Hoboken, New Jersey
- [12] Nitinder Mohan, Jussi Kangasharju. Edge-Fog Cloud: A Distributed Cloud for Internet of Things Computations. In: CIoT'16. Available from: <https://ieeexplore.ieee.org/document/7872914>
- [13] Mikhail Ilchenko, Teodor Narytnyk, Vladimir Prisyazhny, Segii Kapshtyk, and Sergey Matvienko. Low-Earth Orbital Internet of Things Satellite System on the Basis of Distributed Satellite Architecture. In: Advances in Computer, Communication and Computational Sciences. Proceedings of IC4S 2019. Advances in Intelligent Systems and Computing, Volume 1158. © Springer Nature Singapore Pte Ltd. 2021pp. 301-314
- [14] Caleb Henry. Cloud Constellation selects LeoStella to build 10 data-storage satellites. Dated May 2, 2019. Available from: <https://spacenews.com/cloud-constellation-selects-leostella-to-build-10-data-storage-satellites/>
- [15] Scott C. Burleigh, Tomaso De Cola, Simone Morosi, Sara Jayousi, Ernestina Cianca, and Christian Fuchs. From Connectivity to Advanced Internet

Services: A Comprehensive Review of Small Satellites Communications and Networks. In: Hindawi Wireless Communications and Mobile Computing, Volume 2019. Article ID 6243505. Available from: <https://doi.org/10.1155/2019/6243505>

[16] ONEWEB NON-GEOSTATIONARY SATELLITE SYSTEM. ATTACHMENT A. Technical Information to Supplement Schedule S.

[17] Daniel Minoli. INNOVATIONS IN SATELLITE COMMUNICATIONS AND SATELLITE TECHNOLOGY. The Industry Implications of DVB-S2X, High Throughput Satellites, Ultra HD, M2M, and IP. Copyright © 2015 by John Wiley & Sons, Inc.

[18] EDRS (European Data Relay Satellite) Constellation. SpaceDataHighway. Available from: <https://directory.eoportal.org/web/eoportal/satellite-missions/e/edrs>

[19] John Nguyen. Overview of Existing and Future Advanced Satellite Systems. Available from: <https://www.intechopen.com/online-first/overview-of-existing-and-future-advanced-satellite-systems>

[20] Operating in an Epic^{NG} Environment. Intelsat, 2014.

[21] Volkan Akan, Erdem Yazgan. Antennas for Space Applications: A Review. Available from: <https://www.intechopen.com/books/advanced-radio-frequency-antennas-for-modern-communication-and-medical-systems/antennas-for-space-applications-a-review>

[22] CLASSIFICATION OF GEOSYNCHRONOUS OBJECTS. Date 28 May 2018 Issue 20 Rev.0. European Space Agency. European Space Operations Centre

Internet of Things Security and Privacy

Ahmad J. Showail

Abstract

The Internet of Things is becoming more and more popular with time. The extremely low cost of sensors is putting the growth of the Internet of Things on steroids. Many industries such as healthcare, construction, agriculture, and transportation are increasingly leveraging this technology. However, security and privacy are two big concerns when it comes to the future of the Internet of Things. Since most of these “things” that are connected to the Internet are simple devices with limited hardware capabilities, it is nearly impossible to harden them via traditional resource-heavy defenses. In this chapter, we discuss the importance of securing the Internet of Things networks, layout the challenges of the Internet of Things security, and briefly discuss potential solutions in the literature.

Keywords: cyber security, Privacy, Internet of Things

1. Introduction

Since the invention of the first ever network in 1972, computers are being connected using various topologies. Ranging from the traditional pair of wires found in industrial plants that connect sensors and actuators to the process control system, to state-of-the-art IPv6-enabled wearable sensors for medical monitoring, the idea is the same. You have a bunch of sensors that should talk to each other securely, whether on the Ethernet, Bluetooth, Zigbee, or a basic 4–20 milliamp electric circuit. We can safely say that IoT was born after the wedding of Information Technology (IT) and Operation Technology (OT). In fact, Fieldbus technology [1], which is one of the variants that is widely adopted in the industry, is a direct result of the advancements in OT that is trying to make the devices in the field ‘smarter’. However, the challenge is how to get these variants to talk to each other, which requires a common ground of communication, such as the Internet Protocol (IP).

We can think of two types of Internet of Things (IoT) implementation, namely greenfield and brownfield [2]. Almost all the implementations in the fields that have no legacy using networked systems, such as health, agriculture, and transportation, are considered as greenfield IoT implementation. On the other hand, a brownfield implementation is the one trying to introduce internet devices in conjunction with traditional networked infrastructure, such as Fieldbus technology in the process automation industry. In both cases, a defined framework for communication is urgently needed to enhance system security and minimize the risk.

On the 21st of Oct 2016, a massive Distributed Denial of Service (DDOS) attack resulted in putting down the web services of famous companies such as Twitter, Amazon, Netflix, Airbnb, and GitHub, among others [3, 4]. The malware name was Mirai and it targets the DNS service provider called Dyn [5]. In the Japanese language, Mirai means “Future” [6]. In reality, what Mirai did was simply switching Linux-based devices into digital weapons by exploiting the vulnerabilities of IoT devices, like factory default settings. In this specific attack, IP cameras and Digital Video Recorders (DVRs) were used to launch the attack. What makes this possible is the fact that many manufacturers of IoT devices leave the passwords hardcoded in the firmware, allowing hackers to easily connect to them using Telnet or Secure Shell (SSH).

In 2010, a specialized piece of code was developed to target nuclear plants. This malware was called “Stuxnet” [7]. Although the fact that most of the nuclear research centers employ the well-known air gap security mechanism, a poisoned Universal Serial Bus (USB) flash drive was used to infect the Programmable Logic Controller (PLC) responsible for the uranium enrichment centrifuges. As a result, the centrifuges suffered from physical damage due to faster than usual spins for extended periods, and abnormal acceleration and deceleration rates. Iran was not the only country that was affected by the attack. Other countries, such as India and Indonesia, were affected as well.

2. Building blocks of IoT security

Most of the network protocols are designed without having security in mind. Hence, network security is often the aftermath. Also, the heterogeneity of network components might be the gateway for predators to compromise the network. One way to solve this issue is to divide the devices in the network into two groups, trusted and untrusted devices. The former group is equipped with “root-of-trust” that could be as simple as an attestation key supplied by the manufacturer [8]. These trusted devices use secure communication mechanisms, namely secure key storage and cryptographic operations. Attestation protocols are used to assess whether untrusted devices are secure enough to join the trustworthiness group. The borderline between the trusted-devices group and the untrusted ones is usually invisible, as it is very difficult to classify these groups based on the type, manufacturer or even use. Moreover, the attestation process is usually dynamic and might change over time or in response to attacks on the network.

Root-of-trust is nothing but the set of trusted functionalities in the device that are assumed to be trustworthy and could never be compromised [9]. For example, the secure booting functionality of the device is a root-of-trust. Another example is the attestation functionality, which proves the validity of claims using cryptographic mechanisms. In fact, an IoT device might have multiple roots-of trust.

The ecosystem to establish trustworthiness in any IoT framework is composed of several building blocks, as shown in **Figure 1**. The first and foremost is the Trusted Execution Environment (TEE), which is responsible for executing trusted application codes and minimizing security risks. It is also responsible for isolating the process execution from other processes running on the same hardware. The second component is the secure communication channel, which is responsible for preserving the confidentiality and integrity of the data flowing between devices in the network using standard encryption mechanisms. The third component is the authentication process, including both the keys themselves, whether symmetric or asymmetric, and the actual key distribution and authentication protocol. The fourth component is the attestation process, involving the attestation key that is provided by the manufacturer, as well as the verification logic. After that, we must ensure that the devices are capable of securely storing all the keys and data gathered from the sensors. Finally, there is the ability to gather all relevant contextual information, such as location and time.

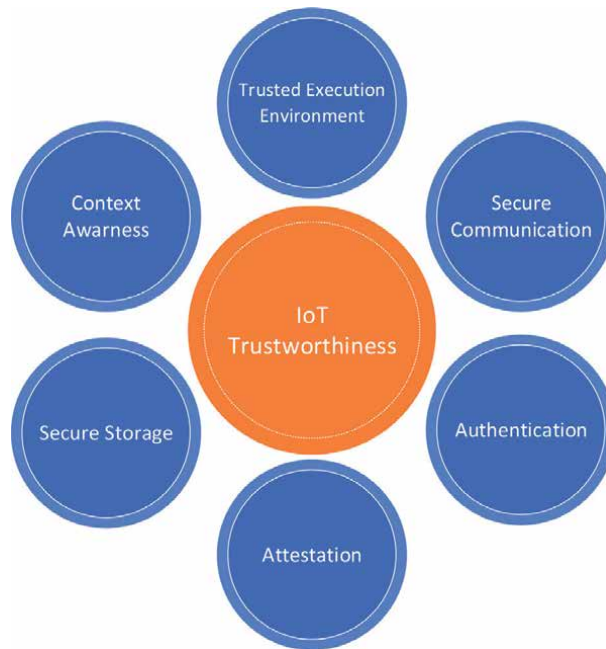


Figure 1.
Building blocks of IoT trustworthiness.

3. IoT device lifecycle

When discussing IoT security, we need first to understand the various stages in which the IoT device is going into overtime. **Figure 2** shows the typical lifecycle of an IoT device. It starts with the development of the software part of the IoT device using the Software Development Kit (SDK) or the Application Programming Interface (API) to hide the complexity. Then, tools are used for building the physical components of the device itself. We cannot stress enough the importance of the right configurations in the lifecycle of the IoT device. In this stage, various parameters are set in multiple components, such as the Central Processing Unit (CPU), the System on Chip (SoC), and the Operating System (OS). After that, it is the time for field deployment and making sure that the connection is established properly. Then, frequent updates are installed to protect the device. Finally, retirement is the reality that outdated devices must face.

4. End-to-end IoT security

The end-to-end concept is important when talking about the security of communication networks. **Figure 3** shows the main components involved in the end-to-end security journey of IoT device communication. Typically, the IoT device will encrypt the data gathered from sensors and send it to the gateway. Sometimes, it might store this data locally after encrypting it. The gateway will decrypt the data and run some analytics, and then encrypt it again to share it with the cloud. The cloud instance will decrypt the data one more time once received and run some analytics before encrypting it again, so it can be stored in the Database (DB).

To increase the portability of nodes, IoT frameworks have been proposed. The IoT framework is a great way to hide the complexity of network topology and type. However, an IoT framework must be designed to support end-to-end secure

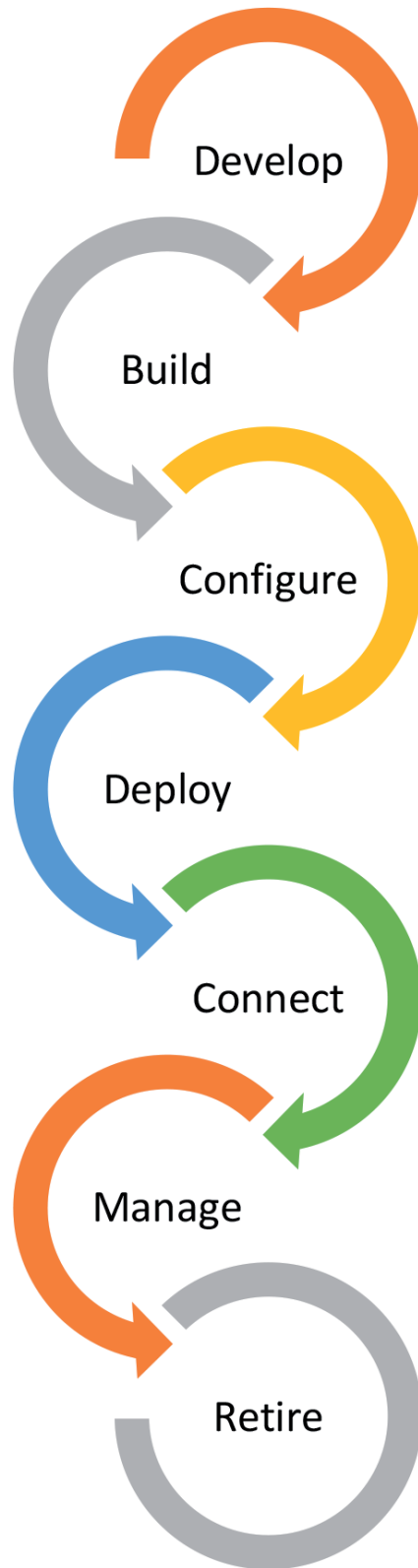


Figure 2.
IoT device typical lifecycle.

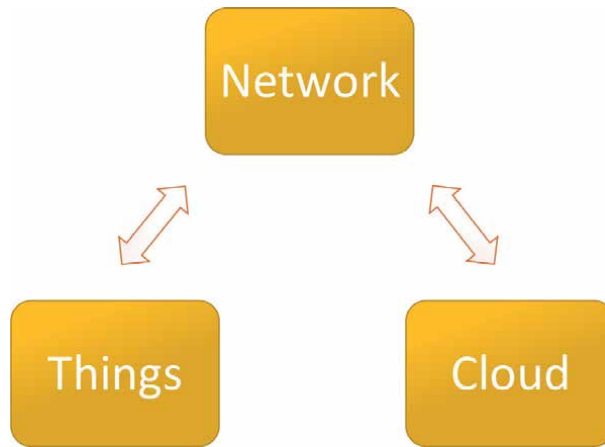


Figure 3.
 Components of end-to-end security in IoT.

IoT framework	Security approach
Open Connectivity Foundation (OCF)	Tackles the security using three strategies: 1. Access control 2. Message encryption. 3. Device lifecycle management. Issue: (no security interoperability with other frameworks such as AllJoyn or UPnP)
AllSeen Alliance/AllJoyn	End-to-end security in the application layer using leaf nodes.
Universal Plug and Play (UPnP)	Security was not in the initial design and it was added later as an optional service through the IoT management and control architecture.
Lightweight Machine 2 Machine (LWM2M)	It achieves security using a secure message exchange with the Datagram Transport Layer Security (DTLS) and an access control list using the bootstrap server.
One Machine to Machine (OneM2M)	By design, it has the capability of performing authorization, access control, data protection as well as privacy preservation.
Open Platform Communications-Unified Architecture (OPC-UA)	OPC-UA is designed with security in mind. Distribute security functions over two layers, namely: the session layer and the secure channel layer. The former is the one responsible for authentication and access control, whereas the latter is taking care of message encryption, using Transport Layer Security (TLS) and HyperText Transfer Protocol Secure (HTTPS).
Data Distribution Service (DDS)	Security is achieved through three techniques: 1. Message security enveloping 2. Security tokens 3. Security plugin modules to add on services, such as authentication, access control and encryption.

Table 1.
 Approaches of various IoT platforms.

communication between nodes, whether from an authentication, privacy, or confidentiality point-of-view. In fact, frameworks vary significantly when it comes to the implementation of this notion of end-to-end security [10, 11]. Some of these discrepancies are highlighted in **Table 1**.

The IoT framework is composed of three different layers: the data object layer, the node interaction layer, and the platform abstraction layer, also known as the connectivity and hardware abstraction layer. These three layers are shown in **Figure 4**. The data object layer is responsible for physical and logical node-to-device mapping. Also, it is the one responsible for managing the node Access Control List (ACL). The second layer is responsible for inter-node communication. The end-point security context must be achieved in this layer. Finally, the platform layer could be further

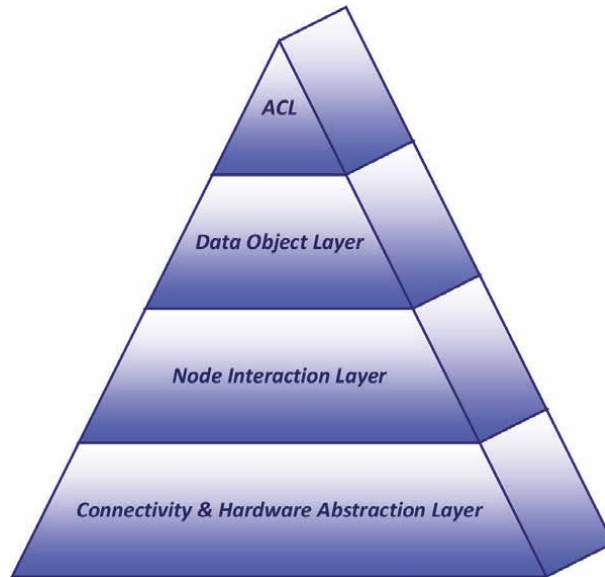


Figure 4.
Layers of IoT framework.

partitioned into three sub-layers, namely: network, sensor, and actuator, as well as security layer. Basically, an IoT network uses the same Internet layering model. This allows the IP to run on top of legacy industrial IoT protocols, like Fieldbus. The only difference is replacing the application layer with the IoT framework layer.

5. Securing IoT nodes

Nodes in any network must be able to do three basic tasks:

1. Neighborhood discovery
2. Authentication
3. Secure communication

To achieve IoT security effectively, we should focus on protecting the device, user identity, and data. We should manage the security at runtime as well. These points are illustrated in **Figure 5**.

The IoT Ecosystem is composed of the device, network, framework, and system management. By system management, we mean the procedure to maintain, replace and retire services. Also, it includes the procedures to update the firmware and apply security updates. In fact, requirements for system management vary a lot with different implementations. Obviously, brownfield and greenfield implementations have different system management requirements.

Let us talk about securing the IoT device itself. IoT devices are often resource constrained when it comes to memory space (storage), computation power, or even battery life. Therefore, the selection of which cryptographic algorithm to use is crucial. Although it is associated with high security impersonation risk, symmetric key cryptography is considered the most suitable type for IoT use. This is because it requires a small memory size and literally no hardware acceleration. Moreover, it is considered post-quantum safe given that the key size is increased from 128 to 256 bits [12].

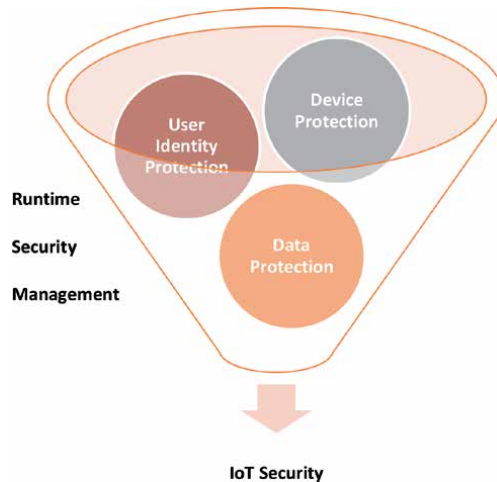


Figure 5.
Technologies involved in securing IoT devices.

Device Identity Composition Engine (DICE) [13, 14] is a secure IoT initiative proposed by Trusted Computing Group (TCG). It tackles the issues of secure booting and attestation without the need of a dedicated co-processor. DICE uses a hardware-based unique number generated among the device's boot called Unique Device Secret (UDS). The device identifier is nothing but the result of hashing the UDS with the device firmware. Thus, any modification to the firmware will result in a different device identifier, which will mark the device as 'untrusted' in the network.

6. Hardware and firmware security in IoT

Figure 6 shows the number of hardware and firmware security vulnerabilities in the past 20 years [15]. It is very clear that the number is on the rise and it should get the community's attention.

To achieve hardware-based security in IoT networks, we need to make sure that four aspects are taken care of, which are:

1. Device Identity
2. Boot Protection
3. Storage Protection
4. Runtime Protection

Intel has utilized available hardware-based technologies to achieve secure IoT networks, as illustrated in **Figure 7**. In the following paragraphs, we explain these technologies briefly:

6.1 Intel trusted execution technology (TXT)

TXT [16] is nothing but a set of hardware extensions to allow advanced security features, such as the measured launch environment and protected execution. TXT allows the user to run a specific program in an isolated space, protecting it from

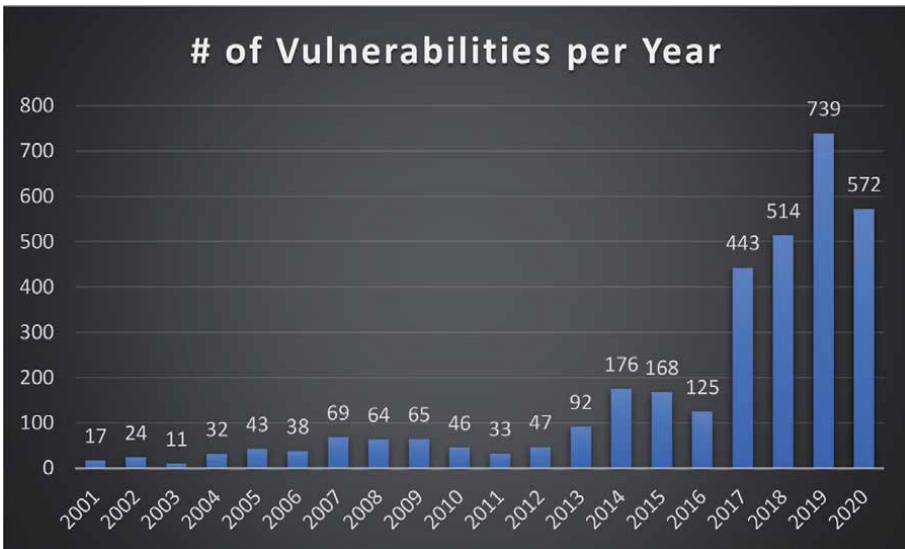


Figure 6.
Statistics of hardware and firmware vulnerabilities for the past 20 years.

other software in the system. For this reason, it improves the trust in the application’s execution environment. As a result, important data can be protected from adversaries running malicious code on the same platform.

6.2 Intel QuickAssist technology (QAT)

This technology supports crypto-acceleration and compression-acceleration in the hardware level. By offloading the security-related computation to a special adapter, the system can utilize its CPU computation power in something else [17]. For example, wireless security and routing algorithms can surely benefit from this technology.

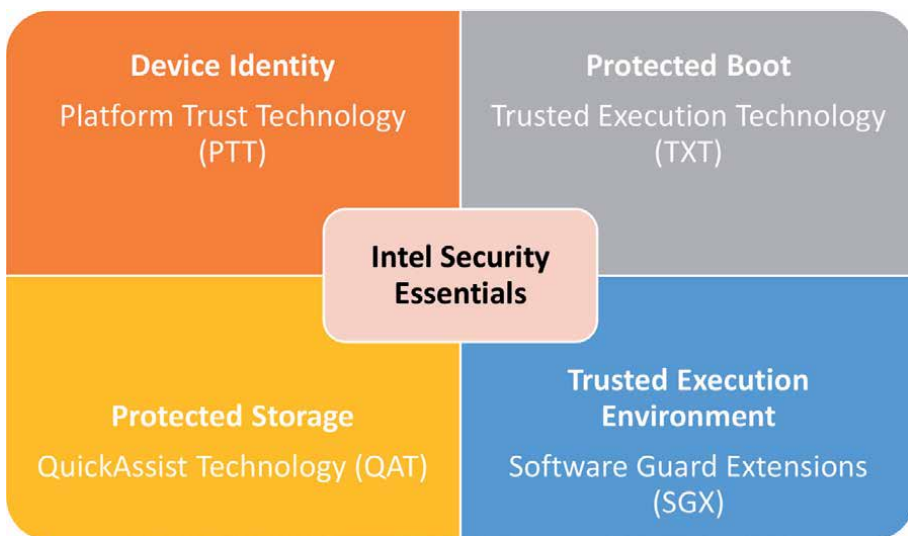


Figure 7.
Intel approach to achieve IoT security.

6.3 Intel platform trust technology (PTT)

This technology is very much related to an older technology called TPM (Trusted Platform Module). The main idea is to store keys in protected chips to authenticate hardware and software components. You can think of it as a digital fingerprint to the machine that is set by the manufacturer. PTT [18, 19] is implemented in the firmware to allow even low cost, low power devices like tablets to benefit from this technology.

6.4 Intel software guard extension (SGX)

Data can still be vulnerable while it is being stored or executed. SGX is a technology that fosters the isolation of process execution and memory allocation [20]. It empowers user-level code to have its own regions in the memory, called enclaves. These enclaves do not grant access to other processes with higher privileges. SGX basically strengthens the defenses by reducing the system's attack surface.

7. OS security in IoT

An operating system is considered the vehicle to control hardware through software. It is the lowest level software in the system hierarchy. OS security takes care of several tasks, such as separate execution and memory allocation, secret storage, and avoidance of programming errors.

The selection of the best operating system to use in IoT environments depends on several factors, like the system type, computing power, and threat level. IoT devices usually have limited power and computation capabilities resulting in limited choices of CPUs. Hence, OS security will be working on a best effort approach. Zephyr OS [21] is an open-source real time operating system that is specifically designed for resource constrained systems. It is unique in a sense that it was designed with security in mind. Consequently, it supports separate thread execution as well as separate memory storage. Moreover, it defines two levels of authority, which are the user level and the supervisor level. However, it lacks a proper authorization mechanism, which is a serious weakness [11].

As shown in **Table 2**, 80% of the top ten products with the highest number of distinct vulnerabilities reported in the past 20 years are found to be operating

#	Product name	Vendor name	Product type	# of vulnerabilities
1	Debian Linux	Debian	OS	3067
2	Android	Google	OS	2563
3	Linux Kernel	Linux	OS	2357
4	Mac OS X	Apple	OS	2212
5	Ubuntu Linux	Canonical	OS	2007
6	Firefox	Mozilla	Application	1873
7	Chrome	Google	Application	1858
8	IOS	Apple	OS	1655
9	Windows Server 2008	Microsoft	OS	1421
10	Windows 7	Microsoft	OS	1283

Table 2.
Top 10 products by total number of distinct vulnerabilities over 20 years [22].

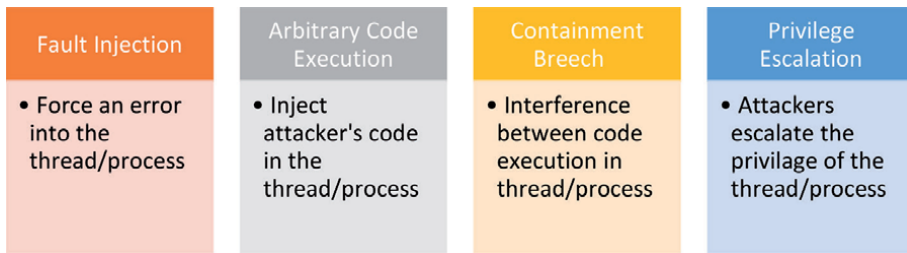


Figure 8.
Mechanisms used to compromise operating systems of IoT devices.

systems [22]. These statistics are not surprising because OS allows the attackers to access almost any part of the system with high privileges. Attackers use different ways to compromise the operating system. Some of these methods are shown in **Figure 8**. A rootkit is a good example of malware that uses these techniques to penetrate the OS and take over some of its tasks.

8. IoT network security

Most of the things in IoT will be connected wirelessly. Actually, there are many technologies available nowadays for connecting devices wirelessly, some of them belongs to Personal Area Networks (PANs), and some to Wireless Local Area Networks (WLANs) and Wide Area Networks (WANs). When talking about security in wireless networks, there are two aspects that must be considered. First, it is important to secure the data in transit using encryption mechanisms. Otherwise, anyone will have access to the data since air is a shared medium. The second one is the security of the wireless devices themselves, such as routers and access points. Unauthorized access to these devices might allow the attackers to reconfigure the network or forward the traffic to unwanted destinations.

Ethernet Time-Sensitive Networking (TSN) is state-of-the-art technology in industrial IoT that promises to bridge the gap between IT and OT [23]. Being vendor agnostic is a great feature of TSN and allows a large degree of interoperability. Furthermore, building it on top of Ethernet allowed a seamless interaction with non-TSN network devices in a plug and play fashion. Moreover, critical and non-critical traffic can co-exist with no worry about the potential increase in latency, thanks to the use of tight-time synchronization methods. Another important feature that allows the coexistence of the high and low priority traffic in the same network is Traffic Scheduling. In fact, TSN uses the notion of multiple queues to store packets with different priorities. TSN implements redundancy on the packet level by transmitting two duplicate packets through two different routes in the network. The one that arrives earlier will be processed whereas the other is simply discarded. This is a great way of assuring reliability in industrial-based networks. Finally, it is important to note that it is possible to use TCN as a link-layer protocol in any framework. OPC-UA is an example of such a case [24].

9. Conclusion

Security and privacy are important aspects of IoT networks. Given the widespread use of IoT devices in many fields, keeping the network secure is becoming increasingly important. Similarly, preserving data integrity is essential, especially

when IoT sensors are used in the medical field. In this chapter, we have encouraged and supported the need for IoT security and privacy by giving examples of past attacks on IoT networks. Then, we described in detail the building blocks of IoT trustworthiness to illustrate the challenges facing IoT system engineers. After that, we explained the typical lifecycle of an IoT device, starting from the development phase until the device is retired. Moreover, we introduced the concept of end-to-end security in IoT networks. Also, we compared seven different approaches for securing IoT platforms and highlighted the strengths and weaknesses of each approach. Finally, we briefly presented challenges and potential solutions for securing IoT from the OS, hardware, network, and from a device point-of-view. The Intel approach to achieve IoT Security is presented as an example.

Nomenclature

Section 1

IT	Information Technology
OT	Operation Technology
IP	Internet Protocol
IoT	Internet of Things
DDOS	Distributed Denial of Service
DVR	Digital Video Recorder
SSH	Secure Shell
USB	Universal Serial Bus
PLC	Programmable Logic Controller

Section 2

TEE	Trusted Execution Environment
-----	-------------------------------

Section 3

SDK	Software Development Kit
API	Application Programming Interface
CPU	Central Processing Unit
SoC	System on Chip
OS	Operating System

Section 4

DB	Database
OCF	Open Connectivity Foundation
UPnP	Universal Plug and Play
LWM2M	Lightweight Machine 2 Machine
DTLS	Datagram Transport Layer Security
OneM2M	One Machine to Machine
OPC-UA	Open Platform Communications-Unified Architecture
TLS	Transport Layer Security
HTTPS	HyperText Transfer Protocol Secure
DDS	Data Distribution Service
ACL	Access Control List

Section 5

DICE	Device Identity Composition Engine
TCG	Trusted Computing Group
UDS	Unique Device Secret

Section 6

TXT	Trusted Execution Technology
QAT	QuickAssist Technology
PTT	Platform Trust Technology

TPM	Trusted Platform Module
SGX	Software Guard Extensions
Section 8	
PAN	Personal Area Network
WLAN	Wireless Local Area Network
WAN	Wide Area Network
TSN	Time-Sensitive Networking

Author details

Ahmad J. Showail^{1,2}

1 Taibah University, Madinah, Saudi Arabia

2 University of Prince Mugrin, Madinah, Saudi Arabia

*Address all correspondence to: ashowail@taibahu.edu.sa

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Foundation Fieldbus [Internet]. Available from: <http://www.foundationfieldbus.com/> [Accessed: 2021-01-11]
- [2] Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A. R., & Tarkoma, S. (2017, June). Iot sentinel: Automated device-type identification for security enforcement in iot. In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS) (pp. 2177-2184). IEEE.
- [3] Koliass, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7), 80-84.
- [4] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J.,... & Zhou, Y. (2017). Understanding the mirai botnet. In 26th {USENIX} security symposium ({USENIX} Security 17) (pp. 1093-1110).
- [5] Dynamic DNS [Internet]. Available from: <https://account.dyn.com/> [Accessed: 2021-01-11]
- [6] Mirai Botnet (malware) [Internet]. Available from: [https://en.wikipedia.org/wiki/Mirai_\(malware\)](https://en.wikipedia.org/wiki/Mirai_(malware)) [Accessed: 2021-01-11]
- [7] Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3), 49-51.
- [8] Schiffman, J., Moyer, T., Jaeger, T., & McDaniel, P. (2011). Network-based root of trust for installation. *IEEE Secur. Priv.*, 9(1), 40-48.
- [9] Abera, T., Asokan, N., Davi, L., Koushanfar, F., Pavard, A., Sadeghi, A. R., & Tsodik, G. (2016, June). Things, trouble, trust: on building trust in IoT systems. In Proceedings of the 53rd Annual Design Automation Conference (pp. 1-6).
- [10] Ammar, M., Russello, G., & Crispo, B. (2018). Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications*, 38, 8-27.
- [11] Cheruvu, S., Kumar, A., Smith, N., & Wheeler, D. M. (2020). Demystifying Internet of Things Security: Successful IoT Device/Edge and Platform Security Deployment (p. 488). Springer Nature.
- [12] Bernstein, D. J. (2010, May). Grover vs. McEliece. In International Workshop on Post-Quantum Cryptography (pp. 73-80). Springer, Berlin, Heidelberg.
- [13] Jäger, L., & Petri, R. (2020, August). DICE harder: a hardware implementation of the device identifier composition engine. In Proceedings of the 15th International Conference on Availability, Reliability and Security (pp. 1-8).
- [14] Device Identity Composition Engine (DICE), Trusted Computing Group [Internet]. Available from: <https://trustedcomputinggroup.org/work-groups/dice-architectures/> [Accessed: 2021-01-12].
- [15] The National Institute of Standards and Technology (NIST) Vulnerability Database [Internet]. Available from: <https://nvd.nist.gov/vuln/search> [Accessed: 2021-01-12].
- [16] Intel Trusted Execution Technology (TXT) [Internet]. Available from: <https://www.intel.com/content/www/us/en/support/articles/000025873/technologies.html> [Accessed: 2021-01-12].
- [17] Intel QuickAssist Technology (QAT) [Internet]. Available from: <https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/quickassist-adapter-8950-brief.pdf> [Accessed: 2021-01-12].

- [18] Intel Platform Trust Technology (PTT) [Internet]. Available from: <https://static.onlogic.com/resources/downloads/OnLogic-PTT-One-Pager.pdf> [Accessed: 2021-01-12].
- [19] Strengthening Security with Intel Platform Trust Technology. [White Paper]. Available from: <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/enterprise-security-platform-trust-technology-white-paper.pdf>. [Accessed: 2021-01-12].
- [20] Intel Software Guard Extensions (SGX) [Internet]. Available from: <https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html> [Accessed: 2021-01-12].
- [21] The Zephyr Project [Internet]. Available from: <https://zephyrproject.org/> [Accessed: 2021-02-05].
- [22] Top 50 Products By Total Number Of "Distinct" Vulnerabilities [Internet]. Available from: <https://www.cvedetails.com/top-50-products.php> [Accessed: 2021-01-12].
- [23] Finn, N. (2018). Introduction to time-sensitive networking. *IEEE Communications Standards Magazine*, 2(2), 22-28.
- [24] Schwarz, M. H., & Börcsök, J. (2013, October). A survey on OPC and OPC-UA: About the standard, developments and investigations. In 2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT) (pp. 1-6). IEEE.

An IoT Based Cloud Deployment Framework for Effective Classification of Machine Conditions

Ganga Dhandapani and V. Ramachandran

Abstract

Cloud services are proposed for real-time data acquisition, data classification, data processing and decision making, which are highly interconnected services for effective condition monitoring of electrical machines. The proposed Software as a Service, Storage as a Service and Platform as a Service layers address the challenges of data storage and scalability while making analysis on the cluster of machines in an Industrial Environment. An experimental setup consisting of two DC motors coupled to AC Generator operating at different locations is considered to evolve the proposed model for effective integrated monitoring and decision making. This cloud-based vibration monitoring model provides services for data acquisition from the IoT devices mounted on the shafts of the DC motors, data storage to store the enormous amount of acquired signal data from multiple sensors, data classification of vibration signals for effective statistical analysis to estimate adaptive cluster of thresholds and appropriate decision-making services on demand over the Internet to utilize the reliable service of the machines in a persistent way. The computational engine will do inherent statistical analysis of the vibration signals to estimate the cluster of thresholds adaptive to various operating conditions. The services have been deployed without any limitation in a cloud environment and the industrial applications can share information using the deployed services from anywhere on demand basis. The deployed cloud service for the enhanced statistical classification algorithm eliminates the false identification of failures, which not only increase the availability of machines for intended operations but also reduce the maintenance cost. The resulting threshold values are compared with that of the vibration analysis carried out on the machine beds locally using myRIO for data acquisition in LabVIEW and the proposed model ensures the integrity in appropriate decision making with assured scalability.

Keywords: condition monitoring, electrical machines, data acquisition, classification, web services, Google Cloud, scalability

1. Introduction

The electrical machines find wide and crucial applications in various industries and power plants. Condition monitoring of electrical machines is extremely

significant for making the industrial processes more efficient with reduced downtime. Condition monitoring is heading as a real-time task, which requires maximum accuracy and embraces a gradual paradigm shift from legacy systems to modern Internet of Things (IoT) enabled systems at every level namely data acquisition, data processing, data integration and decision making. The uptime and efficiency of the plant operations shall be maximized through proper condition monitoring diagnostics and as well as preventive or predictive maintenance. The current condition monitoring systems make effective decisions using the knowledge repository, which is populated using various algorithms by the way of observing and storing the defective and unusual behaviour details of the machines. The major challenge faced by many industries is not only inadequate storage space but also the scalability when many machines inside the plant or operating at remote locations are to be monitored online and enormous amount of data have to be acquired from the machines for the interpretation of their behaviour at dynamic or abnormal operating conditions.

In the late 90's, very few online condition monitoring applications came into existence with the primary motive to collect vibration signals from various machines operating at different locations, but the analysis has been made considering each local operating environment to make effective decisions. This methodology leads to better preventive maintenance, but predictive maintenance is still a challenge. During those initial stages of online condition monitoring, the accessibility of such applications was through personal computers and laptops via World Wide Web. The scalability is the major issue in the current on-line condition monitoring applications. Over the period, due to the advancements in networking technologies, higher data rates for communication have paved a path to expansion in the field of cloud services. Especially, the industries which are facing the challenges such as inadequate storage space for data and scalability, the cloud environment will provide appropriate solution to those issues.

The cloud based model must be designed to handle operations by various industries without any hurdle to exchange data due to heterogeneous nature. The cloud environment provides inherent dynamic scalability for the operations of electrical machines at different locations at different operating conditions. Cloud computing does not require global standard architectures, and it does not necessarily need a standard, open, general purpose protocol. Furthermore, cloud computing supports interfaces that are syntactically simple, semantically restricted and of high-level. The cloud environment provides an added value of being able to share and compare the local machine condition data with other similar machines across the plant, or with other machines at multiple plants wherever they are located.

The growth of data analysis methods such as statistical, signal processing and machine learning techniques has moved condition monitoring of electrical machines towards the regime of predictive maintenance with the application of predictive analytics. Integrating the predictive maintenance techniques with the IoT enabled technologies will enable the industries to avoid unnecessary equipment replacement and improve process safety, availability and efficiency. Predictive maintenance adopted in industries employs predictive analytics to detect the problems well ahead to the occurrence of failures using which the corrective measures are planned. Prediction avoids unexpected process failures and prolongs the life of the system. In condition monitoring, though threshold estimation has more significance, it has not yet been given due consideration and thousands of false alarms are generated in dynamic operating conditions due to adoption of default threshold levels. The precise and faster short-term forecasting of machine's physical signals predict the probability of failures and intensity of deterioration during abnormal conditions and provide performance optimization under normal or dynamic

operating conditions. Condition monitoring stands effective only when the process of extracting information from the data becomes faster with more details. In this chapter, a new scalable and reliable model has been investigated to perform online condition monitoring of multiple machines in real-time industrial conditions and to perform predictive maintenance to enhance the process coordination and fault tolerance in industrial automation.

The statistical classification based vibration analysis algorithm has been developed as a Web service and deployed as a cloud service to demonstrate real-time condition monitoring of electrical machines. The proposed cloud based condition monitoring system collects the vibration data of machines from various locations and processes the same in the cloud by comparing the data of one machine with the data of other similar machines for reliable and effective decision making. These features of interfaces are underlying factors for rapid adoption of cloud computing services in the condition monitoring applications.

2. State of the art

More research works regarding condition monitoring and predictive analysis are carried out for accurate assessment and prediction of machine conditions in real-time. The prediction models designed for monitoring real-time operation of electrical machines need to be robust and online in order to make accurate and faster data prediction.

It is hard to fulfill the practical requirement of application specific scientific approaches for the industries while performing real time data analysis and condition assessment towards preventive maintenance as the machines are operating at different environmental conditions. Diego Galar et al. [1] have cited that single valued thresholds provided by the manufactures are not suitable for fault identification under non-stationary operations, environmental changes and aging. The vibration severity characterized by ISO 10816 has been specified as a static threshold suitable for new machines and said to provide incorrect reference for machines in use. Instead, the authors provide dynamic thresholds adaptive to operational conditions as a better solution for SMART maintenance.

Continuous monitoring and measuring of machine parameters such as vibration, temperature, etc., with and without external disturbances will lead to make proper decisions for effective maintenance and thereby prolong the useful life and reliable operation of electrical machines. Recently, smart Internet of Things technologies are evolving for effective condition monitoring of electrical machines. IoT enables online monitoring of the machine as it runs and data have been acquired by an embedded device or a gateway and transmitted to a server for analysis and maintenance scheduling. The practical challenges faced by maintenance engineers are the introduction of new technologies for the enhancement of plant productivity, methods of data acquisition and analysis, inconsistent outcomes and shortage of resources. R. Kirubashankar et al. [2] has explained about Internet based automation architecture for the control of the devices and equipments of process plants for optimal control and reduced unplanned downtime. A Web based architecture has been proposed for the control of devices remotely over Ethernet with Programmable Logic Controller (PLC) and Supervisory Control and Data Acquisition (SCADA) system networked in Virtual Private Network (VPN). Larry Combs [3] has stated that the functionality and reliability of conventional SCADA gets enhanced if it is hosted as Software as a Service in cloud platform. IoT enabled condition monitoring will identify potential problems using sensors and able to take necessary preventive measures before any issues occur and hence prevent damages and reduce

maintenance costs. The changes in vibration, temperature etc., have been tracked by sensors and any issues such as misalignment, imbalance etc., shall be detected and accordingly service maintenance is scheduled automatically ahead of time to prevent failures thus avoiding unplanned downtime.

Omid Givehchi et al. [4] had designed a general cloud based architectural model that allows automation functions in industries to be offered as services from a dynamic infrastructure. The authors have envisaged the importance of cloud solution for the control and field levels of automation. The physical devices at these levels are integrated by encapsulating the services and functions inside the delivery standards of cloud. Cloud computing is seen as solution to provide platform for integration of growing information technologies such as Internet of Things, Service Oriented Architectures and mobile computing. Omid Givehchi and Jasperneite [5] have delivered the Virtual PLC as 'Control as a Service' through Microsoft Azure cloud environment. Omid Givehchi et al. [6] have illustrated about the development of Virtual PLCs on the Virtual Machines of the private cloud created using VMware's vCloud suite.

Cloud technology can be applied in two ways for automation of industrial processes. They are collaborative application development and real time publishing of data to the cloud server for remote monitoring and control. Though real time publishing is adopted for remote monitoring, collaborative applications also has equal significance. This is because of the option available to decide the factors of design, customization, updates and changes in the system before deployment. Also collaboration allows multiple consumers to monitor the process data simultaneously. This is highly required in places where the systems are inherently distributed such as irrigation systems, wind farms, cell towers, agriculture etc. The SCADA system maintained as a central monitoring system in local network, when taken to cloud server with collaborative software tools can aid in easy information exchange at multiple locations. The real time publishing can be made so easy in SCADA using SaaS (Software as a Service) in cloud. The software running the application will be on cloud server enabling easy and secured data publishing and data request. This also sends the data to multiple clients such as iPad, smart phones or other networks. The SCADA built in cloud with software tools providing collaboration leads to the creation of new business model using direct and shared access of multiple processes to a control expert. Further, on the data received from industrial systems, analysis can be made in the server side and feedback can be provided [7].

In condition monitoring, the comparison of the data acquired with that of baseline standards is a widely adopted strategy [8]. NI in its artefact of fleet wide monitoring emphasizes the importance of continuous and automated data collection from industrial assets in order to realize meticulous comparative results so that real-time maintenance decisions are improved significantly. Such kind of maintenance strategy could successfully be achieved only with IoT based condition monitoring of industrial assets executed in cloud platform [9]. The cloud can be of public, private or hybrid nature. Fran Dougherty, CTO of the Worldwide Incubation Enterprise and Partner Group of Microsoft had appreciated the use of private and public clouds by industries for innovation, scalability and business growth in the special report composed by Jim Montague [10]. However, hybrid cloud was considered to be the best option by him, as industries can choose the type of analysis dynamically as per the requirements. Advantech in its white paper [11] has discussed on the importance of the implementation of cloud-based SCADA system using Industrial IoT and points out that the adoption of cloud offers pervasive analytics and decisions additionally irrespective of the hardware used and thus making Industry 4.0 effective. Steve Lacey [12] while discussing the ground realities of condition monitoring in industries asserted the need for skilled technicians

for predictive maintenance. The author perceived cloud based condition monitoring to expand the monitoring horizons and assure direct connectivity with maintenance specialists whose availability has always been a challenge and also hinted on the execution of cloud based condition monitoring in Schaeffler due to various benefits. New analysis techniques have been implemented when unknown signal patterns are observed at the user end.

The plant-wide condition monitoring of rotating electrical machines have extensively been discussed by Mallikarjun Kande et al. [13]. The existing machine condition monitoring and industrial automation techniques have been reviewed and the application of artificial intelligence for machinery diagnostics has been perceived as the future scope. While discussing about on-equipment and on premise integration methods, the need for on-cloud monitoring using IoT gateway has been substantiated to meet the requirements of advanced diagnostics and data platforms for enhanced computation. The condition monitoring system and Distributed Control System are integrated over the cloud for continuous monitoring of the equipment with high update rates from the sensors and for effective diagnosis. The efficient integration of various data acquisition and other devices in real time demands lightweight and uniform communication standards.

A comprehensive investigation has been made [14] that shifts the focus from the monitoring of specific machine components for fault prognosis to an approach scanning the overall system execution in an integrated manner to deliver desired performance for the application in an optimum manner. The authors address the challenge in the determination of absolute vibration thresholds adaptive to the machine operating conditions for reliable condition monitoring. A statistical classification based signal decomposition algorithm has been proposed for segmented vibration signal analysis as a measure of improving the precision in condition monitoring of electrical machines.

In this chapter, IoT based cloud services for real-time condition monitoring of electrical machines are proposed. It is focused towards estimation of vibration thresholds adaptive to the machine condition, which persuades to realization of incipient and critical abnormal conditions fully. Considering the immense raise towards the importance of predictive maintenance applications and connected IP based data acquisition devices, a generalized cloud framework is proposed to provide services for effective condition monitoring diagnostics and to maintain a knowledge repository for effective decision making with respect to maintenance scheduling. The main objective of this chapter is to explore real-time implementation of IoT enabled cloud services to formulate pre-emptive, strategic and operational decisions. The proposed IoT based model for vibration analytics of electrical machines addresses the challenges of data storage and scalability. A Web Application Framework has been developed by introducing cloud services for real-time data acquisition, data classification, data processing and decision making for effective condition monitoring.

The statistical classification based signal decomposition algorithm discussed in [14, 15] identifies the denser vibrating levels of machine under dynamic operating conditions and enumerates cluster of thresholds adaptive to the operating conditions for quick and accurate prediction of abnormalities. This algorithm is integrated with IoT based model through LabVIEW client application to enable real-time condition monitoring of machines located anywhere whose data are acquired by sensors and transmitted to cloud storage. The cloud services which have been developed for data acquisition and processing are tested locally before deploying in the Django Web Framework and implemented in Google Cloud Platform. The results of the classification algorithm, i.e., the adaptive threshold class clusters pertaining to each machine are used to create contextual vibration references for

making efficient and quick decisions in the condition assessment of machines of same type exposed to similar operating conditions.

3. Cloud services for effective condition monitoring

An effective cloud-based model is proposed to estimate the cluster of thresholds adaptive to various operating conditions of the electrical machines, to eliminate the identification of false failures or alarms, and to make decisions for effective maintenance scheduling. To make its implementation more general and scalable for real-time analysis, cloud services are introduced in different layers in accordance with Model-View-Template (MVT) pattern of Django Web Framework (<https://media.readthedocs.org>) and deployed in Google Cloud Platform. The “Models” represent the structure and manipulation of data, “Views” encapsulate the processes both at server and client ends and “Templates” present the rendered information to the end user. The basic building blocks of the proposed cloud-based model are to analyse the various dimensions and metrics such as, density of oscillations between classified amplitude levels, maximum number of oscillations, oscillations with similar and dissimilar amplitudes and the adaptive thresholds etc.

Cloud services are deployed for real-time data acquisition, data classification, data processing and decision making, which are highly interconnected services for effective condition monitoring of electrical machines. The proposed Software as a Service, Storage as a Service and Platform as a Service layers address the challenges of data storage and scalability while making analysis on the cluster of machines in an Industrial Environment. An experimental setup consisting of two DC motors each coupled to AC Generator operating at different locations is considered to evolve the proposed model for effective integrated monitoring and decision making. The threshold values estimated using cloud services are compared with that of the vibration analysis carried out on the machine beds locally using myRIO for data acquisition in LabVIEW ensures the integrity of the cloud-based model with assured scalability. Though security and big data processing overheads are encountered in the connected enterprises, employing cloud computing has been widely embraced by industries for its collaborative nature, optimized performance, better diagnostics, higher productivity and sustainability. IoT and cloud-based processing have been adopted for condition monitoring of multiple machines operating at different locations as they evolve as a better choice due to the attributes of cloud storage, flexible application development, data aggregation, scalability and platform of multiple services. The deployed cloud services eliminate the false identification of failures, which not only increase the availability of machines for intended operations but also reduce the maintenance cost.

3.1 Proposed Cloud-based condition monitoring model: a layered approach

The proposed cloud services for machine vibration monitoring using IoT based framework have been modelled as a layered architecture as shown in **Figure 1** and implemented on the experimental set up as shown in **Figure 2**. The main layers [16] of the proposed model are the Platform as a Service (PaaS) layer, which is Google Cloud Platform, Software or Application as a Service (SaaS) layer where all the proposed services are deployed and Storage as a Service layer where all the data stores have been maintained. In online condition monitoring applications, the sensors are used to acquire vibration signals and communicated to the cloud storage for further processing. The proposed framework will enhance the machine condition monitoring functionality with methodologies of scalable and platform independent

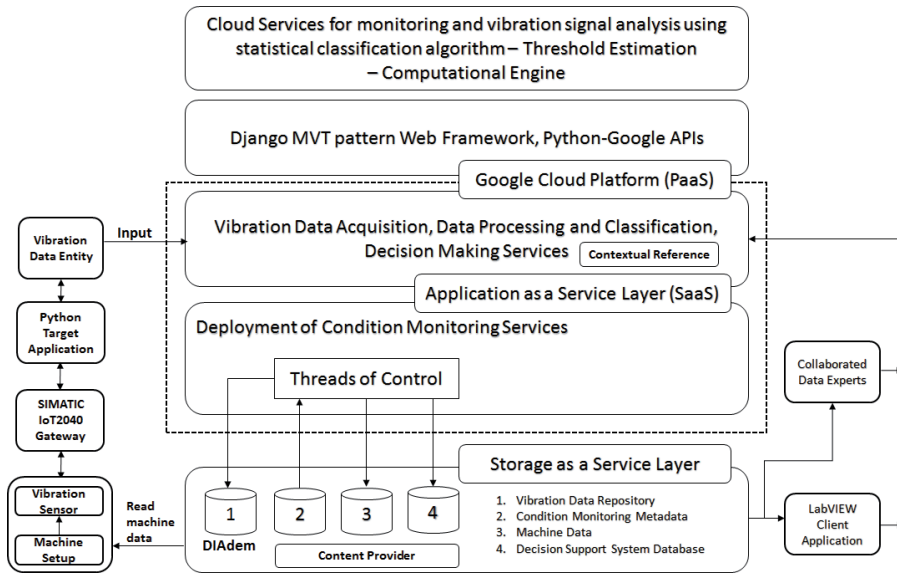


Figure 1.
 Cloud Services - Layered Architecture for Condition Monitoring Model.

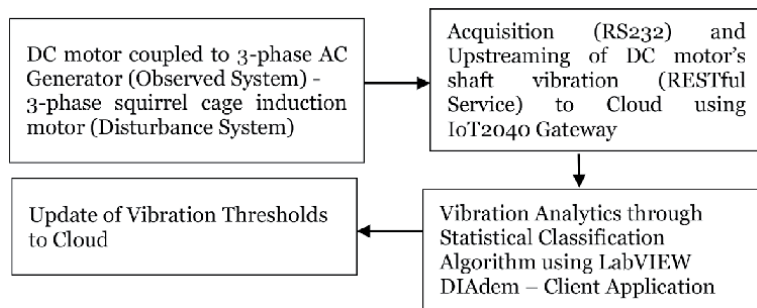


Figure 2.
 Experimental Setup.

data aggregation and collaborative analysis that the real-time industrial applications demand extensively.

The cloud infrastructure provides the fundamental resources needed to share upper level platforms and services. The “Storage as a Service” layer (Model perspective) and the physical resources needed for the “Computational Engine” and for communication among services along with core services for condition monitoring (View perspective) form the basis for delivering Infrastructure as a Service (IaaS). Google App Engine is used as Platform as a Service (PaaS) in the proposed model that provides a conducive environment to implement the cloud services. Google Cloud provides a set of APIs to aid the interaction between cloud components and end user applications, to enhance scalability, and ease deployment and management.

The Django Web Framework provides an environment to deploy the services developed in accordance with Model-View-Template pattern and the “google-cloud” API for Python provides interfaces for interacting with Google App Engine, which is configured as Python based cloud runtime environment. Software as a Service layer (SaaS) is for end users i.e., for the application developers. The services in this layer are typically accessed through Web portals using Templates. The

condition monitoring services provided with this model are normally referred to as SaaS and implemented in Views of Django Web Framework. The content providers of various services using a separate thread of control for each service, which have been developed by the registered end users, i.e., participating industries have stored the data in different formats under heterogeneous environment. The server application in the cloud environment has the control to manage the databases maintained in the Storage as Service layer. With the App Engine, each participating industry can write their application code, test it on their local machines and then deploy on cloud environment.

3.2 Creating, uploading and registering Cloud services for condition monitoring

Google App Engine is used for registering, uploading and accessing the condition monitoring services in the proposed cloud-based model for estimation of adaptive thresholds among various machines. The App Engine Software Development Kit (SDK) for Python is used to create and to link the services to the cloud. The location independent “google-cloud” client API as well as the shell SDK are providing the Python based cloud interfaces which can be accessed by the Django Web Services framework and Internet browser. The cloud applications need a configuration file i.e., “app.yaml” to deploy and run the application. The condition monitoring and decision making services configured using App Engine are easy to build, easy to maintain, and easy to scale as traffic and data storage needs grow. The services have been uploaded for ready to serve to any of number of machines located at distributed industrial environments.

3.3 Configuring condition monitoring services in Django Web framework

The Django “admin” is used to create a project named “ConditionMonitoring” using the command “django-admin startproject” with an application registered as “MonitoringApp”. The python script, “settings.py” defined within the project enumerates the default backend database and all the registered applications as detailed below:

```
#settings.py
DATABASES = {'default': {'ENGINE': 'django.db.backends.mysql',
                        'NAME': 'VIBRATION_DB', 'USER': 'username',
                        'PASSWORD': 'password', 'HOST': '192.168.1.112', 'PORT':
                        '5000'},}
INSTALLED_APPS = ['django.contrib.admin', ..., 'MonitoringApp', ]
```

The python script, “manage.py” defined within the project starts the Webserver, migrates and synchronizes and flushes the databases if required. The script, “urls.py” defines the URL patterns to link the Views. The Views render the request/response to the Templates as XML for processing and HTML for presentation. The application, “MonitoringApp” describes and defines the required services in the Views. In the proposed cloud framework, the Model represents the Storage as a Service layer (virtual storage), the View represents the Application as a Service layer (SaaS) and the Template represents the presentation tier which includes Computational Engine.

3.3.1 Model

The logical schema for various database tables used in real-time condition monitoring to store the machine data, repository for previous decisions, maintenance schedules, data pertaining to historical conditions (i.e., decision support system) are implemented as Model entities under Django Web Framework, where each Model maps to a single database table dynamically. Model is the single definitive source of information about the data and it is defined in the “models.py” script. The metadata of the vibration signals acquired for condition monitoring of every machine, the metadata for the specification of machines of all participating industries and the metadata used for “Decision Support System” are described in the Model as follows:

Vibration Signals Metadata

[Component Name, Operating Condition, Disturbance Nature, Input Current, Vibration, Samples]

Machines Metadata

[Machine_ID, Machine_Type, Rated Voltage, Rated Current, Rated Speed]

Decision Support System Database (Repository)

[Machine_ID, No. of Classes, Range of Classification, Class Width, Total Oscillations, Upper Threshold Class Cluster, Lower Threshold Class Cluster, Excess Positive Slopes, Excess Negative Slopes]

3.3.2 View

In general, the View retrieves data according to the path parameters defined in the “URL patterns” list, loads a template, renders the template with the retrieved data and returns the HTTP Response instance as output. Each View is a python service and an appropriate View is chosen by examining the URL that is requested as per the configurations made in the URL patterns. The codes pertaining to acquisition of vibration data, statistical classification algorithm, threshold estimation etc., are defined as python functions in the View’s sub-directory of the application.

The vibration data of the DC motor in the specially created experimental setup have been acquired when started at no load condition as well as loaded by AC Generator at fixed load changes. It is well known that the factory floor generally has lot of machines running together. Thus to create a similar field condition of the factory floor, an additional motor was installed nearer to the DC motor in the experimental setup and the shaft vibration data are again acquired for the stated conditions. The acquired vibration data under the operating conditions of starting to no load speed with and without external disturbance and loading are streamed to cloud through IoT2040 gateway. The LabVIEW client application enables collaborated real-time condition monitoring of any machine by integrating the non-stationary vibration analysis algorithm with a cloud service. The analysis results updated to the decision-making service lead to effective condition monitoring and make the maintenance of other connected devices/machines automatic and perfect scheduling. The updated results create contextual vibration references for assessing the condition of any other machine of same type that has been exposed to similar operating conditions.

3.3.3 Templates

A project can be configured with one or several template engines (<https://media.readthedocs.org>). Django defines a standard API for loading and rendering templates regardless of the backend. Loading consists of finding the template for a

given identifier and pre-processing it, rendering means interpolating the template with context data and returning the resulting string. The “templates” sub-directory should be created within the application directory by the end user manually. The template “dataframe1.html” to receive the response from the “ConditionMonitoring_StandAlone” service is defined as follows:

```
<html>
<body bgcolor="#bg99FF">
<p>Type:{{Machine_Type}}</p>
<p>AcquisitionDevice:{{ SensorList }}</p>
<p>Machine_ID: {{Machine_ID}}</p>
<p>OperatingCondition:{{ Disturbance / Standalone / loading }}</p>
<p> DisturbanceNature: {{Constant Speed}}</p>
<p> Vibration: {{ samples .tdms }}</p>
<p>Speed:{{RatedSpeed}}</p>
</body>
</html>
```

The corresponding URL patterns entry is mentioned in the “urls.py” as follows:

```
from django.contrib import admin
from django.urls import path
from import .views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('dataAcquisition/', views.acquireVibrationData),
    path('dataframe1/', views.ConditionMonitoring_Standalone),
    path('dataframe2/', views.ConditionMonitoring_Disturbance),
    path('dataframe3/', views.ConditionMonitoring_Loading),
    path('analysis/', views.StatisticalAnalysis),
    ... ..]
```

To communicate between various services, the data acquisition as well as estimated threshold values have been generated as XML for which the corresponding schemas are defined in the View’s subdirectory. The XMLized representation of the dataframe generated by the ConditionMonitoring_Standalone service is given below:

```
<!--Dataframe representation-->
<?xml version="1.0"?>
<Dataframe>
  <Machine_Type>DC Motor</Machine_Type>
  <Machine_ID>EE-M5864</Machine_ID>
  <OperatingCondition>Standalone – Starting to No load speed
</OperatingCondition>
  <DisturbanceNature>NIL</DisturbanceNature>
  <Vibration>snl.tdms</Vibration>
  <Speed>1500</Speed>
</Dataframe>
```

In accordance with the response of the “ThresholdEstimation” service, the “DecisionMaking” service generates appropriate maintenance schedules in the XML form and helps in segregating false identification of failure status. If the attributes

of any data store have been changed, the corresponding Model updates the entries in the XML representation dynamically through its Views.

3.4 Deployment of vibration analysis based condition monitoring on Google Cloud platform

Upon testing the Condition Monitoring Django Application in the local machine with the local SQL server running in the backend, the application is deployed in the Google App Engine Standard Environment (<https://cloud.google.com>) [17]. For deploying the application in the Cloud, a Google Cloud Platform (GCP) project is created in the GCP console. Further, the respective operating system's compatible Google Cloud SDK Shell is installed in addition to the Python Google-Cloud client libraries in the local development environment. The APIs are enabled in the GCP console and the Cloud SQL Proxy, which provides a secure access to the Cloud SQL is also installed with respect to the operating environment. The deployment of the application in Google Cloud is explained in the sequence of steps mentioned below:

3.4.1 Creation and Initialization of Cloud SQL Instance

Google Cloud provides Cloud SQL as Storage as a Service (SaaS), which supports all the database transactions with respect to the application to be deployed in the Cloud, which have been controlled by user defined threads of control.

3.4.2 Configuring the application with the Google Cloud

The configurations for the Database dictionary in “settings.py” with MySQL are set accordingly as per the Cloud SQL Instance connection name, database user, password and port.

3.4.3 Execution of the application in the local development environment

Before deploying the application, verification of the same is carried out in the local development environment by following the standard Web application execution procedure defined by Django Web Framework. The migrations are also carried out to set up the Models.

3.4.4 Deploying the application on the Google App Engine Standard Environment

All the static files of the application are gathered into a single directory by executing the command “python manage.py collectstatic”. These static files are moved to the production site while deploying the application to the Google Cloud. The “requirements.txt” file is created to mention the dependencies and “app.yaml” which contains the environment, runtime and entry point is also created and finally, the application can be deployed on to the Google Cloud by executing the command “gcloud app deploy”. Threads of control are introduced to populate the content provider by initiating the services in parallel mode to gather vibration data, machine data, to analyse the vibration signal data and to store the historical information in the “Decision Support System” data store for validation and further processing.

3.5 Statistical classification algorithm: computational engine

A new vibration analysis algorithm developed with statistical classification and clustering extracts the detailed signal features and estimates adaptive thresholds [14]. This signal decomposition algorithm, which forms the base for computational engine of the proposed IoT based cloud services model tracks the vibration signal transitions between the classes at multiple levels of amplitudes and computes the number of oscillations between the levels. While classifying the signal, the amplitude is segmented into a desired set of classes of equal and desired width. A transition matrix with ‘n’ number of classes and signal transitions is formulated out of the statistical classification of the data read from the cloud and streamed to DIAdem [18] as “.tdms” file by client application developed in LabVIEW platform for computation of the oscillation nature of the machine vibration.

In the proposed vibration analysis technique, the number of signal transitions has been considered to calculate the oscillations between every class and other classes to extract the threshold levels of vibration. The transition of the signal from a lower class to higher class and vice-versa are accounted as positive and negative slope respectively in the transition matrix. The analysis has been carried out by considering the transition matrix and progressing through the upper diagonal matrix row-wise and lower diagonal matrix column-wise elements or its vice-versa. The ‘n’ column vectors of upper diagonal matrix give the signal transitions from a class corresponding to the row to class of the diagonal element and ‘n’ row vectors of the lower diagonal matrix represent the signal transitions from class of the diagonal element to lower class corresponding to the column. The proposed algorithm determines the oscillations in the real time non-stationary vibration signal at multiple class levels using statistical classification of the signal amplitude as well as the transition matrix and delivers the features as Oscillation Matrix. The dominant classes of the Oscillation Matrix having higher count of oscillations with that of lower classes have been clustered together and identified as upper and lower threshold class clusters.

This algorithm is validated on the vibration signals acquired by IoT2040 gateway which is integrated with Google Cloud Platform and offered as cloud service in real time while testing cloud based condition monitoring. The computational engine has been integrated with the signal analysis to add more investigations towards the changes happening in the vibration during different operating conditions and offer adaptive thresholds for condition monitoring.

3.6 Integration and evaluation of the Cloud services based vibration analytics algorithm

To realize the effectiveness of the vibration thresholds estimated from the IoT based data analysis, a comparative analysis has been made with the results of vibration data acquired through myRIO. The same experimentation has been carried out for acquiring the shaft vibration data using myRIO1900 [19] as acquisition device and tri-axial accelerometer (ADXL345) as vibration sensor. The data acquired in both cases have been fed to the computational engine for carrying out statistical classification analysis for fixing the clusters of vibration thresholds precisely at the following dynamic operating conditions:

- Starting to no load speed and load changes in standalone mode of operation
- Starting to no load speed with induced external disturbance

In all test cases and operating conditions, the single valued vibration threshold is replaced with a cluster of upper and lower thresholds for safe operation of a machine. This has been implemented on 3,96,000 samples of vibration signal acquired from the DC motor under the stated conditions. The acquired data are logged in an excel file and imported to the computational engine in which the statistical classification technique is implemented using NI DIAdem [18]. This technique has been tested on the DC motor shaft vibration signals at the above stated operating conditions using different classification criteria. In each case, the same vibration signal is segmented to study about the oscillations pattern.

The shaft vibration signal of DC motor pertaining to the operating conditions of starting to no load speed (standalone and disturbance conditions) and loading (standalone condition) have been acquired by myRIO application developed using LabVIEW FPGA and RT programming through the tri-axial accelerometer ADXL345 having sensitivity of 256 LSB/g. The vibration data during the same operating conditions are acquired by IoT2040 gateway [20] from a Piezo electric sensor through serial interface under the measurement unit of ‘g’ and transferred the scaled value in LSB to cloud simultaneously. The upper and lower threshold class clusters are identified with the application of signal processing algorithm on the vibration signal and the non-stationary vibration data corresponding to different operating conditions are characterized using the transition matrix obtained from the statistical classification.

3.6.1 From Starting to No Load Speed – Standalone Condition

The acquired data pertaining to this operating condition with respect to myRIO and IoT Gateway (as shown in **Figure 3**) applications have been fed to the computational engine for statistical classification and analyzed with different classification configurations. The observed signal peak values and the configuration settings assumed for classification in each case are listed in **Table 1**.

The vibration analysis is carried out by the computational engine and the estimated upper and lower threshold clusters are rendered to the “VibrationAnalysis”

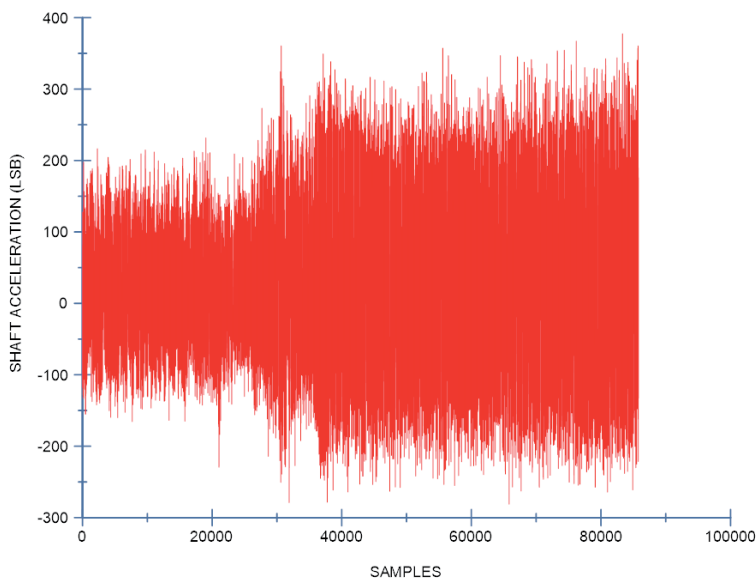


Figure 3.
Shaft Acceleration Acquired by IoT Gateway – Standalone Condition.

Observed Peak Values			Classification Configuration	
Maximum Amplitude	Minimum Amplitude	Range	No. of Classes	Class Width
myRIO				
368 LSB	-286 LSB	654	12	54.5
IoT Gateway				
377 LSB	-281 LSB	658	12	54.83

Table 1.
Measured Values and Classification Configuration - Standalone Starting to No Load Speed

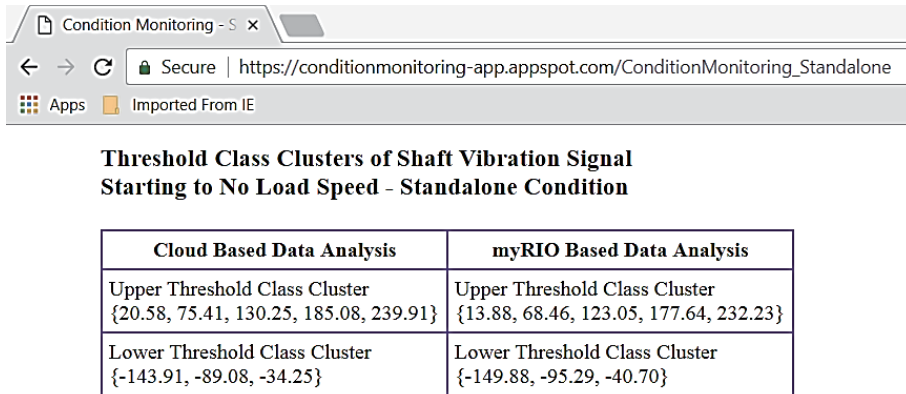


Figure 4.
Adaptive Threshold Class Clusters – Standalone Condition

template, which can be accessed through the cloud environment by specifying the URL: “https://conditionmonitoringapp-appspot.com/ConditionMonitoring_Standalone” and the results are displayed in the form of HTML as shown in the **Figure 4**. Thus, the occurrence of faults or any abnormality at this operating condition can be diagnosed precisely with the shift in the oscillation percentages of the denser class regions and threshold class clusters from the predetermined values.

3.6.2 From starting to no load speed with external disturbance

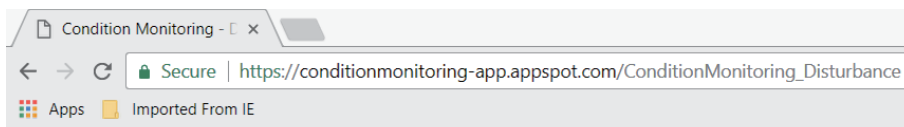
Similar to the standalone mode, the experiment has been carried out with an external disturbance using IoT and myRIO based data acquisition systems and the

Observed Peak Values			Classification Configuration	
Maximum Amplitude	Minimum Amplitude	Difference	No. of Classes	Class Width
myRIO				
453.6 LSB	-296.3 LSB	749.90	14	54.01
IoT Gateway				
460.19 LSB	-296.08 LSB	756.27	14	54.01

Table 2.
Measured Values and Classification Configuration - Starting to No Load Speed under External Disturbance.

shaft vibration pattern has been examined. While performing analysis on the vibration signals with mechanical disturbance, the observed peak values of the vibration signal and the configuration settings made for analysis are listed in **Table 2**.

Using the resulted transition matrix, the oscillations existing between every class and its lower classes are calculated and the dominant classes with more percentage of oscillations measured during the presence of external disturbance have been identified to form the upper threshold class cluster. To form the lower threshold class cluster, every class of the upper threshold class cluster that has made 65 percent or more number of oscillations cumulatively with its lower classes are considered and the results are rendered to the “Disturbance Condition” template, which can be accessed through the cloud environment by specifying the URL: *https://conditionmonitoringapp-appspot.com/ConditionMonitoring_Disturbance* and the results are displayed in the form of HTML as shown in **Figure 5**. There are changes in the vibration pattern due to the external disturbance and it is observed from the results shown, the fact of fixing adaptive threshold for a machine when exposed to external disturbances at a particular operating condition. The severity of the disturbance can be observed by measuring the range of shifts from the limits of the upper and lower threshold class clusters estimated during standalone condition.



Threshold Class Clusters of Shaft Vibration Signal Starting to No Load Speed - Disturbance Condition

Cloud Based Data Analysis	myRIO Based Data Analysis
Upper Threshold Class Cluster {1.03, 55.05, 109.06, 163.08, 217.10}	Upper Threshold Class Cluster {-1.68, 51.88, 105.46, 159.03, 212.61}
Lower Threshold Class Cluster {-161.03, -107.01, -52.99}	Lower Threshold Class Cluster {-162.41, -108.84, -55.26}

Figure 5.
Adaptive Threshold Class Clusters – Disturbance Condition.

3.6.3 Load Changes at Standalone Condition

The shaft acceleration i.e., the vibration signals acquired by the myRIO and IoT based cloud applications during the load changes made at standalone running condition of the DC machine are shown in **Figures 6** and **7**. The observed characteristics and analysis settings used in statistical classical algorithm for condition monitoring analysis in both cases are presented in the following **Table 3**.

The results of upper and lower threshold class clusters obtained from the implementation of the signal decomposition algorithm are rendered to the “LoadingCondition” template, which can be accessed through the cloud environment by specifying the URL: *https://conditionmonitoring-app.appspot.com/ConditionMonitoring_Loading* and the results are displayed as shown in **Figure 8**.

The analysis carried out using the statistical classification based signal decomposition technique for different machine operating conditions is based on the machine vibrations occurring within the permissible limits. The consistency in the

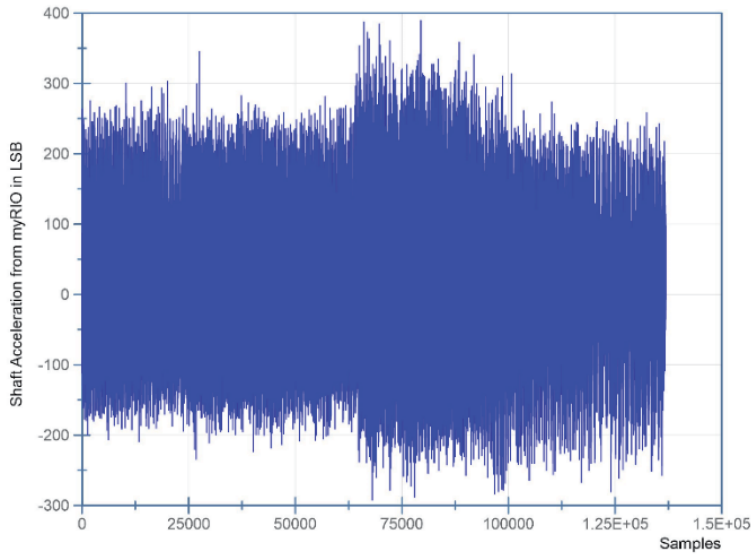


Figure 6.
Shaft Acceleration acquired by myRIO during Loaded Condition.

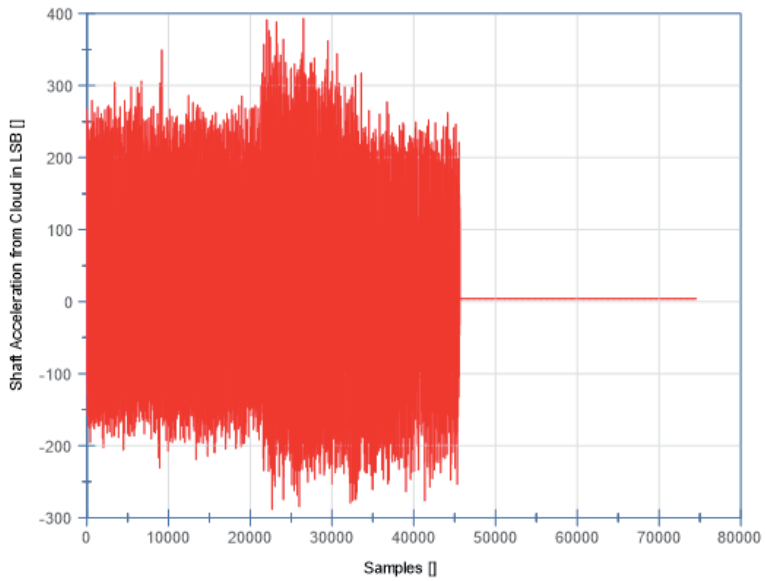
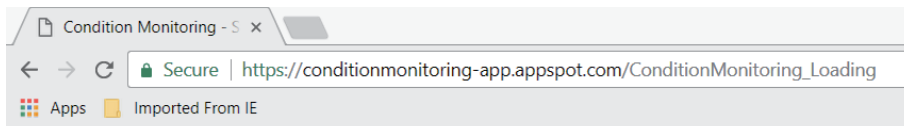


Figure 7.
IoT based acquisition of Shaft Acceleration during Loaded Condition.

Observed Peak Values			Classification Configuration	
Maximum Amplitude	Minimum Amplitude	Difference	No. of Classes	Class Width
myRIO				
389 LSB	-292 LSB	681.00	13	52.3
IoT based Cloud Application				
393.51 LSB	-288.49 LSB	682.00	13	52.4

Table 3.
Measured Values and Classification Configuration – Load Changes.



Threshold Class Clusters of Shaft Vibration Signal During Load Changes - Standalone Condition

Cloud Based Data Analysis	myRIO Based Data Analysis
Upper Threshold Class Cluster { 0.08, 52.54, 105.01, 157.48, 209.94}	Upper Threshold Class Cluster {-3.79, 48.67, 101.13, 153.60, 206.07}
Lower Threshold Class Cluster {-157.32, -104.86, -52.39}	Lower Threshold Class Cluster {-161.18, -108.72, -56.25}

Figure 8.
Adaptive Threshold Class Clusters – Loaded Condition.

total number of oscillations and marginal deviations in the threshold class clusters reveal the flexibility of choosing any of the desired classification criteria during real time implementation. Contrary to the currently adhered thresholds, the adaptive cluster based thresholds possess the significance of tracking the condition of the machine without demarcating the naturally occurring vibration variations as crossing limits during condition monitoring. Hence, this technique avoids false identification of failures caused due to incorrect thresholds and also identifies the ignored failures. This cloud based analysis of random vibration signal is a perception that has been believed to offer better alternative for deriving decisions for efficient condition monitoring when other analysis techniques find challenges in bringing out precise and faster solutions for condition monitoring under dynamic conditions.

3.7 Results and discussion

The DC motor shaft vibration pattern has been examined by acquiring the vibration signal through IoT2040 gateway using Python interface and myRIO using LabVIEW interface considering the same machine operating conditions. The vibration signals acquired by various data acquisition devices have been analyzed by the statistical classification based signal decomposition algorithm considering three different modes of application platform as given below:

- i. An independent vibration analysis application in LabVIEW platform (myRIO based analysis)
- ii. LabVIEW application integrated with IoT service (IoT based analysis)
- iii. LabVIEW application hosted as a cloud service in Google Cloud Platform (Cloud based analysis)

The upper and lower threshold class clusters of DC motor’s shaft vibration determined using the above applications are furnished in **Table 4** which define the scope of the amplitude levels between which majority of the shaft vibrations oscillate during the specified operating conditions. In either case of analysis, the proposed algorithm has uniformly brought out the changes that had happened in the vibration pattern and upholds the fact of fixing thresholds adaptive to the operating

Starting to No Load Speed (Standalone Condition)			Starting to No Load Speed (Disturbance Condition)			Loading (Standalone Condition)		
Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis	Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis	Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis
Upper Threshold Class Cluster								
{	{	{	{	{	{	{	{	{
20.58	15.20	13.88	1.03	0.03	-1.68	0.08	-0.76	-3.79
75.41	69.79	68.46	55.05	53.67	51.88	52.54	52	48.67
130.25	124.37	123.05	109.06	107.32	105.46	105.01	104.76	101.13
185.08	178.95	177.64	163.08	160.96	159.03	157.48	157.53	153.60
239.91	233.54	232.23	217.10	214.60	212.61	209.94	210.30	206.07
}	}	}	}	}	}	}	}	}
Lower Threshold Class Cluster								
{	{	{	{	{	{	{	{	{
-143.91	-148.54	-149.88	-161.03	-160.89	-162.41	-157.32	-159.07	-161.18
-89.08	-93.95	-95.29	-107.01	-107.25	-108.84	-104.86	-106.30	-108.72
-34.25	-39.37	-40.70	-52.99	-53.60	-55.26	-52.39	-53.53	-56.25
}	}	}	}	}	}	}	}	}

Table 4. Adaptive Threshold Clusters Identified by Different Analysis Methods.

Starting to No Load Speed (Standalone Condition)				Starting to No Load Speed (Disturbance Condition)				Loading (Standalone Condition)			
Cloud Based Analysis		IoT Based Analysis		Cloud Based Analysis		IoT Based Analysis		Cloud Based Analysis		IoT Based Analysis	
D	%	D	%	D	%	D	%	D	%	D	%
	D		D		D		D		D		D
6.7	1.0	1.32	0.2	2.71	0.4	1.71	0.2	3.87	3.03	0.6	0.4
7.0	1.1	1.33	0.2	3.17	0.4	1.79	0.2	3.87	3.33	0.6	0.5
7.2	1.1	1.32	0.2	3.6	0.5	1.86	0.2	3.88	3.63	0.6	0.5
7.4	1.1	1.31	0.2	4.05	0.5	1.93	0.3	3.88	3.93	0.6	0.6
7.7	1.2	1.31	0.2	4.49	0.6	1.99	0.3	3.87	4.23	0.6	0.6

Table 5. Deviations of Adaptive Threshold Clusters of Same Operating Condition from myRIO Based Analysis.

condition. The values of deviation between the deduced threshold levels considering the same and various operating conditions are enlisted in **Tables 5** and **6**. The margin of deviations between the threshold class clusters specific to an operating condition determined using IoT [14] and cloud based analysis methods with reference to the amplitude range of vibration signal measured using myRIO (**Table 5**) are observed to be negligible as compared to the deviation between threshold class clusters identified by any particular analysis method for different operating conditions (**Table 6**). In each of the cases considered, the incipient faults or abnormalities during any of the operating conditions can be diagnosed precisely by analysing the margin of deviations in the threshold class clusters. The difference in the deviation values observed from **Tables 5** and **6** reveals that the threshold class clusters obtained by the analysis of data acquired either from IoT device, cloud

Disturbance Condition (Deviation D)			Loading at Standalone Condition (Deviation D)		
Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis	Cloud Based Analysis	IoT Based Analysis	myRIO Based Analysis
20	15	16	21	16	18
20	16	17	23	18	20
21	17	18	5	20	22
22	18	19	28	21	24
23	19	20	30	23	26

Table 6.
Deviations of Threshold Class Clusters Between Different Operating Conditions for specific method of analysis

service or from myRIO device do not lead to incorrect decisions and tends to recognize the change of operation conditions without ambiguity.

In spite of some minor deviations in the threshold values determined for every operating condition during different methods of analysis, the substantial differences in the threshold class clusters identified due to the change of operating conditions in all the analysis methods validate also the fact that the incipient faults or abnormalities (during any of the operating condition or change in the operating conditions) can be diagnosed without ambiguity using the proposed statistical classification algorithm integrated with the cloud environment.

In this work, the actual machine condition is monitored in online mode by continuous acquisition of vibration signals and simultaneous estimation of the threshold levels at different operating conditions using developed cloud services of data acquisition and data processing. The machine's vibration signature will not change over a period of time until some disturbance or loading has occurred. During abnormal conditions, the vibration amplitude may increase or decrease from the normal value. Integrating the actual deviation from the mean values of the upper and lower threshold clusters with historical data for the same or similar machine under the same operating condition facilitates to evaluate the current condition of the machine and hence to schedule suitable maintenance action.

It is observed from the analysis that the vibration threshold class clusters identified for any particular operating condition cannot be maintained as an alarm for monitoring of machines at all conditions to decide appropriate maintenance schedules. Further the analysis reveals the fact, a neighboring machine running at constant speed of 1500 rpm creates a vibration effect equivalent to that of loading in other machines which substantiates the fixation of the operational limits for mechanical and electrical loading, speed and torque capacity etc., during machine maintenance.

The consolidated results obtained from the implementation of the proposed algorithm under unequal and equal classification criteria in local and cloud platforms illustrate the reliability of the analysis technique in yielding consistent results. As illustrated above, the characteristic of the technique to extract the signal features in such a way that they identify the unremarkable disturbances in the operating conditions, and effects of change of operating conditions enable fixation of cluster based thresholds adaptive to machine operating conditions rather than fixed and ambiguous threshold levels. Moreover, the consistency of the observed threshold values under any of the classification parameters ensures the efficiency of the technique. The technique employed on shaft vibration signal at different operating

modes has identified ignored disturbances, intensity and characteristics of such disturbances and incipient changes in the operational behavior both in the on premise analysis and cloud based analysis. Thus the proposed IoT based cloud deployment will help to prescribe the operational constraints for machines in real time applications so that the machine can deliver improved performance and have extended lifetime.

4. Conclusion

In either case of analysis based on myRIO, IoT device or Cloud based, the investigation uniformly brings out the changes that had happened in the vibration pattern and upholds the fact of fixing thresholds adaptive to the operating condition. The integration of vibration sensors and actuators through Python and LabVIEW interfaces with cloud in real time ascertains generic, interoperable and ubiquitous computational nature of the model for implementation of effective condition monitoring. In this research, the convergence of cloud and IoT technologies for analysis of real-time systems has been brought into implementation for condition monitoring of electrical machines to support scalable and interoperable data exchange with features of flexible and collaborative analytics, fixation of adaptive alarms with contextual thresholds and control of multiple machines in real-time operating environment. The model gives collaborative access to machine data from any geographical location for analysis and decision making.

In summary, the cloud-based vibration monitoring model implemented in Google Cloud Platform offers services for

- Data acquisition from the sensors mounted on the shafts of the DC motors.
- Data storage to store the enormous amount of acquired signal data from multiple sensors.
- Data classification of vibration signals for effective statistical analysis to estimate adaptive cluster of thresholds and
- Decision making for condition assessment.

These services have been offered on demand over the Internet to utilize the reliable service of the machines in a persistent way. The computational engine, which is included in the model performs inherent statistical analysis of the vibration signals to estimate the cluster of thresholds adaptive to various operating conditions. The services have been deployed without any limitation in a cloud environment and the industrial applications can share information using the deployed services from anywhere on demand basis. The threshold values estimated using cloud services are compared with that of the vibration analysis carried out on the machine beds locally using myRIO for data acquisition in LabVIEW ensures the integrity of the cloud-based model with assured scalability.

List of nomenclature

IoT	Internet of Things
RIO	Reconfigurable Input Output
ISO	International Organization for Standardization

SCADA	Supervisory Control and Data Acquisition
PLC	Programmable Logic Controller
VPN	Virtual Private Network
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
GCP	Google Cloud Platform
MVT	Model-View-Template
PaaS	Platform as a Service
SaaS	Software as a Service
IaaS	Infrastructure as a Service
API	Application Programming Interface
.tdms	Technical Data Management Streaming format
SDK	Software Development Kit
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
URL	Uniform Resource Locator
XML	eXtensible Markup Language
SQL	Structured Query Language
SDK	Software Development Kit
FPGA	Field-Programmable Gate Array
RT	Real Time
g	Acceleration due to gravity (9.81 m/s ²)
LSB/g	Least Significant Bit per g
% D	Percentage Deviation

Author details


Ganga Dhandapani^{1*} and V. Ramachandran²

¹ Department of Electrical and Electronics Engineering, NIT Nagaland, Dimapur, India

² Department of Computer Science and Engineering, DMI College of Engineering, Chennai, India

*Address all correspondence to: gangaadhan@gmail.com

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Diego Galar, Peter Sandborn, Uday Kumar, Carl-Anders Johansson, S.: SMART: Integrating Human Safety Risk Assessment with Asset Integrity. In: Giorgio Dalpiaz, Gianluca D'Elia, Riccardo Rubini, Marco Cocconcelli, Fakher Chaari, Mohamed Haddar, Radoslaw Zimroz, Walter Bartelmus (eds.), Proceedings of the Third International Conference on Condition Monitoring of Machinery in Non-Stationary Operations CMMNO 2013, LNME, Springer; 2014. p. 37–59
- [2] Kirubashankar, R., Krishnamurthy, K., Indra, J., Vignesh, B: Design and Implementation of Web Based Remote Supervisory Control and Information System. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2011:1-4:2231-2307
- [3] Larry Combs: Cloud Computing for SCADA. White Paper from InduSoft.2011
- [4] Omid Givehchi, Henning Trsek, Juergen Jasperneite: Cloud Computing for Industrial Automation Systems - A Comprehensive Overview. In: Proceedings of the IEEE 18th Conference on Emerging Technologies and Factory Automation; 2013.p.1-4.
- [5] Omid Givehchi, Juergen Jasperneite: Industrial Automation Services as Part of the Cloud: First Experiences. In: Proceedings of the Jahreskolloquium Kommunikation in der Automation – KomMA; 2013.
- [6] Omid Givehchi, Jahanzaib Imtiaz, Henning Trsek, Juergen Jasperneite: Control-as-a-Service from the Cloud: A Case Study for using Virtualized PLCs. In: Proceedings of 10th IEEE Workshop on Factory Communication Systems; 2014.p.1-4
- [7] Francisco Macia-Perez, Jose Vicente Berna-Martinez, Diego Marcos-Jorquera, Iren Lorenzo-Fonseca, Antonio Ferrandiz-Colmeiro: A New Paradigm: Cloud Agile Manufacturing. International Journal of Advanced Science and Technology. 2012:45:47-53.
- [8] Bureau of Indian Standards IS 12075, “Mechanical Vibration of Rotating Electrical Machines with Shaft Heights 56 mm and Higher - Measurement, Evaluation and Limits of Vibration Severity”, [ETD 15: Rotating Machinery], 2008
- [9] National Instruments, White Paper: Addressing Challenges of Fleetwide Monitoring. 2016. Available: download.ni.com/evaluation/crio/24346_Fleetwide_Monitoring_WP_IA.pdf [Accessed: 2020-12-29]
- [10] Jim Montague. Flexibility Driving the Future of Automation – Flexible Production, Cloud Computing Pave the Way to Future for Rockwell Automation and User, Automation Fair - A special report. Control and control design [Internet]. 2012. Available : <https://www.controlglobal.com/articles/2012/pauto-fair-17/> [Accessed: 2020-12-29]
- [11] Advantech Technical White Paper. Cloud-Based SCADA as an IIoT Gateway. 2015. Available: <https://silotips.com/download/cloud-based-scada-as-an-iiot-gateway>, p.1-11 [Accessed: 2020-12-29]
- [12] Steve Lacey. From local to cloud-based condition monitoring. Quarry Management [Internet]. 2017. Available: <https://www.engineerlive.com/content/local-cloud-based-condition-monitoring> [Accessed: 2020-12-29]
- [13] Mallikarjun Kande, Alf J. Isaksson, Rajeev Thottappillil, Nathaniel Taylor: Rotating Electrical Machine Condition Monitoring Automation—A Review. Machines. 2017:5-4:24. DOI : <http://dx.doi.org/10.3390/machines5040024>

- [14] D. Ganga, V. Ramachandran: IoT based Vibration Analytics of Electrical Machines. IEEE Internet of Things Journal. 2018:5-6:4538 –4549. ISSN: 2327-4662, DOI.: 10.1109/JIOT.2018.2835724
- [15] D.Ganga, V. Ramachandran: Dynamic analysis of Vibration Signals and Adaptive Measures for effective Condition Monitoring of Electrical Machines. International Journal of Condition Monitoring and Diagnostic Engineering Management. 2018:21-2: 29-38. ISSN 1363 – 7681, 2018
- [16] Balasingh Moses. M, Ramachandran. V, Lakshmi. P: Cloud Services for Power System Transient Stability Analysis. European Journal of Scientific Research. 2011:56-3:301-310.
- [17] Google Cloud. Running Django on App Engine Standard Environment [Internet]. 2018. Available: <https://cloud.google.com/python/django/appengine> [Accessed: 2020-12-29]
- [18] DIAdem 2017, Part Number 370858N-01, National Instruments [Internet]. 2017. Available: <http://zone.ni.com/reference/en-XX/help/370858N-01/> [Accessed: 2020-12-29]
- [19] National Instruments. NI myRIO-1900 User Guide and Specifications [Internet]. 2013-2016. Available: <https://www.ni.com/pdf/manuals/376047c.pdf> [Accessed: 2020-12-29]
- [20] Siemens. SIMATIC IOT SIMATIC IOT2020 SIMATIC IOT2040, Operating Instructions Manual - A5E37656492-AB [Internet]. 2016. Available:https://support.industry.siemens.com/cs/attachments/109741658/iot2000_operating_instructions_e_en-US.pdf [Accessed: 2020-12-29]

Compound Cryptography for Internet of Things Based Industrial Automation

J.S. Prasath

Abstract

Internet of things based industrial automation systems are widely used for process monitoring, and control applications. The security threats increase due to the internet is an open environment. This proposed work is the implementation of secure monitoring of plant information through the Supervisory Control and Data Acquisition (SCADA) system. The modified asymmetric and hash algorithm is proposed which generates the large key size of 4096-bit and 512-bit respectively. This proposed security algorithm is implemented using the ARM Cortex A53 processor which performs data encryption and decryption. It provides authentication and integrity of process information across the internet. It achieves a data transfer rate of 300 Megabits per second and more than 95 percent efficiency. This proposed work can be applied for securing the internet-enabled industrial automation process and allows secure monitoring of plant information in remote areas. The security of sensitive process parameters is enhanced through the proposed large key size in asymmetric algorithms. This proposed security algorithm prevents the damage to industrial devices from unauthorized access and modification. It assures the smooth functioning of plant operations and also provides safety to plant operators.

Keywords: Security, industrial networks, SCADA, encryption, decryption, Internet

1. Introduction

Industrial automation plays a major role in real-time data acquisition and control applications. Modern industries depend on vastly more automation and intercommunication. Industrial process equipment is automated to do periodic data collection, event detection, control operation, real-time data acquisition, real-time inventory management, alarming etc. Industrial automation system makes installation flexibility, reduces the repairs costs, disintegration of machine control functions, monitoring the mechanical equipment parameters, error detection and improves the overall efficiency of plant operations. An industrial automation system is a computer system which monitors and controls the various industrial processes such as petrochemical plants, power plants, water treatment plant, oil and gas, food production etc. The behavior of the process changes due to the attack during data communication between devices. Automation devices such as SCADA and Programmable Logic Controller (PLC) does not have inbuilt security mechanisms.

The suitable security algorithm is essential to protect the process equipment and its information from unauthorized access.

The SCADA system is widely used in industrial automation for monitoring and controls the process parameters. It is used for data gathering in a variety of applications such as power generation, petrochemical, sewage and water treatment systems, food and pharmaceutical industry. The monitoring and control of process parameters takes place in remote areas to keep up the steady state of process. SCADA systems include Master Terminal Unit (MTU), Remote Terminal Unit (RTU), network devices and SCADA software. SCADA alerts operators by alarm when conditions become hazardous. The SCADA system includes RTU, and Programmable Logic Controllers (PLC) which collect data from end-point devices like actuators, pumps, or other sensors and control ongoing processes in a plant. The plant sensitive information is transmitted between MTU and RTU that is unsecure and unsafe plant operations. The process data can be accessed and modified by the attackers. The security mechanisms are essential to protect the SCADA system from unauthorized access and to give safety for plant operators.

PLCs are used to control the process parameters and to ensure smooth plant operation. PLC and SCADA system is used together in automation and management of processes in real-time. PLCs are connected to a Human-Machine Interface (HMI) which presents current input and output values to the operators and accepts commands from the user. In SCADA system, RTU provide high processing power, communication capabilities and flexibility as compared to PLCs. The data transmitted from the PLC need to be protected from the attackers. The process data must be encrypted using suitable cryptography and the cipher text is to be transmitted over the internet to ensure confidentiality. The decryption algorithm is to be used at the receiver to get the process data in original plain text. The security policies and security mechanisms are essential for internet enabled industrial automation system.

2. Advances in industrial automation system

The industrial data gathering and monitoring has greatly improved by the wireless standards and internet. The real-time process information can be transmitted through wireless medium and monitored through the internet anywhere in the world. The plant information can be monitored and controlled with the SCADA system through the internet. The control operations and management of sensitive process information are carried out in the master station. Human Machine Interface (HMI) allows operators to read various physical parameters and status of alarm. The general monitoring and supervisory functions are carried out in the corporate networks. The functions of Remote Terminal Unit (RTU) are to monitor the field analog and digital parameters and transmit data to the central control room. RTUs are connected through the remote networks.

The need for security increases due to the integration of industrial networks with Information Technology (IT) networks. Wireless and Internet technologies are essential to monitor and control the process data efficiently. The benefit of wireless technologies in industrial networks provides mobility, to manage substations, and it requires little installation and preservation cost. The control and automation functions can be performed in real-time over the internet by the use of TCP/IP standard in SCADA transmissions. The technological advancement in industrial network operations gives rise to various security risks and challenges in managing IT networks while integrating with both SCADA and corporate networks. The use of Internet in industrial networks creates additional security hazards and safety

issues in the automation system. The major intrusion takes place in communication medium and data modification. The existing industrial automation equipments were not built with security mechanisms. Attackers may create new process information, can alter the process data and capture the physical channels. This leads to failure of process equipments and heavy loss to industries. It is essential to propose the novel security mechanism for secure operation in web-based industrial networks.

3. Review of security issues in SCADA networks

The major technological, operational and organizational changes increase the security problems. Most of the industries focus on improving the security in data communication, safety standards and cost reduction by applying innovative technology design. The standards and regulations of data security have to be applied during design, implementation and execution of the industrial process to ensure adequate safety, consistency and lifecycle effectiveness for all parties involved in the plant operations. Industrial Control System security requires secure management of work flow and policies. The security management involves physical access control, physical intrusion detection etc. It also requires the device security where the hardware, software and firmware need to be protected. The security in communication is another aspect where the message or data need to be protected. The supervisory and control operations are carried out by integrating the SCADA devices with remote web-based networks. Due to the web-based operation, SCADA devices become more vulnerable to various attacks.

Intrusion detection system is one of the software applications which monitors the network activities for violations and produces reports to the management. The status of security is to be monitored and tested by the continuous security assessment in the security management system. Cryptography is used to address important aspects of communication security, such as, message authentication and integrity as well as confidentiality. The hybrid cryptography algorithm is proposed which combines the asymmetric, symmetric and hash algorithms along with the dedicated hardware key all together strengthens the plant information security [1]. This hybrid algorithm provides confidentiality, integrity and ensures privacy in accessing the sensitive process data. It is essential to propose strong security mechanisms for accessing the process information through internet. The highly secure encryption decryption algorithm is proposed which is simple and it can be used for cloud computing-based applications [2]. This algorithm is based on efficient logical operations, such as XORing, addition, and subtraction as well as byte shifting. It allows selecting the secret key length and the number of rounds to generate the cipher text. Key management is the most dynamic field of research in cryptography and there are challenges in the area of industrial plant key management. The critical information such as passwords and encryption keys should be kept confidential due to security concerns in industries.

The industrial process parameters should be protected from unauthorized access during transmission. The security mechanisms are essential for data monitoring, storage and control. An enhanced data security algorithm is proposed to ensure security in the cloud [3]. The SHA-256 hashing and AES encryption algorithms are used to maintain integrity and confidentiality in the cloud. A novel parallel cryptographic algorithm is proposed which overcomes the drawback of symmetric security algorithm and hash algorithm [4]. The analysis was done with respect to computation time. The run time is less as compared to the RSA-MD5 algorithm. The additional layers of hybrid function can be performed to enhance the data integrity and security. A peculiar security protocol is formed to increase the level of security [5]. It increases

the level of security by incorporating MD5 algorithm and combining the AES with RSA algorithms. The encryption and decryption of image files can be performed using the hybrid algorithm. A hybrid cryptographic algorithm is proposed which combine the Blowfish and MD5 hashing algorithm to increase data security in the cloud [6]. The various parameters include file size and execution time is evaluated. It takes less time for encryption and decryption and it occupies less storage space. An innovative identity based hybrid encryption is proposed to increase the security of outsourced data [7]. The encryption is performed using RSA and Elliptic Curve Cryptography (ECC). The data is encoded along with receiver identification. The identity and the keyword are encrypted using Proxy Re Encryption. It achieves efficiency and assures the security of user message. The hybrid cryptography algorithm is proposed which includes symmetric and hash algorithms that ensure confidentiality and integrity of process parameters [8]. It is implemented with the embedded system which enables secure monitoring of plant information over internet.

The Intrusion Detection System (IDS) is essential to preserve the SCADA system from internet attacks. IDS monitor the network activities and host to detect the security threats. The clustering based IDS are proposed to detect the attacks on SCADA systems [9]. SCADA attacks were detected by normal and critical states of process parameters of target system. When the process parameter reaches the critical state, alarms are raised. The criticality scoring algorithm is proposed to determine the state of the target system. The distributed and networked approach of SCADA system increases the cyber-attacks. The major threats are unauthorized access to the control software and network intrusion. The various possibilities of cyber-attacks on SCADA system is evaluated by using two Bayesian attack graph models [10]. The probabilities of the intruder influence the destination is determined by the Bayesian attack graph model. The evaluation results infer that the reliability of the power system becomes less due to the increase in attacks against cyber components and skill levels of attackers. The energy efficient security architecture is proposed for wireless based industrial automation systems [11]. The packet protection based on encryption consumes energy in the case of battery powered devices. The packet based selective encryption is also proposed which reduces energy consumption and detection of attacks. The results infer that the intrusion is difficult to distinguish from normal disruption at industrial operations. A Dynamic Security management mechanism is proposed which reduces security hazard, deadline miss ration and process elimination ratio of discontinuous actual process compiling on server systems [12]. The time and power utilization of extensively used security mechanisms are measured. A security hazard measures is introduced which quantifies the strength of security in real-time operations. A dual-level feedback control scheme is designed to notify the task scheduling issues. The future work includes proposal of security assessment for shared control in enterprise networks and integrity protection. A multilayer cyber-security scheme is proposed which is based on Intrusion Detection System (IDS) for safeguarding SCADA in smart grids [13]. In this work, external malicious attack is identified by a SCADA-specific IDS technique. A cyber security test-bed used to investigate vulnerabilities and hybrid intrusion detection approach is implemented in a SCADA system. The test-bed is the setup of grid connected solar panel based SCADA system in real-time. This proposed multi-attribute SCADA-IDS provides early alert, intrusion detection and prevention and abnormal behaviors in SCADA based automation system. A key management scheme is evaluated which includes session and master key updates [14]. The master station is responsible for producing the session keys. The Elliptic Curve Diffie-Hellman protocol is used in the master key update phase. This scheme of key management supports the MODBUS implementation with the required speed, greater efficiency and achieves high degree of security in SCADA communication.

The cryptography is essential for secure communication of plant information through SCADA networks. The characteristics of cryptographic algorithms are analyzed in terms of energy and time related for embedded real-time systems [15]. The analysis indicates that energy consumptions of security algorithms are non-linear to the size of the plain text. The energy cost is proportional to the run time of security algorithm with variable data size. Based on this analysis, the application of cryptographic algorithms can be extended in embedded real-time applications. The security issues in Industrial Automation and Control System (IACS) are analyzed which includes risk assessment, countermeasures, validation and monitoring of results [16]. The analysis ensures the satisfied security level can be achieved for a distributed industrial system. The efficient security management solutions will become tough due to the complexity and size of IACS. It is essential to propose advanced mechanisms to support IACS security.

A network filtering approach is proposed for the detection and mitigation of cyber-attacks [17]. It is based on the packets analysis of communication between master and slaves of SCADA system and monitoring the state of the protected system. The benefit of this proposed work is that it provides less number of negative results. A Critical State Analysis and State Proximity for detection of intrusion are proposed for SCADA systems [18]. A multidimensional metric approach is introduced which provides the measurement related to the length between a critical state and the given states. The unique security issues in electric power system are addressed which is based on SCADA Networks [19]. The SCADA system is secured by using symmetric encryption. The master station takes the Key Distribution Center (KDC) and it initiates the communication. The slave station includes security devices which generate the session key, perform the key encryption with the master key and transmit it to the equipment on the master station.

The trust system is proposed which perform active security analysis and response in order to increase the security of SCADA systems [20]. The status information delivery, issue of network node commands, packet delivery analysis in various protocols and arrangements are performed by the trust system.

The key management architecture is proposed for SCADA System that requires less number of keys stored in a RTU [21]. It reduces the operational cost for group communication. Group link is attained by using the key hierarchy configuration. The Master Terminal Unit (MTU) is able to send the information between a Sub-Master Terminal Unit and Remote Terminal Unit. In this proposed key structure, two classes of communication which includes communication between MTU and Sub-MTUs and between Sub-MTUs and RTUs. The impact of traditional Information and Communications Technologies (ICT) malware is focused on SCADA systems [22]. The experimental test-bed which includes software toolkit called MAISim (Mobile Agent Malware Simulator). MAISim agent class is used for simulation of malware. The vulnerabilities exist in the SCADA systems due to network connections, access control, protocols and software. A vulnerability estimation scheme is proposed to estimate the susceptibility of SCADA systems in terms of access points [23]. This work quantifies the potential impact on causes of attack. The method used in this work is to assess the losses in power system and computer networks susceptibility due to cyber-attack.

4. Overview of existing security mechanisms

The security is a major concern for industrial operations and the process plant information should be protected from unauthorized access. The existing security mechanisms are adopted for intrusion detection, cyber-attacks, risk management, data

Algorithm	Key size	Block size	Rounds	Encryption Speed	Security
AES	128, 192, 256 bits	128 bits	10, 12, 14	Fast	Considerably Secure
DES	56-bits	64 bits	16	Very Slow	Inadequate Security
3 DES	112-bits	64 bits	48	Very Slow	Adequate Security
RC2	8–128 bits	64 bits	18	Fast	Vulnerable
RC5	2040 bits	128 bits	255	Fast	Considerably Secure
Blowfish	32–448 bits	64 bits	16	Fast	Vulnerable
Proposed Algorithm (RSA and SHA)	4096-bits and 512-bits	470 bytes, 1024 bits	80	Fast	Highly Secure

Table 1.
Comparison between standard and proposed cryptography algorithms.

protection by cryptography, network firewall etc. The security threats increase due to process monitoring and control through internet. It is essential to ensure process data security and privacy in accessing the plant information in the automation system.

Table 1 shows the existing security mechanisms, its advantage and disadvantage. It is identified that there is a large number security issues arises due to the integration of SCADA Network with the Information Technology Networks. Traditional ICT countermeasures cannot provide complete protection to SCADA systems. Conventional Security mechanisms are not suitable to handle the new security problems. Even though the varieties of security mechanisms are proposed, still there is a lack of security in the modern industrial automation systems. It is essential to propose efficient and less complex security algorithm to secure the data communication takes place between SCADA Networks.

The existing security mechanism for SCADA networks are related to hybrid encryption, intrusion detection, key management, and packet based encryption etc. The lack of strong dynamic security management mechanisms exists related to cryptography for securing SCADA systems. The SCADA system deals with remote monitoring and control of sensitive process parameters. The strong cryptographic algorithm is essential to protect the process information and equipment from unauthorized access. The existing security mechanisms and algorithms are inadequate to achieve strong security. The attackers can easily capture the process data, modifies it and retransmit to the destination. The security attack leads to failure of process instruments, major losses to the management and unsafe working condition to operators. The hybrid security algorithm is proposed to secure the plant parameters in wastewater treatment process across the internet [24]. It includes symmetric and secure hash algorithm to protect the wastewater parameters from unauthorized access and modification. It is essential to implement the protocols for secure data transmission in embedded system with wireless networks.

This proposed work focuses on securing the process information by incorporating modified asymmetric and hash algorithms. It ensures secure monitoring of plant information in real-time applications. It combines the asymmetric encryption and hash algorithm which provides data confidentiality and integrity. The large key size of 4096-bits is generated using asymmetric encryption which is not exists in the previous work and it enables secure transmission and monitoring of process information through the internet.

5. Proposed compound cryptography algorithm

The temperature and gas process data is secured by performing hybrid cryptographic algorithm which includes modified asymmetric encryption and hash algorithm. The public and private keys are generated in the asymmetric algorithm to perform data encryption and decryption in order to ensure data confidentiality.

5.1 Key generation

5.1.1 Modified asymmetric algorithm

Asymmetric algorithm involves usage of public key and private key. The public key is used for encryption of process data and the private key is used for decryption of process data. It enhances the security level of sensitive plant information due to the usage of two keys.

The various steps involved in asymmetric algorithm are given below. These include

- Generation of public and private keys
- Encryption
- Decryption

5.1.1.1 Algorithm steps

- Select two different prime numbers: i and j
- Calculate $s = i*j$
- Calculate $g(s) = (i-1)(j-1)$
- Select integer 'd' such that $\gcd(g(s)) = 1$; $1 < d < g(s)$
- Calculate $e, e = d^{-1}(\text{mod}(g(s)))$
- Public key, $PU = (d,s)$
- Private key, $PR = (e,s)$

5.1.2 Encryption

After generation of public and private keys, encryption is performed to convert the raw input data into cipher text that is., unreadable format. The encryption of plant information is performed at the transmitter. The encrypted data is transmitted across internet.

Assume that the original input is denoted by 'T'. The cipher text is obtained by the formula given below.

- Original text: T
- Cipher text: $C = T^d \text{ mod } s$
where d – Public key.

5.1.3 Decryption

The cipher text is obtained at the receiver and performs decryption. It converts process data in unreadable format to original plain text. The original plain text is obtained by the formula given below.

- Original text: $T = C^e \text{ mod } s$
where e – Private key.

The large key size of 4096-bit is generated in this proposed modified asymmetric algorithm. The hash algorithm is proposed which generates different hash value in order to ensure data integrity. The SHA (Secure Hash Algorithm)-512 generates intermediate hash value using the message block as key. The block size is 1024-bits, the word size is 64-bits and the number of rounds is 80. The SHA-512 algorithm is highly secured as compared to the MD5 (Message Digest) algorithm.

5.2 Secure hash algorithm (SHA-512)

The SHA hash function converts input value of approximate to a constant length. The hash is smaller than the input data and it is a tiny representation of a big data which is referred to as digest. The hashing algorithm involves processing of hash function and each block size varies depending on the algorithm. The capacity of the block varies from 128-bits to 512-bits. It involves round function in which each round takes an input of a uniform size, typically merging of the latest information block and the result of the last round. The modified SHA1 algorithm is developed which expands the hash value from 160-bits to 1280-bits [25]. It is achieved by allocating four buffer registers in each round inside the compression function for eight times. This hash value was not hacked against brute force attack. The hash algorithm protects the password storage and it is used to check the data integrity.

5.2.1 Description of SHA-512

The input message is padded first to obtain the block size of 1024-bits. The message schedule is generated to process the 1024-bit block size of the input message. It consists of eighty 64-bit words. The first 16 words are directly obtained from the 1024-bit message block. The remaining words are generated by performing permutation and mixing functions to the previously generated words. The modified asymmetric and hash algorithm is proposed that generates large key size of 4096-bit and 512-bit respectively [26]. It provides authentication and integrity of process information across internet. Authentication is essential to ensure the plant information is accessed and controlled by the authorized users.

The message block consists of two inputs which are 512-bit hash buffer and the 1024-bit message block. The hash buffer contents are processed along with the input which is called round function. The round function is to be performed for each block of 1024-bit input message. The eighty rounds are to be carried out for each message block. The eightieth round output is added to the hash buffer contents at the starting of the round process. This addition is performed for each 64-bit word of the output. The message digest is obtained from the content of hash buffer which is the processing of all N-message blocks. The key generation and encryption algorithm is proposed for ensuring privacy in Mobile Ad-Hoc Networks [27]. This key generation algorithm adds scrambling factors to generate random key sequences with essential length but incurred low execution overhead, whereas the encryption/decryption algorithm utilizes the One Time Pad (OTP) system by

adding scrambling factors for data confidentiality which satisfies the randomness, diffusion, and confusion tests.

Figure 1 shows the generation of message digests of SHA 512 algorithm. The input message is first divided into block of 1024-bits long. The messages of each 1024-bit block are denoted by $M(1), M(2) \dots M(N)$. The message blocks are processed one at a time, starting with a fixed initial value $H(0)$, sequentially compute

$$H(i) = H(i-1) + C_M(i)(H(i-1))$$

where C – Compression function.

Figure 2 shows the processing of single 1024-bit block. The message schedule array has eighty 64-bit words. Each 1024-bit block is performed with 80 rounds to generate hash value.

Figure 3 shows round function of SHA-512 hash algorithm. The intermediate output is generated which is equivalent to the addition of modulo 2^{32} sum of.

The following quantities are performed logical XOR operation.

- Rotation of block towards right by 14 places
- Rotation of word towards right by 18 places
- Rotation of word towards right by 41 places

The additional quantities are also appended with the eighth word in the block modulo 2^{64} :

The following quantities are performed with logical XOR operation.

- Rotation of the first word in the block towards right by 28 bits

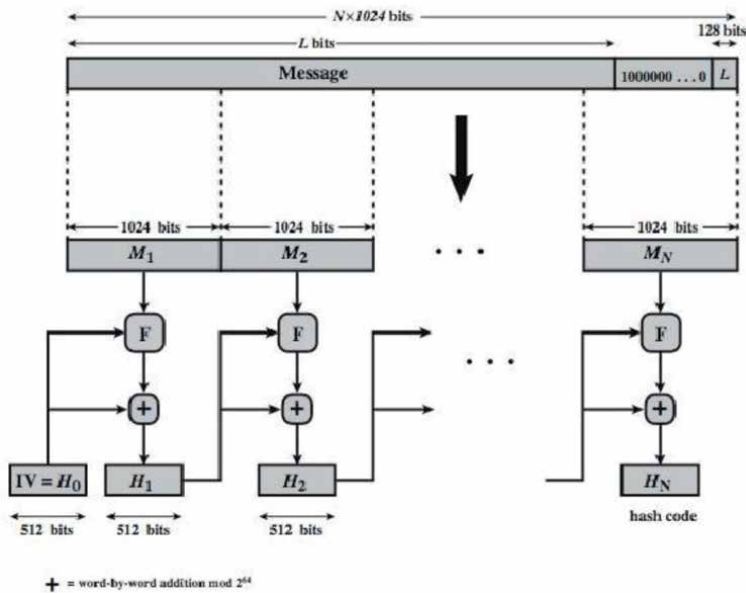


Figure 1.
 SHA-512 for generation of message digest.

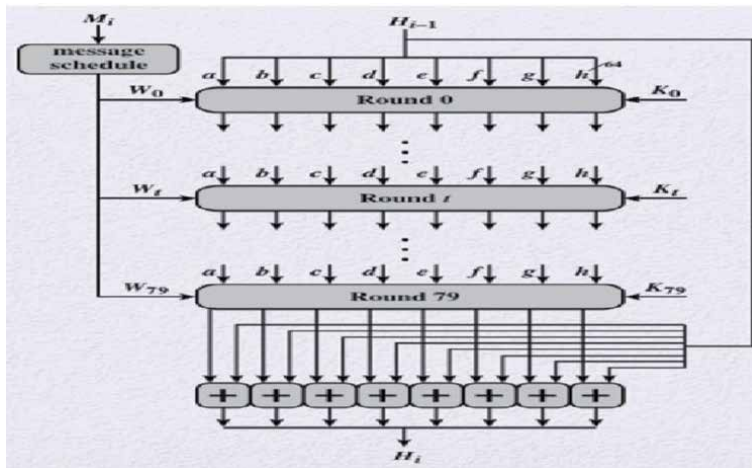


Figure 2.
Processing of SHA-512 single 1024-bit block.

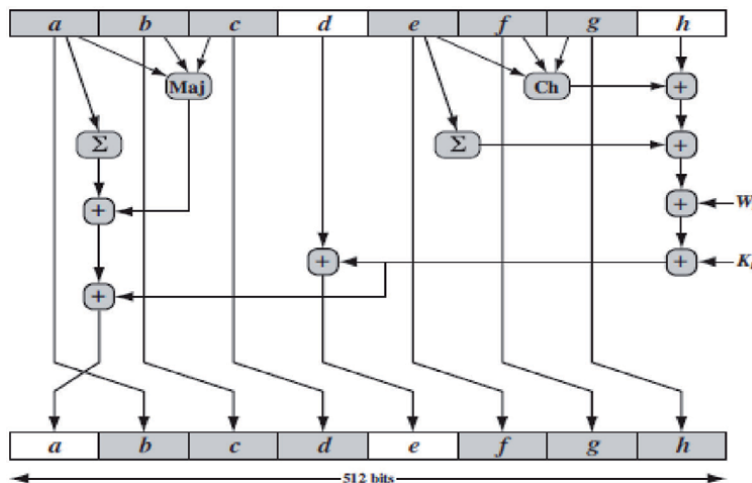


Figure 3.
SHA-512 round function.

- Rotation of word towards right by 34 bits
- Rotation of word towards right by 39 bits

Finally, each of the eight words of the block that will ultimately become the hash is moved to the position of the next word in the block, with the first word in the block being replaced by the modified eighth word in the block.

The first step is to perform modified asymmetric encryption using public key. The SHA-512-bit block cipher algorithm is performed to generate hash value. The hash algorithm ensures IP security and data integrity. The process data in cipher text is transmitted across the internet.

This proposed work uses large key size of 4096-bit in the modified asymmetric algorithm and the number of rounds can be varied. It performs data encryption at very high speed. This proposed hybrid cryptographic algorithm achieves higher level of data security. It can be applicable for securing the

Authors	Algorithm	Key size	Block size	Rounds	Security
Vikas K.Soman [2017]	AES, ECDSA, SHA-256	128, 256 bits	128 bits	10, 12, 14	Medium Security
Adviti Chauhan [2017]	Blowfish, MD5	32–448 bits	64 bits	16	Medium Security
M. Harini [2017]	AES, RSA, MD5	128, 1024 bits	128 bits	10	Medium Security
Anushka Gaur [2017]	Blowfish, MD5	332–448 bits	64 bits	16	Medium Security
Prabukanna [2016]	RSA, ECC	1024 bits, 256 bits	128 bits	—	Highly Secure
Proposed Modified Compound Cryptography algorithm	RSA and SHA	4096-bits and 512-bits	470 bytes, 1024 bits	80	Highly Secure

Table 2.
 Comparison between existing and proposed cryptography algorithms.

sensitive plant information in industrial applications. The cipher text is received through the internet. The modified asymmetric decryption is performed using 4096-bits private key at the receiver. The key length is a major factor in securing the sensitive process data. The larger key size ensures that the brute force attack is infeasible. The process data in original numerical form is monitored through the SCADA system.

Table 1 shows the comparison between standard and proposed cryptographic algorithms. As compared to standard algorithms, the large key size as well as block size is generated in the proposed security algorithm. The proposed asymmetric algorithm produces the large key size of 4096-bits that strengthens the security to higher level. The number of rounds used in SHA-512 is 80 for each message block. The size of each message block is 1024-bits long. It achieves high speed of encryption. This proposed algorithm strengthens the level of security. It is suitable for securing highly sensitive plant information in industrial operations.

Table 2 shows the comparison between existing and proposed cryptographic algorithms. The asymmetric algorithm used in the proposed work generates large key size and provides authentication. The hash algorithm is also used which ensures data integrity. The key size of the existing security algorithms is low and the key size is increased in this work. The number of rounds also increased during the process of encryption. This proposed work uses one key for encryption and another key for decryption.

6. Implementation of embedded based secure process monitoring through SCADA system

Figure 4 shows the transmission of temperature and gas process data in cipher text. The temperature and gas process data is sensed by the sensor and it is transmitted to the embedded system. This process data is encrypted using the embedded system. The hybrid encryption algorithm is proposed which combines the asymmetric encryption and hash algorithm. The encrypted data is transmitted over the internet.

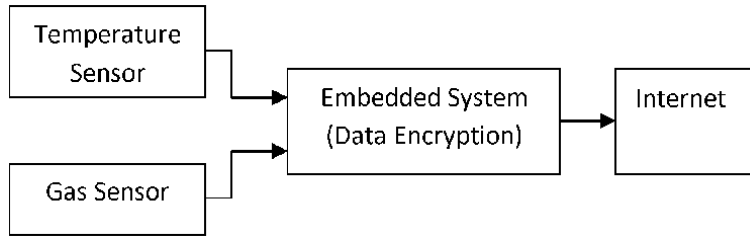


Figure 4.
Transmission of process data using embedded system with internet.

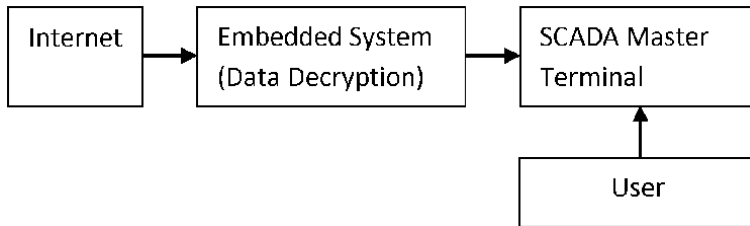


Figure 5.
Reception of process data using SCADA system with internet.

Figure 5 shows the reception of process data in cipher text through internet. The decryption is performed using embedded system and the original data in numerical form is monitored through SCADA master terminal unit.

7. Results and discussion

This proposed modified compound cryptography algorithm is performed using python. This modified asymmetric algorithm generates large key size of 4096-bit and the modified hash function of 512-bit message digest is generated which ensures data integrity over wireless networks. The private key is used only by the receiver to decrypt the process data. The public key is used by the sender to encrypt the process information.

7.1 Generation of private key and public key

The large key size of 4096-bit private key is generated from the modified asymmetric encryption which strengthens the security of sensitive process data. The private key and public key generated from the proposed modified asymmetric algorithm is given below.

-----BEGIN RSA PRIVATE KEY-----.

```

MIIEpAIBAACKCAQEAmS7TPybmKXuzbEGcfQsBuHu2SigegjXbzlrS94ktXN
evH40cpjEGEFUYxX3qoUwJXhTSNb9TnoTRNdL6cgwhdByly07dEM7 + sfK1Jw/
lvLjsZQmYuoIWFjJAmNey55rD/oqkFV6wnpG5O97JJEHjCEDqpqbcUoqmbPBBAUs-
P5yZcvAhKJorhicPajBnN8ZOoYm6pv/1KmVBtNxY/edSKQFUsekbbMvjgkpWcqaB-
bGsR62NWPErK58jUReJrPYI39u + 97yGEEu3Wm2zOXjAqmTX2 + 6Jb1cXC7IMzdZ/
UOQRz9Fw + BdHCIEjRMUktjdQD4BNq5kub4tTAcqU2h6AyUQIDAQABAoIBAG
Pd5P0qdTejG3hM40zvWs7OUAyK0ROi9weqI8q4XeE06q5p8/9qRMY03SqaVNB1It
3khK9Tm/f5KpWUYyhLlxE2oeYEHcyJvFjDgAWRBd23VhjfFzLiwIVv0Jac/lhj + r/
  
```

OVbn3PyOeXacBBo1vuZGKpoTrrI465//ZZAWAk5Uukb9h9CzHCiSQofbx68qXMK/
bXuiWFFGRWSdOSN53eX3j/gm8 + wvWRwYBnahIhgoLIQd8mVwzSoimg-
4sQnAenep7y6a + 0znATQNU1boANn2vDyUHtKLlBLibLBI9fHAycWg3 + nKQ
AUBTFsxvPSBulAFalfHbSgLGsuUW+pk1HiCKECgYEAxcXyor8Flys1Gd/
IOGJPdsOitnlvecQgTzjKks+Hqfferxketdvd0mG7Hiimmz75QN + 8D6yHR/
rl4rlKERTGMqm/5K6C + HQ5qUOHmneyWefRV + gKu1Zt1YcLSSY0D-
pbn2LUqW6YHueBjJLPkBM7IyZGNtcn9niQPjda8MvcP32UCgYEAyGK-
bNrdP4U8RIJlz6vbyo4F0viQh1ydNY6PgX/038y19dey + mPk8MQh3nZFW-
vN0rpsSgcOqjSj/1avXETmlGNMhFM2IfR5jnGW0oQMD8nRXfe0qheB2sEeV
xIQIIThP2WAxD0elKff0iq4yJlC5Y0utpzlC5Xq8Rq8RcA4xn0
CgYEAiFggHzyr4PyjnhPx1b5I5CqZOU1cocipMHW + ahnyg-
CXm+jXKKzviPzCrLG5/9ZUjhyr3XqLlnKUG6RguTLpSrUjDhyccGacevWdVzBLq/
PpJlI5QT7iU/dkc2bAhwVEdwxOagRZkSyu7jekKsJnSaMwUsxfu5aAcrP82Pbh/09
UCgYEAAtyAGILb2uBIWx10jVUYFktK/19F4o3ur3 + nsk7hQHMaD86uv0MvByZY-
0LY2Aq2y50We + PgCGulljay2jWgaILmuj69L5TP6coa0AqbSLwum3ock/9yDu1qJU
6e60D + Y0JC + qwaM65TeVgAey3v/Q9t9TNWeKGaxkDPsV29ITCjECgYA0cNjdb/
ifHRL0QMMy3oJJjn3HAfDwbpO1UN0CQ2SoVfob1Cy7byq2NTnf-
PjHjheeVmLW6e3zMXHfezAJ42y3SNLHH5vVJkauecorZZMnVC8iVla8v0D/
Yvti8bkiqt4YcQGSWpTE8TrdjfdR6gNOgrvVjrVHWvD4R78ftZS7O + 5A=
=7fEnw52DyQMSF4U35duRjfs/g3HsNGDyhLlxE2oeBRDGrTKWdgDVR-
5ghes4xf63jkhueijvzdfhuucTG8jtjdihdbnhnxndflklddVhjgFfzLiwIVv0Jac/lhJ + r/
OVbn3PyOeXacBBo1vuZGKpoTrrI465//ZZAWAk5Uukb9h9CzHCiSQofbx68qXMK/
bXuiWFFGRWSdOSN53eX3j/gm8 + wvWRwYBnahIhgoLIQd8mVwzSoimg-
4sQnAenep7y6a + 0znATQNU1boANn2vDyUHtKLlBLibLBI9fHAycWg3 + nKQ
AUBTFsxvPSBulAFalfHbSgLGsuUW+pk1HiCKECgYEAxcXyor8Flys1Gd/
IOGJPdsOitnlvecQgTzjKks+Hqfferxketdvd0mG7Hiimmz75QN + 8D6yHR/
rl4rlKERTGMqm/5K6C + HQ5qUOHmneyWefRV + gKu1Zt1YcLSSY0Dpbn-
2LUqW6YHueBjJLPk7IyZGNtcn9niQPjda8MvcP32UCgYEAyGKbNrdP4U8RIJlz-
6vbyo4F0vih1.

ydNY6PgX/038y19dey + mk8MQh3nZFWvN0r-
psSgcOqjSj/1avXETmlGDZiAy5w7cvghhRTdxujNh-
j2gbdrbcsxgnhhsvdDVGgtsrWAxD0elKff0iq4yJlC5Y0utpzlC5Xq8Kvdyij8dpsufjk3-
etundfDGTyhu4HYVsuiv7MRC4JTNEVthr6JFB8xnfmsHJRMCF7fklnuhivduhsuih-
HBGdbvhdn4hjbh9hjbhBVH3jvbkj4shrh/rTNKJnbuir4bhjsbvBGHH9uhuiHgfsvd/
gubSJ5ohybvj8afhvuIBJKugibv.

-----END RSA PRIVATE KEY-----.

-----BEGIN PUBLIC KEY-----.

MIIBjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAMs7TPybmK
XuzbEGcfQsBuHu2SigegjXbzlrS94ktXNevH40cpjEGEfUYxX3qoUwJXhTSN
b9TnoTRNdL6cgwhdByly07dEM7 + sfK1Jw/lvLjsZQmYuoIWfJJAmNey55rD/
oqkFV6wnpG5O97JJEHjCEDppqbcUoqmbPBBAUsP5yZcvAhKJorhicPa-
jBnN8ZOoYm6pv/1KmVBtNxY/edSKQFUsekbbMvjgkpWcqaBbGsR62N-
WPERK58jURerPYI39u + 97yGEEu3Wm2zOXjAqmTX2 + 6Jb1cXC7lMzdZ/UOQRz
9Fw + BdHCIEJRMUktjdQD4BNq5kub4tTAcqU2h6AyUQIDAQAB.

-----END PUBLIC KEY-----.

The modified hash algorithm of 512-bit message digest is proposed which operates on eight 64-bit words. Each block is considered as sixteen 64-bit words, eighty 64-bit words are produced.

The initial input value to SHA-512 is hexadecimal and is given below.

7D03A66713842D93 1F83D9ABFB41BD6.
3C6EF372FE94F82B A54FF53A5F1D36F1.
5BE0CD19137E2179B05688C2B3E6C1F.
2BF549C5158E2A72510E527FADE682D1.

Each of the eight words in a block becomes the hash which is shifted to the position of the next word in the block. The first word in the block is being replaced by the modified eighth word in the block. The constant words of length 80 used in SHA-512, obtained from the fraction of cube roots of the first eighty primes, which are:

766A0ABB3C77B2A8A831C66D2DB43210240CA1CC77AC9C6E2748774CD
F8E5D353380D139D95B3DF4CC5D4BECB3E42B6923F82A4AF194F9B
CA273ECEEA26619391C0CB3C5C95A63243185BE4EE4B28C550C7DC3
D5FFB4E2983E5152EE66DFAB72BE5D74F27B896F80DEB1FE3B1696B19BDC06A7
25C71235 C19BF174CF692694C 92722C851482353B6EFBE4786384.

F25E3AB5C0FBCFEC4D3B2 A0FC19DC68B8CD5B00327C898FB213FEAD.
A7DD6CDE0EB165CB0A9DCBD41FB D876F988DA83115312835B01457.
06FBE2DE92C6F592B0275BF597FC7BEEF0EE47137449123EF65CD2J
L7C6E00BF33DA88FC2D5A79147930AA72559F111F1B605D019142929670A0E6E7
03GU281C2C92E47EDAEE62E1B21385C26C92619 A4C116B8D 2D0C8
4D2C6DFC5AC42AE50A73548BAF63DE428A2F98D728AE22E49
B69C19EF14AD227B70A8546D22FFCN6KVC24B8B70D0F89791A4506CEB
DE82BDE9C76C51A30654BE308CC702081A6439EC4A7484AA6EA6
E483 5FCB6FAB3AD6FAECA2BFE8A14CF1036 CF40E35855771202E9B5DBA581.

89DBBC3956C25BF348B538F57D4F7FEE6ED178 4B0BCB5E19B48AD807.
AA98A30302426C44198C4A4758174C9EBE0A15C9BEBEC90BEFFFA23631
E2597F299CFC657E2AC67178F2E372532B106AA07032BBD1B884C87814.

A1F0AB72C28DB77F523047D841B710B35131C471B78A5636F43172F604
2CAAB7B40C72493D7AB1C5ED5DA6D81181E376C085141AB5D186B8C.

721C0C207 6D192E819D6EF521806F067AA72176FBA0A637DC5A2C898.
A6113F9804BEF90DAE A81A664BBC423001682E6FF3D6B2B8A3BEF9A3.
F7B2C67915431D67C49C100D4C5B9CCA4F7763E373Y4N06CA6351E003.
826F748F82EE5DEFB2FC4ED8AA4AE3418ACB D69906245565A910.

The above hash value changes when the input value applied to the modified hash algorithm is changed. It ensures data integrity during transmission over wireless networks. The combination of modified asymmetric and hash algorithms ensures secure monitoring of plant information and protects the sensitive process data from unauthorized access. It also ensures smooth functioning of plant equipments which deals with data monitoring and control applications. Asymmetric algorithm is complex and it achieves higher level of security than the symmetric algorithm. Hash function provides protection of password and ensures data integrity. It is necessary to propose the security algorithm that ensures end-to-end secure plant operations, low latency and high speed.

8. Conclusion

This proposed work is the implementation of modified asymmetric and hash algorithms using embedded system with process monitoring through internet. The temperature and gas process data is read through the sensor and encrypted using the embedded system. The strength of the proposed modified asymmetric encryption is it generates large key size of 4096-bit and the 512-bit message digest to ensure confidentiality and integrity. This proposed modified asymmetric algorithm


provides authentication and modified hash algorithm provides data integrity as well as Internet Protocol (IP) security. This encrypted data is transmitted across the internet. The cipher text is received through the internet by providing the correct IP address. The decryption algorithm is executed at the embedded system to obtain the plain text. The original process data is monitored through the SCADA master terminal. This proposed work achieves data integrity as well as data confidentiality. It offers low latency and achieves higher efficiency of more than 95 percent in securing the sensitive plant information. It allows secure monitoring of plant information through the SCADA system. This proposed work can be applicable for securing sensitive process information in any industrial applications. It provides the cost-effective solutions in protecting the expensive industrial devices from unauthorized attacks and ensures workers safety.

Author details

J.S. Prasath
KCG College of Technology, Chennai, India

*Address all correspondence to: jsprasath@gmail.com; prasath.ei@kcgcollege.com

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] J. S. Prasath, U. Ramachandraiah, G. Muthukumar, "Modified Hardware Security Algorithms for Process Industries Using Internet of Things," Taylor and Francis Journal of Applied Security Research, Article in Press, pp. 1-14, 2020.
- [2] Amiruddin, Anak Agung Putri Ratna, Riri Fitri Sari, "New Key Generation and Encryption Algorithms for Privacy Preservation in Mobile Ad Hoc Networks," International Journal of Communication Networks and Information Security, Vol. 9, No. 3, pp. 376-385, 2017.
- [3] Vikas K.Soman, Natarajan V, "An Enhanced Hybrid Data Security Algorithm for Cloud," IEEE International Conference on Networks and Advances in Computational Technologies, Trivandrum, India, pp. 416-419, 2017.
- [4] Adviti Chauhan, Jyoti Gupta, "A Novel Technique of Cloud Security Based on Hybrid Encryption by Blowfish and MD5," IEEE International Conference on Signal Processing, Computing and Control, Solan, India, pp. 349-355, 2017.
- [5] M. Harini, K. PushpaGowri, C. Pavithra, M. Pradhiba Selvarani, "A Novel Security Mechanism Using Hybrid Cryptography Algorithms," IEEE International Conference on Electrical, Instrumentation and Communication Engineering, Karur, India, pp. 1-4, 2017.
- [6] Anushka Gaur, Anurag Jain, "Analyzing Storage and Time Delay by Hybrid Blowfish-MD5 Technique," IEEE International Conference on Energy, Communication, Data Analytics and Soft Computing, Chennai, India, pp. 2985-2990, 2017.
- [7] G. Prabu Kanna ; V. Vasudevan, "Enhancing The Security Of User Data Using The Keyword Encryption And Hybrid Cryptographic Algorithm In Cloud," IEEE International Conference on Electrical, Electronics, and Optimization Techniques, Chennai, India, pp. 3688-3693, 2016.
- [8] J.S.Prasath, U.Ramachandraiah, S.Prabhuraj, G. Muthukumar, "Internet of Things based Hybrid Cryptography for Process Data Security." Journal of Mathematical and Computational Science, Vol. 10, No. 6, pp. 2208-2232, 2020.
- [9] Abdul Mohsen Almalawi, Adil Fahad, ZahirTari, Abdullah Alamri, Rayed AlGhamdi, Albert Y. Zomaya, "An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems," IEEE Transactions on Information Forensics and Security, Vol. 11, pp. 893-906, 2016.
- [10] Yichi Zhang, Lingfeng Wang, Yingmeng Xiang, Chee-Wooi Ten, "Power System Reliability Evaluation with SCADA Cyber Security Considerations," IEEE Transactions on Smart Grid, Vol. 6, pp. 1707-1721, 2015.
- [11] Riccardo Muradore, DavideQuaglia, "Energy-Efficient Intrusion Detection and Mitigation for Networked Control Systems Security," IEEE Transactions on Industrial Informatics, Vol. 11, pp. 830-840, 2015.
- [12] Wei Jiang, Yue Ma, Nan Sang, Ziguozhong, "Dynamic Security management for real-time embedded applications in Industrial Networks," Elsevier Journal of Computers and Electrical Engineering, Vol. 41, pp. 86-101, 2015.
- [13] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, H. F. Wang, "Multi-attribute SCADA-Specific Intrusion Detection System for Power

Networks,” *IEEE Transactions on Power Delivery*, Vol. 29, pp. 1092-1102, 2014.

[14] AbdalhosseinRezai, ParvizKeshavarzi, Zahra Moravej, “Secure SCADA communication by using a Modified Key Management scheme,” *Elsevier Journal of ISA Transactions*, Vol. 52, pp. 517-524, 2013.

[15] Wei Jiang, ZhenlinGuo, Yue Ma, Nan Sang, “Measurement-based research on Cryptographic algorithms for Embedded Real-time Systems,” *Elsevier Journal of Systems Architecture*, Vol. 59, pp. 1394-1404, 2013.

[16] Manuel Cheminod, Luca Durante, Adriano Valenzano, “Review of Security Issues in Industrial Networks,” *IEEE Transactions on Industrial Informatics*, Vol. 9, pp. 277-293.

[17] Igor NaiFovino, AlessioColetta, Andrea Carcano, Marcelo Masera. 2012. Critical State-BasedFiltering System for Securing SCADA Network Protocols. *IEEE Transactions on Industrial Electronics*, Vol. 59, No.10, 3943-3950.

[18] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. NaiFovino, and A. Trombetta, “A Multi-Dimensional Critical State Analysis for Detecting Intrusions in SCADA Systems,” *IEEE Transactions on Industrial Informatics*, Vol. 7, pp. 179-186, 2011.

[19] D.J. Kang, J.J. Lee, B.H. Kim, D. Hur, “Proposal strategies of Key management for Data encryption in SCADA network of Electric Power Systems,” *Elsevier Journal of Electrical Power and Energy Systems*, Vol. 33, pp. 1521-1526, 2011.

[20] Gregory M. Coates, Kenneth M. Hopkinson, Scott R. Graham, Stuart H. Kurkowski, “A Trust System Architecture for SCADA Network Security,” *IEEE Transactions on Power Delivery*, Vol. 25, pp. 158-169, 2010.

[21] Donghyun Choi, Sungjin Lee, Dongho Won, Seungjoo Kim, “Efficient Secure Group Communications for SCADA,” *IEEE Transactions on Power Delivery*, Vol. 25, pp. 714-722, 2010.

[22] Igor NaiFovino, Andrea Carcano, Marcelo Masera, Alberto Trombetta, “An experimental investigation of malware attacks on SCADA systems,” *International Journal of Critical Infrastructure Protection*, Vol. 2, pp. 139-145, 2009.

[23] C. Ten, C. Liu, G. Manimaran, “Vulnerability Assessment of Cyber Security for SCADA systems,” *IEEE Transactions on Power Systems*, Vol. 23, pp. 1836-1846,2008.

[24] J.S. Prasath, S. Jayakumar, K. Karthikeyan, “Real-Time Implementation for Secure monitoring of Wastewater Treatment Plants using Internet of Things,” *International Journal of Innovative Technology and Exploring Engineering.*, Vol. 9, No. 1, 2997-3002, 2019.

[25] Esmael V. Maliberan, “Modified SHA1: A Hashing Solution to Secure Web Applications through Login Authentication,” *International Journal of Communication Networks and Information Security*, Vol. 11, No. 1, pp. 36-41, 2019.

[26] J.S.Prasath, U.Ramachandraiah, “Modified Asymmetric and Hash Algorithms for Internet Enabled Industrial Automation,” *Test Engineering and Management Journal*, Vol. 83, pp. 7431-7444, 2020.

[27] Amjad Y. Hendi, Majed O. Dwairi, Ziad A. Al-Qadi, Mohamed S. Soliman, “A Novel Simple and Highly Secure Method for Data Encryption-Decryption,” *International Journal of Communication Networks and Information Security*, Vol. 11, No. 1, pp. 232-238, 2019.

Smart Home Monitoring System Using ESP32 Microcontrollers

Marek Babiuch and Jiri Postulka

Abstract

This chapter deals with the implementation of our own monitoring system with home security. The system is designed using IoT modules and uses ESP32 microcontrollers. The chapter describes the design of the system, its hardware components, software implementation, security solutions, communication, the collecting and monitoring of processed data, as well as the quantification of costs for the production and deployment of this system. The proposed system secures a house by detecting an intruder in the building, triggering an alarm and capturing it all with camera images, and then sending data to the owner's smart mobile phone. The secondary task of the system is to collect data from sensors for monitoring the temperature of an object and presenting it via a web server.

Keywords: ESP32, microcontroller, MQTT, IoT, monitoring systems

1. Introduction

The main idea was to use the ESP32 chip, which we have installed with a camera module to monitor a house, as well as to monitor the temperature of individual rooms such as the corridor and boiler room. The proposed system contains PIR sensors and temperature sensors in the monitored rooms and connected camera modules to the ESP32 microcontroller. The data collected from the sensors is sent wirelessly to the control unit, which is the Raspberry PI Zero. The other created HW modules are an input panel with a touch screen and input panel with an ESP32 Wroom board and a membrane keyboard. It is used to unlock and lock the house to activate the motion sensor and cameras. The control unit is extended using the GSM module IoT-GA5-B. It ensures the sending of messages to the mobile phone of the homeowner, which inform him/her about the security status of the house. Other HW modules are touch screens showing the current status and temperature in the rooms and control LEDs, recording the status of the entire system. The main aspects of the whole system are the following:

- Motion detection using PIR sensors
- The capturing of a camera image
- The monitoring of the physical quantities of the household such as temperature, humidity, the possibility of extension to other monitored physical quantities

- The storage and monitoring of measured data
- Access to data via a web server
- Responsive applications for mobile devices
- Wireless communication, MQTT communication, System security

GSM communication with the system (at the request of the owner)

In the individual chapters of this article, we will gradually describe the selection of hardware components, the software implementation of the entire system, the design of system security, the installation and testing of the monitoring system and the financial aspects of implementation with the possibility of expansion.

2. Background and related works

Shortly after its introduction, the ESP32 microcontroller became fully integrated into industrial automation, mainly into the deployment of embedded systems and various IoT tasks. Its great advantage is undoubtedly its price, circuit structure, the possibility of connecting peripherals and IoT modules and other sensors, as well as having excellent support for creating applications. The ESP32 chip is well implemented as a web server, using wireless Wi-Fi communication, Bluetooth and mostly the MQTT communication standard at the level of exchanging messages with the surroundings. It often works with another suitable microcomputer such as the Raspberry Pi. In [1] we have described in detail the creation of an architecture for an embedded system based on the ESP32 chip and the processing of a monitoring task using Dual-core for its operation, while one core takes care of obtaining and processing data from the sensors, the other core solves communication issues with the surrounding devices. The suitability of using multiple cores is also discussed [2] in the field of deployment of ESP32 in the field of machine learning and neural networks. Currently, ESP32 is implemented in a wide range of IoT industries and applications in the areas are listed in this chapter with a specific example of use.

The ESP32 chip is suitable for implementation in applications of various monitoring and security tasks. The common element of the implemented applications is its high-quality, low-cost solution in multiple areas, such as the Solar Water Pumping System in agriculture [3] or various types of monitoring systems in farming [4, 5]. Other suitable deployments can also be found in the field of monitoring air quality systems [6], monitoring LPG leakage [7] or waste management systems [8]. Article [9] describes the use of ESP32 as a web server for a real-time photovoltaic monitoring system. With the help of other suitable peripherals such as a camera, linear actuators and drivers, we can use ESP32 to implement a position control system [10] or a Smart surveillance system [11]. The ESP32 microcontroller is also used to perform Device-Free Passive (DfP) tasks such as detection, localization and the tracking of human entities [12], or utilized using GPS as an indoor positioning system [13] or security system [14]. SCADA systems have also begun to focus on IoT in recent years [15]. SCADA architectures have evolved over the years from monolithic (stand-alone) through distributed and networked architectures to the latest Internet of Things (IoT) architecture. Article [16] presents the design and implementation of the Open Source SCADA system by using a local server IoT platform as Master Terminal Units for handling data processing and human-machine interactions and an ESP32 micro-controller as the Remote Terminal Unit for receiving, processing and sending the remote data from the field instrumentation devices.

Article [17] describes the design of a Smart home IoT system and notes the trade-off between scalability, security and data collection efficiency for the Internet-of-Things sensor networks. The article [18] discusses security risks in detail and states that IoT systems carry risks in terms of security and privacy. Without eliminating these risks, the IoT requirements are not adequately fulfilled. The article describes the vulnerability of the IoT system at individual layers of the network system and proposes a solution in the form of a security model. Article [19] presents an environment monitoring system based on ESP32 together with a wireless sensor network. The result is not only its low cost but also its low power system, which is today a sought-after benefit when deploying a monitoring system. An integral part of IoT today is the area of iHealth and health monitoring using smart embedded systems. This area is evidenced by the use of ESP32 for monitoring heart rate [20], an image recognition system for blind people in article [21], or a smart saline monitoring system in intravenous therapy [22]. Article [23] describes the possibilities of implementing a control system with the help of human gestures on wearable devices. ESP32 chips are implemented not only in iHealth monitoring systems, but now smart systems are also being developed for the comfort and convenience of mental health and well-being. The article [24] documents a case study of the creation of an intelligent lighting system according to the detection of human emotions. It uses, among other things, necessary hardware components, the predecessor of the ESP32 chip, namely ESP8266, and the already mentioned MQTT communication standard, which is the most common communication standard. It is possible to use the encrypted MQTTs protocol with SSL/TLS certificates to secure MQTT communication, as described [25]. Other security features of communication such as the Algorithm of Advanced Encryption Standard (AES), implemented on ESP32 with a LoRa module to secure wireless communication are described in [26]. Lora communication techniques and the used Zigbee communication standard in the field of home automation are dealt with in [27, 28]. IoT architecture, in general, is described with its requirements and paradigms in [29] and the integration of IoT and cloud computing in terms of the configuration of embedded systems is described in a research article [30]. In the field of embedded systems and monitoring, ESP32 development variants with touch screens can be used, either integrated into the ESP32 Wrover board or connected externally [31]. A comparative analysis of ESP32 and other modules with the ESP32 recommendation for the IoT area is discussed [32]. We agree with his conclusions, as the authors express the idea that the microcontroller operating system FreeRTOS is open source software providing great support for real-time applications. Thus, it is expected that ESP32 will play a major role in the design of future IoT systems and embedded projects. The performance evaluation mentioned in the article [33] - Enabling ESP32-based IoT Applications in Building Automation Systems confirms the suitability of using ESP32 in low-cost applications with an industrial-grade performance. The results of this article are essential for developing cheap but nevertheless reliable industrial solutions based on SoC.

3. Hardware modules

The system will therefore include the Raspberry Pi as a control unit. The set will also include three camera modules equipped with a PIR and a temperature sensor. The ESP32 equipped with a display and a membrane keypad for entering the boiler room and a touch screen also controlled by the ESP32 at the main entrance will be used to lock the house. The set also includes one ESP32 microprocessor equipped with two temperature sensors, the same as the camera module, so we will not describe this device in the next chapter. Some basic components are shown in **Figure 1**.



Figure 1.
Some hardware components.

3.1 Input panel with membrane keyboard

The system contains the following elements:

- ESP-WROOM-32

ESP-WROOM-32 belongs among Microcontroller units (MCU), which are fundamentally computer board platforms with CPU, memory, busses and built-in peripherals indispensable to read connected sensors or drive actuators [34].

- Monochrome character 16x2 (16 characters per line and 2 lines)

The LCD display is connected via a I2C bus. It is a serial bus that divides the connected device into a master or slave category. One wire is used to transmit the clock signal (SCL - synchronous clock) and is the data channel (SDA - synchronous data).

- Matrix membrane 4x4 keyboard for single-board computers

The keyboard contains characters (1 ÷ 9, A ÷ D and special characters # and *), and these are connected to 8 pins. The buttons in the individual rows and columns are always connected to a common wire. After pressing the button, the row and the column are always connected at a given point. The manufacturer guarantees a service life of 100 million presses.

- Alarm

The last component in this set is an alarm whose operating voltage is in the range of 3-24 V and intensity 95 dB. Its function is to trigger a loud tone in the event of an intruder entering the house, which should alert you that there is an intrusion in your home.

3.2 Input panel with touch screen

The main input will be the same ESP32 model as the previous set. As for the display unit, a 2.8" USART touch screen from NEXTION will be used. The most significant advantage of these displays is the graphic editor, in which you can quickly and without problems define the graphical and touch environment and perform simulations. This editor saves a lot of time when developing applications. The resolution of

this display is 320x240 together with 65 thousand colors, and adjustable brightness will make the perfect choice for all kinds of applications. The biggest advantage is the communication via USART, through which the created graphical environment can be uploaded directly to the display, in which a 4 MB flash memory is integrated. The second variant is using a micro SD card, for which this display has a slot. The USART interface is also used to communicate with the microprocessor, through which it receives variables and sends a click response to the element. Any other display can be used to implement the system. Our goal was not to reduce the price of components to the lowest level but to develop a monitoring system in a comfortable, user-friendly environment, so we chose this display [35]. Custom-made input panels and boards with the help of a 3D printer are shown in **Figure 2**.

3.3 Camera modules

The composition contains the following elements:

- Ai-Thinker ESP32-CAM. Again, this is a Wi-Fi + BT MCU module, expanded by 520 KB SRAM, including external 4MPSRAM. It also supports OV2640 and OV7670 cameras and SD cards. ESP32-CAM can be widely used in various IoT applications. It is suitable for home intelligent devices, industrial wireless control, wireless monitoring and other IoT applications.
- RGB LED module. It is a type of RGB LED with a common anode, i.e. the individual ledges are switched by connecting to the ground. Its function will be to give information about the current status of the camera module (whether it is connected to a Wi-Fi network, or whether there was an error loading the camera or other error messages) indications in different colors.
- Temperature sensor DALLAS DS18B20, This sensor allows you to measure the temperature in a range of -55 to $+125$ degrees Celsius, while in the range of -10 to $+85$ degrees Celsius it has a guaranteed accuracy of $\pm 0.5^\circ$ C. It is available in a TO-92 package, which is similar in size to ordinary transistors, it is also available in a waterproof variant, where the sensor is sealed in a stainless steel stick. OneWire bus is used for communication, which uses only one communication pin. This sensor also supports the so-called parasitic mode, where only two wires are needed to connect the sensor to the microcontroller.
- PIR module: There are many variants of motion sensors designed for applications similar to this one. Again, we chose from two variants. The first of them



Figure 2.
Input board in boiler room.



Figure 3.
ESP32 camera modules in the corridors and living room.

is called HC - SR501, whose supply voltage is in the range of 4.5 to 20 V and the output logic is 0 / 3.3 V. The size of the sensor is 32x24mm, so it is a larger sensor but allows you to set the sensitivity of the sensor and timing using two potentiometers. Also, its detection distance is more than sufficient; the manufacturer states a distance of up to 7 meters with a sensing angle of 120°. The second variant is a smaller module called AM312. The operating voltage is in the range of 2.7 to 12 V, while the output logic is the same as for the previous module. The dimensions of the plate are only 10x8mm. However, the detection distance is also smaller, approximately in the range of 3–5 m at a scanning angle of 110°. **Figure 3** shows the mounted camera modules in the individual rooms.

3.4 Control unit

Even though the Raspberry Pi is already running with the fourth generation of the development board, the economical Pi Zero variant is sufficient for our system. Also, Zero W has built-in Wi-Fi and Bluetooth. Zero is built with a single-core ARMv6 processor clocked at 1 GHz. It also has 512 MB of RAM, audiovisual output via Mini-HDMI, a classic micro-USB 2.0 connector and power supply via micro-USB. It also has a 40-pin GPIO interface. The control unit is then further expanded by a GSM module called IOT-GA6-B. It is an ideal solution for IoT devices that communicate via a serial line at the TTL level, and it supports voice calls, SMS, GPRS data transfer and standard AT commands. Of course, it would be possible to connect this module to the ESP32 microprocessor. Since it would be necessary to forward all messages from the control unit to the input panel and then to the terminal, this implementation would still be too complicated. For this reason, we decided to omit one intermediary and connect the device directly to the control unit. The architecture of our IoT system is therefore centralized, for our smart system, the central point is the control unit, which is also an MQTT broker that communicates with ESP32 modules and input panels. The conditions under which it would be appropriate to decentralize the IoT system describes [36], among other things, a detailed comparison of the decentralized IoT architecture approach with different approaches.

4. Application software

This chapter describes the individual services that have been used in this system. There are also descriptions of source code fragments that are uploaded to individual

devices. Thanks to this outline, it is possible to understand better the way the whole assembly works. **Figure 4** shows the resulting software application, which is described in this subchapter.

The operating system is the latest version of Raspbian called Buster and runs on Debian Linux version 4.16. It is an operating system designed directly by the manufacturer of this board, thanks to which the reliability and stability of this system should be guaranteed.

The Raspberry Pi runs:

- MQTT Server - Service for communication with microcontrollers
- Apache 2 - Web server
- MariaDB - SQL database for measured values.
- Python Scripts - Covers all logic in communication with microcontrollers, as well as storing the data in a database.

4.1 Communication

All communication takes place wirelessly via Wi-Fi. The protocol by which individual devices communicate is MQTT. MQTT (formerly: Message Queuing Telemetry Transport, today MQ Telemetry Transport) is a simple and undemanding protocol for transmitting messages between clients via a central point - a broker. Thanks to this simplicity, it is easy to implement it even in embedded devices and it is spread relatively quickly. For the MQTT protocol, the transmission is performed using TCP and uses the publisher-subscriber design pattern. So there is one central point (the MQTT broker) that takes care of exchanging messages. In our case, the broker will be Raspberry Pi. **Figure 5** shows a block diagram of MQTT Communication. Messages are sorted into so-called topics, and the device either publishes in the given topic (publish), i.e. it sends data to a broker, which stores and distributes them to other devices, or is subscribed to a topic or more topics (subscribe). The broker then sends all messages with the given topic to the device. Of course, one device can be a publisher in some topics and at the same time a subscriber in others.

4.2 Input panel

Its purpose will be to lock a house by pressing a key, as well as to unlock it by entering a 4-digit numeric code. At the same time, it serves as an element to alert the intruder via an alarm. In the source code, we have programmed macros that allow you to define how many times it is possible to enter a pin when the house is locked, the time delay of attempts, the default display of data on the screen, reading keystrokes and communicating using the MQTT protocol. Furthermore, in the source code, which we will not mention here, we perform Wi-Fi configurations and check whether a connection to Wi-Fi has taken place, followed by repeated attempts, LED signaling and the reconfiguration of the device. Another functionality programmed in the input panel is the function for setting the MQTT client, which defines the address of the broker, the communication port and instructions with the settings for receiving messages from the broker. The main loop of the program first calls a function that verifies if the ESP is still connected to the Wi-Fi network, and if not, it tries to connect it several times. If this still does not work, the device will restart. Subsequently, a function is called that verifies the connection

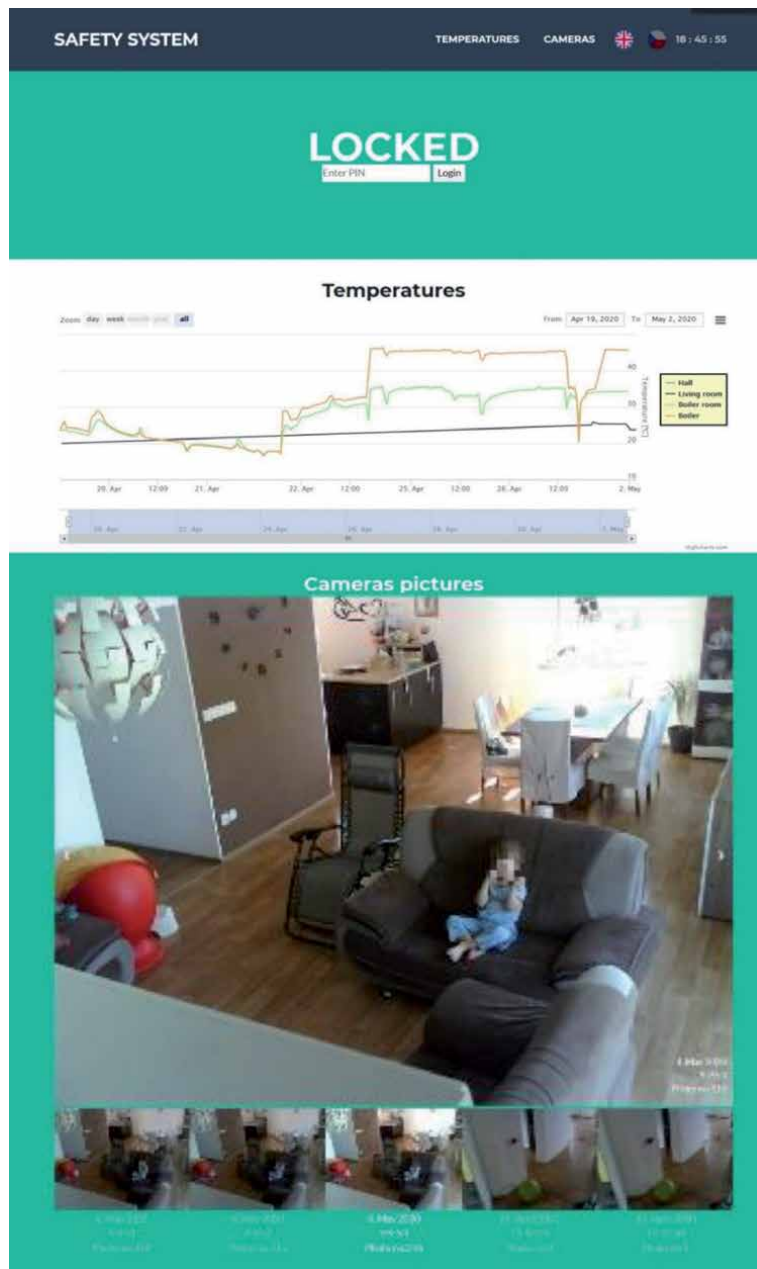


Figure 4.
Measuring and monitoring application.



Figure 5.
MQTT communication.


```

void loop()
{
  WifiCheck();

  mqttRun();

  temp_current_millis = millis();
  if (temp_current_millis - temp_last_capture_millis > temp_interval) // If the time
  // for the next temperature measurement has elapsed
  {
    temp_last_capture_millis = millis();
    Temperature();
  }

  if (locked) // If house is locked
  {
    bool PIRSENSOR = digitalRead(PIR);
    if (PIRSENSOR) // If movement is captured
    {
      if (!last_PIR)
      {
        Serial.println("Public Intruder");
        MQTTclient.publish(PublishIntruder, MQTTid, MQTTpubQos); // Publish message
      }
      current_millis = millis();
      if (current_millis - last_capture_millis > capture_interval) // 1 sec elapsed
      {
        last_capture_millis = millis();
        take_send_photo(); // Také a photo and send snapshot
      }
    }
    else
    {
      last_PIR = PIRSENSOR;
    }
  }
}
}

```

Figure 7.
Fragment of the source code of the ESP32 camera module.

interval, processing the measured values and sending it to the broker. Another check finds out if the broker received the message about a locked house. If this happens, then the movement in the area is checked by reading the value from the PIR sensor. If the movement has been detected, a house intrusion report is sent to the broker. At the same time, it is checked whether the time required to take another screenshot has elapsed since the last loop. If so, a function is called to take a picture using the connected camera. This image is then sent and then saved in the directory.

4.4 Control unit

The purposes of this unit are as follows: To mediate communication using the MQTT protocol, to store incoming data in a database and to enable access to them, and also to provide an HMI interface. We use the MariaDB database to store data, which is a relational database that is community developed by the successor branch of MySQL.4 users with different rights to use the database. This is mainly to ensure that we secure access to information as much as possible. The primary user is the Administrator with rights related to the *esp_db* database, in which all tables are stored, telephone numbers, measured values, captured photographs and the individual states of closing and opening the house. Users are also WRITE, who can write into the individual TABLES and users READ who can only read data.

The telephone numbers and names of the users to whom these telephone numbers belong are stored in the DB Phones table. The other two columns of this table indicate the rights of individual telephone numbers, one column indicates the right to unlock or lock the house via SMS, while the other column indicates the manipulation of the gateway. The last is LOGIN, which stores a 4-digit PIN to unlock the house.

As already mentioned, Raspberry runs a web server, which is mediated by the Apache 2 service. This service runs several PHP scripts supplemented by an HTML structure. JavaScript then takes care of the dynamics of the pages together with a jQuery. Everything is unified in design and formatted using a CSS style.

The application screen on the webserver is divided into three parts. The first is the option to log in by entering the PIN code. After a successful login, the house can be unlocked or locked remotely. We see this directly in the application on the mobile device in **Figure 8**.

In the next section, there is a *HighCharts* graph, which clearly shows the course of temperatures over time measured on individual devices. In the chart, it is possible to select a specific device from which we want to see the course of temperatures. It is also possible to choose which time range we want to display the temperature. There are also additional options in which graphs can be exported either to an image or, for example, to an Excel worksheet or similar files. In the last section called the Cameras picture, we will find all captured photos from all connected camera devices that are connected to the set. These images are sorted according to the date that is displayed for each photo and are assigned a serial number for greater clarity. The site can be switched according to the language preference to the local environment, if necessary. This site has also been customized for mobile devices such as smartphones. We assume that these sites will mainly use these devices.

4.5 Python scripts

These scripts mediate communication for the MQTT protocol, as well as for storing data in a database and communication with the GSM module. There are three python files on Raspberry, one of which acts as an MQTT client. It listens to everything that is sent to the broker and performs a specific action based on the incoming message. The second python script contains functions that allow access

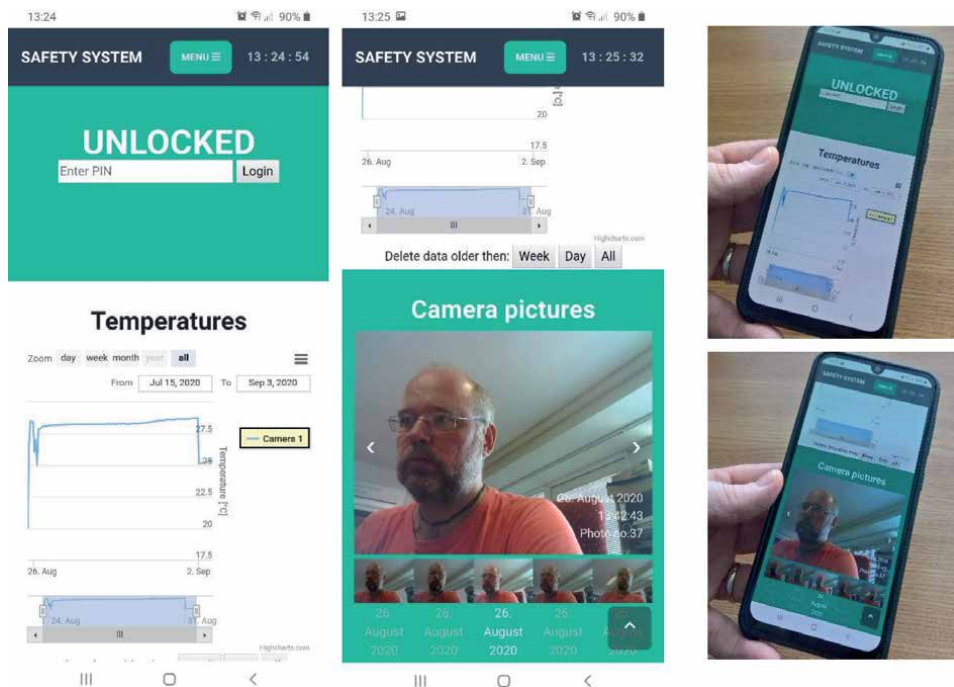


Figure 8.
Web server – Application on a mobile device.

to the database. The third communicates via a serial line with the GSM module and mediates communication with the end-user.

The Python script contains functions that are used to retrieve values from the database, such as the current state of the house (locked/unlocked). Another function is used to write values into tables with temperatures that come as a message from the camera modules via the MQTT protocol.

The first one, *on_message*, is called when the MQTT broker sends a message. An example of this code is shown in **Figure 9**. In this function, the message is then filtered using a topic, where we are interested in what the name of the message is. In the case of a topic called Refresh, we know that a new device has just joined the communication and is requesting an update of the data. Therefore, a message is sent to this command, which calls a function that reads the current pin code and house status from the database. It sends these messages after an encrypted communication.

The second function then takes care of changing the state of the house (unlocking/locking). If it occurs, the message will be retransmitted after 5 seconds to make it 100% certain that all connected devices will receive the message. At the same frequency, messages are also sent about the state of the intruder (i.e. if one of the camera devices detected movement), if the house is in a locked state.

The script that mediates communication with the GSM module first sets up this module. It then goes into an endless loop in which it checks to see if an intruder has entered the house. If so, then an SMS message is sent to all telephone numbers that are registered in the database. If a telephone number is dialed, the number is first verified whether it is written in the database and has the appropriate rights to open the gateway. Then there is contact on the relay module, which is connected to the Raspberry Pi, it closes and thus the gate opens. We added this function at the request of the owner of the house on which the system is installed. By sending an SMS “lock” or “unlock” it is possible to manipulate the security system of the house. An example of this code is shown in **Figure 10**. Of course, when sending an SMS, we first check whether the number from which the SMS was sent is registered in the database together with the appropriate permissions. Subsequently, an SMS message is sent to this telephone number. This message contains the current time, the status of the house (unlocked/locked), the last measured temperatures from all sensors and, last but not least, the remaining credit on the SIM card (so that the user knows

```
def on_message(client, userdata, message):
    global last_LockState, last_IntruderState
    if ("Refresh" in message.topic):
        print("Requirement for Refresh")
        client.publish('Read/ESP/Lock', payload=last_LockState, qos=1, retain=False)
        time.sleep(1)
        client.publish('Read/ESP/Pin', payload=ReadFromDB("Read/ESP/Pin"), qos=1, retain=False)
        time.sleep(1)
        client.publish('Read/ESP/Intruder', payload=last_IntruderState, qos=1, retain=False)
    elif ("Write" in message.topic):
        print("Topic: " + message.topic + " Data: " + message.payload.decode('utf-8'))
        SaveToDB(message.topic, message.payload.decode('utf-8'))

def publish_state():
    global last_LockState, last_IntruderState
    while client.connected_flag:
        time.sleep(2)
        if ReadFromDB("Read/ESP/Lock") is not last_LockState:
            last_LockState = ReadFromDB("Read/ESP/Lock")
            client.publish('Read/ESP/Lock', payload=last_LockState, qos=1, retain=False)
        if last_LockState and ReadFromDB("Read/ESP/Intruder") is not last_IntruderState:
            last_IntruderState = ReadFromDB("Read/ESP/Intruder")
            client.publish('Read/ESP/Intruder', payload=ReadFromDB("Read/ESP/Intruder"), qos=1, retain=False)
```

Figure 9.
Fragment of python script for MQTT communication.


```
while True:
    reply = bytes.decode(ser.read(ser.inWaiting()))
    if reply != "":
        print (reply)
        if "+CLIP:" in reply: # if calling
            phoneNumber = reply[reply.index("+CLIP: ") + 8:reply.index("+CLIP: ") + 20]
            for x in range(len(PhoneTable)):
                if PhoneTable[x][0] in phoneNumber and 1 is PhoneTable[x][2]:
                    print("i am opening the gate")
                    GPIO.output(17, GPIO.LOW)
                    time.sleep(2)
                    GPIO.output(17, GPIO.HIGH)
                    # SendSMS(phoneNumber)
            ser.write(str.encode('ATH\r')) # stop call

        if '+CMT: ' in reply: # if SMS
            phoneNumber = reply[reply.index("+CMT: ") + 8:reply.index("+CMT: ") + 20]
            print (phoneNumber)
            for x in range(len(PhoneTable)):
                if PhoneTable[x][0] in phoneNumber and 1 is PhoneTable[x][1]:
                    if "lock" in reply.lower():
                        print("i am locking the house")
                        ChangeState("true")
                        time.sleep(1)
                        SendSMS(phoneNumber)
                    elif "unlock" in reply.lower() or "unlocking" in reply.lower():
                        print("i am unlocking the house")
                        ChangeState("false")
                        time.sleep(1)
                        SendSMS(phoneNumber)
                    else:
                        SendSMS(phoneNumber)
            ser.flushInput() # Clear buf
            ser.flushOutput() # Clear buf
            time.sleep(1)
```

Figure 10.
Fragment of python script with GSM module communication.

when it is necessary to recharge the mobile credit). At the request of the owner, a function has been added that controls the boiler temperatures, which are sent from one of the modules. The owner is informed by an SMS message when a limit of 80° C is exceeded.

5. System security measures

The proposed system must, of course, comply with safety principles. As these are a number of interconnected technologies, there are several recommendations for security measures, from completely common to complex solutions. The system uses a number of hardware modules, communication elements, a web server, a database, configurable access to the control unit, an input pin to unlock the system and all this must be secured. The basic safety recommendation is to separate the wireless network for the IoT modules of our system from the wireless network of the house in which we use PCs, laptops and other common devices. This can be done either by the total separation and operation of two separate networks or by different variants of VLAN and micro segmentation of the network, depending on the available hardware of the specific solution. In our case, it is a completely separate network.

Another important element is the principle of preventive protection of the Raspberry Pi control unit, securing SSH remote access, compliance with security principles and best practices for MariaDB database such as Avoiding running *mysqld*

as root, Limit *ssh* access, Limit *sudo* access on the MariaDB and the use of security plugins [37]. The limitation of the number of attempts to enter the input Pin on the instrument panel is solved by the possibility of configuring the selected time delay after the unsuccessful entry of 3 consecutive attempts, including sending a notification to a mobile phone to pre-selected phone numbers.

Last but not least, securing the communication of the MQTT protocol is also important, as this protocol was not originally designed for security, but for its availability and undemanding implementation. We can apply the security of this communication in implementations of later Mosquitto brokers. This is done using authentication and authorization mechanisms that allow you to add plugins. This is done using authentication and authorization mechanisms that allow you to add plugins. This is a client authentication using a client IDs, Access Control List or x509 Clients certificates. To protect the contents of your MQTT messages you can use TLS or SSL Security and Payload Encryption.

6. Installation and testing of security system

To be able to implement the device into operation and place it in the house, these devices must have mechanical protection and at the same time, allow the most convenient possible installation. For this reason, we decided to produce cases for camera modules and input panels with a keyboard and touch screen on a 3D printer.

It is necessary to insert ESP32, an LCD, a keyboard, a speaker and DC-DC voltage converter in the input panel, which will reduce the distribution voltage of 24 V DC to the required 5 V DC. The housing is designed for wall mounting around the front door, with a height suitable for the user. The camera module contains the following elements: ESP32-CAM, temperature sensor, a PIR sensor, a RGB LED, IR LED, a NPN transistor, a constant current source, a DC-DC converter and an antenna for Wi-Fi. We designed the housing so that the camera can be mounted in the corner of the room so that it blends in with the surrounding walls as much as possible and does not disturb the integrity of the room in any way. The camera is placed in a rotating sphere to be set in which area of the room we want to record. After the implementation of hardware modules, the 3D printing of assembly boxes and the mounting and software implementation of the entire system, the testing and installation phase took place, first in the laboratory and after verification of the functionality of all modules assembly into a residential house. **Figure 11** shows

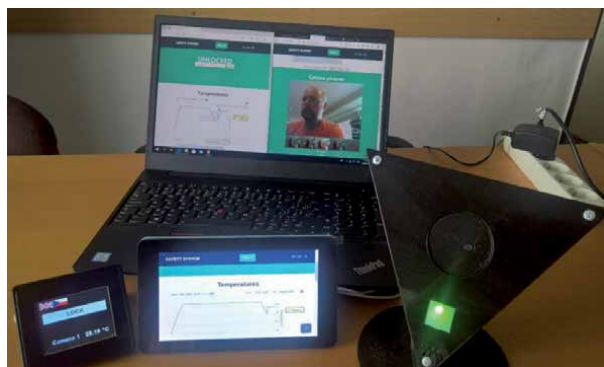


Figure 11.
Testing of hardware modules and software functionality.



Figure 12.
Installation of the ESP32 cam module in the living room.

the testing of the input panel, the touch screen, one camera module and the display of monitoring temperatures and captured photos on the webserver in individual modes simulating a closed and unlocked house. This test also related to testing the alarm and sending messages by the GSM module to pre-selected telephone numbers.

We installed three camera modules and two input panels in the house, one with a touch screen and the other with a membrane keyboard for more comfortable handling in a dusty room, with one module for collecting temperatures in the boiler room and directly from the boiler and one Raspberry Pi Zero W control unit.

All components of the device are powered from a 12 V source with direct voltage so that even on long routes, the voltage of individual devices is at least 5 V. **Figure 12** shows one of the three camera modules that has just been activated and signals its initialization with the color green. You can see the input panels in figures in the Hardware Modules chapter. On the left, we can see a panel with a membrane keyboard, which is located in the boiler room, and the other is located at the main entrance. Raspberry Pi Zero W is located in the distribution board, together with a GSM modem and a relay module. The individual devices were tested of course, after connecting all the modules, some parts of the code were fine-tuned to ensure the greatest possible comfort of use and reliability.

7. Price of the created monitoring system

The price of the created monitoring system is around 250 euros, when, of course, we add up only all the purchased hardware modules. The price of the 3D printing of the designed boxes is negligible. We did not include the price of our own development work and software implementations in the price of the system. The advantage of the whole system is that when it is extended by other measuring IoT modules in the current configuration, the final price increases only by a few Euros, when increasing the number of existing measuring units, the final price increases in the price of a particular module, whether it is another ESP32 module equipped with a camera and sensors or input or display units, the price of which is given in the table. The price of the resulting device could be reduced, but our goal was not to choose the cheapest components. In the production of more pieces of the monitoring system and with the quantity discount of selected components, we would get even lower prices. Prices in the **Table 1** are given according to the European local market, it is possible to get lower prices elsewhere.

Device	Item	Price in Euro
Control unit	Rpi Zero	12
	SD card	10
	GSM Modem	8
	Relay module	2
	DC – DC converter	4
	RTC real time module	2
1 piece	Total	38
Camera module	ESP32 – CAM	14
	Temperature sensor	1,9
	PIR sensor	2,3
	DC – DC converter	1,5
	RGB LED	0,6
	Resistances 1k Ω , 4k7 Ω	0,4
3 pieces	Total (rounded)	62
Input panel with Touch display	ESP32	8
	Touch display	25
	DC – DC converter	2
1 piece	Total	35
Input panel	ESP32	8
	Keyboard	1,5
	LCD Char Display	4,5
	Alarm	3
	DC – DC converter	2
1 piece	Total	19
Measurement device	ESP32	8
	Temperature sensor	2
	Outdoor temperature sensor	4
	DC – DC converter	2
1 piece	Total	16
Power supply	Power supply	30
Electrical wiring	House wiring, cables, LED indicators	30
Total price		230 Euro

Table 1.
Price of components.

8. Conclusion

We designed a camera system with an ESP32 chip, which was installed on a house; the system is connected to its Wi-Fi network. The set was extended using input panels at the main entrance and the entrance to the boiler room. It was necessary to create a parent device that would serve as a data store and control for

the entire report. This was done using a Raspberry Pi, whose function is to pass communication to all other elements in the assembly. At the same time, it serves as a repository of photographs captured from camera modules and a web server is running on it, which allows access to data on this unit.

In an idle state, the camera modules are inactive (they do not take camera pictures). However, if the house is locked, which can be done by using two input panels or using a web server or by sending an SMS command, then after a certain time delay, the camera modules will be activated after leaving the house. These check whether motion has been detected on the PIR sensors. If this situation occurs, then this information is sent to the control unit and sent to all camera modules. Immediately afterwards, images are taken from all camera modules with a second interval. An alarm located in one input panel is also activated, and an SMS message is sent from the control unit via a GSM modem to predefined telephone numbers stating that an intrusion into the house has occurred. Images taken from the camera modules are sent via HTTP post to the control unit, where they are stored.

ESP32 modules are also equipped with temperature sensors for monitoring the temperature in the rooms. These temperatures are processed and sent to the control unit, where they are stored in the database. The whole set is further expanded by one device that reads the temperatures in the boiler room and the temperature directly from the boiler. This makes it possible to monitor the dependence between the temperatures of the boiler and in the individual rooms. Thanks to this, a house could be heated more efficiently.

The web server, which also runs on this unit, and ensures easy and clear access to all data that is continuously stored here. The current status of the house and the possibility of locking with it after entering the PIN code is displayed on the webserver. The application displays a graph that clearly plots the temperatures measured on individual devices as a function of time. There is the possibility of filtering a certain device from which the data were measured, as well as the possibility of displaying a certain time interval. The last section of the website is a gallery of photos that were taken on camera modules and are sorted from the most recent, with each photo showing the exact time when it was captured.

A GSM modem is connected to the control unit, which enables the remote manipulation of the house via an SMS message, but only for selected telephone numbers. By sending an “unlock” or “lock” command, this action is performed. Back then, the user is informed about the execution via an SMS message, which contains the current state of the house (unlocked/locked). The last measured temperatures from all connected sensors and the credit balance on the SIM card are also sent. If any other SMS message is sent, again only from selected numbers, the user is sent an SMS with the same content as when sending the order. When the telephone number rings, the relay is connected, which is connected to the gate for entering the property of the house. At present, we are also dealing with taking camera pictures in poor visibility conditions or complete darkness. This problem could be solved by removing the IR filter located on the camera lens and illuminating the room by adding, for example, IR LEDs, the light of which would not attract attention because it is not visible to the human eye.

Acknowledgements

This work was supported by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems project,

CZ.02.1.01/0.0/0.0/16_019 /0000867 within the Operational Programme Research, Development and Education and the project SP2020/571 Research and Development of Advanced Methods in the Area of Machines and Process Control by the Ministry of Education, Youth and Sports.

Author details

Marek Babiuch* and Jiri Postulka
Department of Control System and Instrumentation, Technical University of
Ostrava, Ostrava, Czech Republic

*Address all correspondence to: marek.babiuch@vsb.cz

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Foltýnek P, Babiuch M and Šuránek P. Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. *Meas. Control*. 2019; 7-8. DOI: 10.1177/0020294019857748.
- [2] Dokic K, Martinovic M and Radisic B. Neural Networks with ESP32 - Are Two Heads Faster than One? *Conference on Data Science and Machine Learning Applications, CDMA 2020*. DOI: 10.1109/CDMA47397.2020.00030.
- [3] Biswas SB. and Tariq Iqbal M. Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller. *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*. DOI: 10.1109/CCECE.2018.8447749.
- [4] Kodali RK and Valdas A. MQTT Based Monitoring System for Urban Farmers Using ESP32 and Raspberry Pi. *International Conference on Green Computing and Internet of Things, ICGCIoT 2018*. DOI: 10.1109/ICGCIoT.2018.8752995.
- [5] Ram CRS, Ravimaran S, Krishnan RS, Ismail, M, et al. Internet of Green Things with autonomous wireless wheel robots against green houses and farms. *Int. J. Distrib. Sens. Netw.* 2020; 6. DOI: 10.1177/1550147720923477.
- [6] Sarjerao BS and Prakasarao A. A Low Cost Smart Pollution Measurement System Using REST API and ESP32. *International Conference for Convergence in Technology, I2CT 2018*. DOI: 10.1109/I2CT.2018.8529500.
- [7] Abdullah AH, Sudin S, Ajit MIM, et al. Development of ESP32-based Wi-Fi Electronic Nose System for Monitoring LPG Leakage at Gas Cylinder Refurbish Plant. *International Conference on Computational Approach in Smart Systems Design and Applications, ICASSDA 2018*. DOI: 10.1109/ICASSDA.2018.8477594.
- [8] Atmajaya D, Kurniati N., Astuti W, et al. Digital Scales System on Non-Organic Waste Types Based on Load Cell and ESP32. *East Indonesia Conference on Computer and Information Technology: Internet of Things for Industry, EIConCIT 2018*. DOI: 10.1109/EIConCIT.2018.8878667.
- [9] Allafi I and Iqbal T. Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring. *2017 IEEE Electrical Power and Energy Conference, EPEC 2017*. DOI: 10.1109/EPEC.2017.8286184.
- [10] Kljakić S, Rajs V, Bodić M and Cvetković N. Position Regulation System with Camera and Microcontroller ESP32. *EUROCON 2019 - 18th International Conference on Smart Technologies*. DOI: 10.1109/EUROCON.2019.8861899.
- [11] Rai P and Rehman M. ESP32 Based Smart Surveillance System. *International Conference on Computing, Mathematics and Engineering Technologies, iCoMET 2019*. DOI: 10.1109/ICOMET.2019.8673463.
- [12] Gunasagaran R, Kamarudin LM and Zakaria A. Embedded Device Free Passive (EDfP) System: Sensitivity of ESP32. *IEEE Student Conference on Research and Development (SCORED)*. DOI: 10.1109/SCORED.2018.8710808.
- [13] Misal SR, Prajwal SR, Niveditha HM, et al. Indoor Positioning System (IPS) Using ESP32, MQTT and Bluetooth. *Fourth International Conference on Computing Methodologies and Communication 2020*, DOI: 10.1109/ICCMC48092.2020.ICCMC-00015.
- [14] Murmu PP, Paul H, Roopa JJ and Timothy AJ. A Novel modernistic

techniques in women security system using ESP32 and Arduino Uno. *International Conference on Signal Processing and Communication, ICSPC 2019*. DOI: 10.1109/ICSPC46172.2019.8976745.

[15] Lojka T, Miškuf M and Zolotová I. Industrial IoT Gateway with Machine Learning for Smart Manufacturing. *IFIP Advances in Information and Communication Technology*, Volume 488, 2016, DOI: 10.1007/978-3-319-51133-7_89.

[16] Aghenta LO and Iqbal MT. Low-Cost, Open Source IoT-Based SCADA System Design Using Thingier.IO and ESP32 Thing. *Electronics 2019*;8. DOI: 10.3390/electronics8080822.

[17] Mishra A, Reichherzer T, Kalaimannan E, et al. Trade-offs involved in the choice of cloud service configurations when building secure, scalable, and efficient Internet-of-Things networks. *Int. J. Distrib. Sens. Netw.* 2020; 2. DOI: 10.1177/1550147720908199.

[18] Aydos M, Vural Y and Tekerek A. Assessing risks and threats with layered approach to Internet of Things security. *Meas. Control.* 2019; 5-6. DOI: 10.1177/0020294019837991.

[19] Catelani M, Ciani L, Bartolini, A, et al. Characterization of a low-cost and low-power environmental monitoring system. *International Instrumentation and Measurement Technology Conference 2020*. DOI: 10.1109/I2MTC43012.2020.9129274.

[20] Škraba A, Kolozvari A, Kofjac D., et al. Prototype of Group Heart Rate Monitoring with ESP32. *8th Mediterranean Conference on Embedded Computing, MECO 2019*. DOI: 10.1109/MECO.2019.8760150.

[21] Kushnir V, Koman B and Yuzevych V. IoT Image Recognition

System Implementation for Blind Peoples Using esp32, Mobile Phone and Convolutional Neural Network. *International Scientific and Practical Conference on Electronics and Information Technologies, ELIT 2019*. DOI: 10.1109/ELIT.2019.8892289.

[22] Ghosh D, Agrawal A, Prakash N and Goyal P. Smart Saline Level Monitoring System Using ESP32 And MQTT-S. *IEEE 20th International Conference on e-Health Networking, Applications and Services, Healthcom 2018*. DOI: 10.1109/HealthCom.2018.8531172.

[23] Nemeč D, Janota A, Gregor M, et al. Control of the mobile robot by hand movement measured by inertial sensors. *J. Electr. Eng.* 2017; 99,4. DOI: 10.1007/s00202-017-0614-3

[24] Cupkova D, Kajati E, Mocnej J, et al. I. Intelligent human-centric lighting for mental wellbeing improvement. *Int. J. Distrib. Sens. Netw.* 2019; 9. DOI: 10.1177/1550147719875878.

[25] Nikolov N and Nakov O. Research of Secure Communication of Esp32 IoT Embedded System to.NET Core Cloud Structure using MQTTS SSL/TLS. *International Scientific Conference Electronics, ET 2019*. DOI: 10.1109/ET.2019.8878636.

[26] Iqbal A and Iqbal T. Low-cost and Secure Communication System for Remote Micro-grids using AES Cryptography on ESP32 with LoRa Module. *IEEE Electrical Power and Energy Conference, EPEC 2018*. DOI: 10.1109/EPEC.2018.8598380.

[27] Fan C and Ding Q. A novel wireless visual sensor network protocol based on LoRa modulation. *Int. J. Distrib. Sens. Netw.* 2018; 14, Issue 3. DOI: 10.1177/1550147718765980.

[28] Yao F, Yang SH and Xia B. A Zigbee based home automation: System design and implementation. *Meas. Control.*

2008; 41, Issue 10, pp. 310-314. DOI: 10.1177/002029400804101003.

[29] Asensio Á, Marco Á, Blasco R, et al. Protocol and architecture to bring things into internet of things *Int. J. Distrib. Sens. Netw.* 2014. DOI: 10.1155/2014/158252.

[30] Puliafito A, Celesti A, Villari M, et al. Towards the integration between IoT and cloud computing: An approach for the secure self-configuration of embedded devices. *Int. J. Distrib. Sens. Netw.* 2015. DOI: 10.1155/2015/286860.

[31] Babiuch M, Folytynek P and Smutny P. Using the ESP32 Microcontroller for Data Processing. *International Carpathian Control Conference, ICC 2019*. DOI: 10.1109/CarpathianCC.2019.8765944.

[32] Maier A, Sharp A and Vagapov Y. Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. *Internet Technologies and Applications, ITA 2017*. DOI: 10.1109/ITECHA.2017.8101926.

[33] Carducci CGC, Monti A, Schraven MH, et al. Enabling ESP32-based IoT Applications in Building Automation Systems. *IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019*. DOI: 10.1109/METROI4.2019.8792852.

[34] Takacs G, Vachálek J and Rohal'-Ilkiv B. Online Structural Health Monitoring and Parameter Estimation for Vibrating Active Cantilever Beams Using Low-Priced Microcontrollers. *Shock Vib.* 2015. DOI: 10.1155/2015/506430.

[35] *Nextion: HMI displays* [online]. 2020 [cit. 2020-08-24]. Available from: <https://nextion.tech/basic-series-introduction/>

[36] Mocnej J, Pekar A, Seah WKG, et al. Quality-enabled decentralized IoT

architecture with efficient resources utilization. *Rob. Comput. Integr. Manuf.*, Volume 67, 2020. DOI: 10.1016/j.rcim.2020.102001.

[37] Percona Live. *MariaDB Security Features and Best Practices* [online]. May 2019 [cit. 2020-09-04]. <https://www.percona.com/live/19/sessions/mariadb-security-features-and-best-practices>



Edited by Fausto Pedro García Márquez

The Internet of Things (IoT) is a closed-loop system in which a set of sensors is connected to servers via a network. The data from sensors are stored in a database and then analysed by IoT analytics. The results are usually employed by either humans, machines, or software to make decisions about the operation of the system. This book provides an interface between the main disciplines of engineering/technology and the organizational, administrative, and planning capabilities of managing the IoT.

Published in London, UK

© 2021 IntechOpen
© ipopba / iStock

IntechOpen

