EAI/Springer Innovations in Communication and Computing

Song Guo
Deze Zeng *Editors*

# Cyber-Physical Systems: Architecture, Security and Application

EAI
RESEARCH MEETS INNOVATION

Springer

# EAI/Springer Innovations in Communication and Computing

**Series editor**
Imrich Chlamtac, CreateNet, Trento, Italy

## Editor's Note

The impact of information technologies is creating a new world yet not fully understood. The extent and speed of economic, life style and social changes already perceived in everyday life is hard to estimate without understanding the technological driving forces behind it. This series presents contributed volumes featuring the latest research and development in the various information engineering technologies that play a key role in this process.

The range of topics, focusing primarily on communications and computing engineering include, but hardly limited to, wireless networks; mobile communication; design and learning; gaming; interaction; e-health and pervasive healthcare; energy management; smart grids; internet of things; cognitive radio networks; computation; cloud computing; ubiquitous connectivity, and in mode general smart living, smart cities, Internet of Things and more. The series publishes a combination of expanded papers selected from hosted and sponsored European Alliance for Innovation (EAI) conferences that present cutting edge, global research as well as provide new perspectives on traditional related engineering fields. This content, complemented with open calls for contribution of book titles and individual chapters, together maintain Springer's and EAI's high standards of academic excellence. The audience for the books consists of researchers, industry professionals, advanced level students as well as practitioners in related fields of activity include information and communication specialists, security experts, economists, urban planners, doctors, and in general representatives in all those walks of life affected ad contributing to the information revolution.

## About EAI

EAI is a grassroots member organization initiated through cooperation between businesses, public, private and government organizations to address the global challenges of Europe's future competitiveness and link the European Research community with its counterparts around the globe. EAI tens of thousands of members on all continents together with its institutional members base consisting of some of the largest companies in the world, government organizations, educational institutions, strive to provide a research and innovation platform which recognizes excellence and links top ideas with markets through its innovation programs.

Throughs its open free membership model EAI promotes a new research and innovation culture based on collaboration, connectivity and excellent recognition by community.

More information about this series at http://www.springer.com/series/15427

Song Guo • Deze Zeng
Editors

# Cyber-Physical Systems: Architecture, Security and Application

Springer

EAI
RESEARCH MEETS INNOVATION

*Editors*
Song Guo
Department of Computing
Hong Kong Polytechnic University
Hong Kong, China

Deze Zeng
School of Computer Science
China University of Geosciences
Wuhan, Hubei, China

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-319-92563-9 ISBN 978-3-319-92564-6 (eBook)
https://doi.org/10.1007/978-3-319-92564-6

Library of Congress Control Number: 2018948682



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

# Preface

Over the last two decades, much effort has been devoted to promoting the integration and interaction between the cyber and physical parts of our world. This motivates the concept of Cyber-Physical System (CPS), which has already attracted attention from the government, academia and industry. Regarding the interaction of the cyber world with the physical world, a hot topic emerging is the Internet-of-Things (IoT), which describes the vision that everything is interconnected. CPS is similar to IoT but presents a higher combination and coordination between the physical and informational world. CPS has been applied in a variety of domains such as: industry, agriculture, transportation, and electricity system (e.g., smart grid), imposing huge potential in promoting the life quality of human beings.

CPS is a cross-discipline topic covering a wide range of fields from hardware to software, like integrated circuit, embedded system, control, computation, communication, system integration, and so on. The development on each field shall advance the development of CPS. It is significant to investigate how to advance the development of CPS, especially with joint consideration of other newly emerging computing and communication technology or concept like cloud computing, big data, fog computing, crowdsourcing, and so on, from various aspects such as architecture, application, security, and privacy.

To this end, we invited pioneering researchers and engineers to discuss some key issues from these aspects and to present some insightful opinions, concepts, innovations and achievements in CPS. Various new CPS technologies from diverse aspects enable a higher level of innovation towards intelligent life are discussed in this book. It provides deep insight to the future integration, coordination and interaction between the physical world, the information world and our human beings. These works shall inspire more future studies to further advance the development and exploitation of CPS.

Last but not least, we would like to thank all the chapter authors for their invaluable sharing and contributions. We would like to thank the kind support from the EAI/Springer Innovations in Communications and Computing book series editor Dr. Imrich Chlamtac. We also appreciate the hardworking of all the people who work together to push forward the publication of the book.

Hong Kong, China — Song Guo
Wuhan, China — Deze Zeng

# Contents

# Part I
# Architecture and Applications

# Chapter 1
# Envisioned Network Architectures for IoT Applications


**P. Sarwesh, N. Shekar V. Shet, and K. Chandrasekaran**

**Abstract** Internet of Things is the auspicious technology that connects different internet enabled devices in single network architecture. IoT contributes effective service in various applications such as industrial automation, health care sectors, and home automation. Availability of low cost devices makes IoT as innovative paradigm in large-scale wireless network research. Challenges in IoT applications vary from each other. For example, in smart grid applications QoS is more important, whereas for land slide monitoring applications, energy efficiency and reliability are the major requirements. Thus, in this chapter, we come up with various network architectures that are suitable for IoT applications. The network architectures are designed by combining different optimization techniques into single network design, to satisfy specific network requirements. This chapter elaborates the major issues that affect the network performance and suitable solutions for those issues by means of efficient network architectures.

## 1.1 Introduction

Internet of Things (IoT) is the ever-growing network of smart devices that promotes global information sharing. In the phrase Internet of Things, the word "things" can include from small watch to big vehicle [1]. It creates smart environment in every fields such as smart city, smart health, smart grid, smart market, smart agriculture, and smart home. MEMS technology is the major reason for IoT development, since availability of low power and low-cost devices is achieved by MEMS technology. Internet enabled devices work autonomously with its features such as sensing, communicating, and computing. IoT network is the combination of

P. Sarwesh (✉)
Madanapalle Institute of Technology and Science, Madanapalle, AP, India

N. S. V. Shet · K. Chandrasekaran
National Institute of Technology Karnataka, Mangaluru, Karnataka, India



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_1

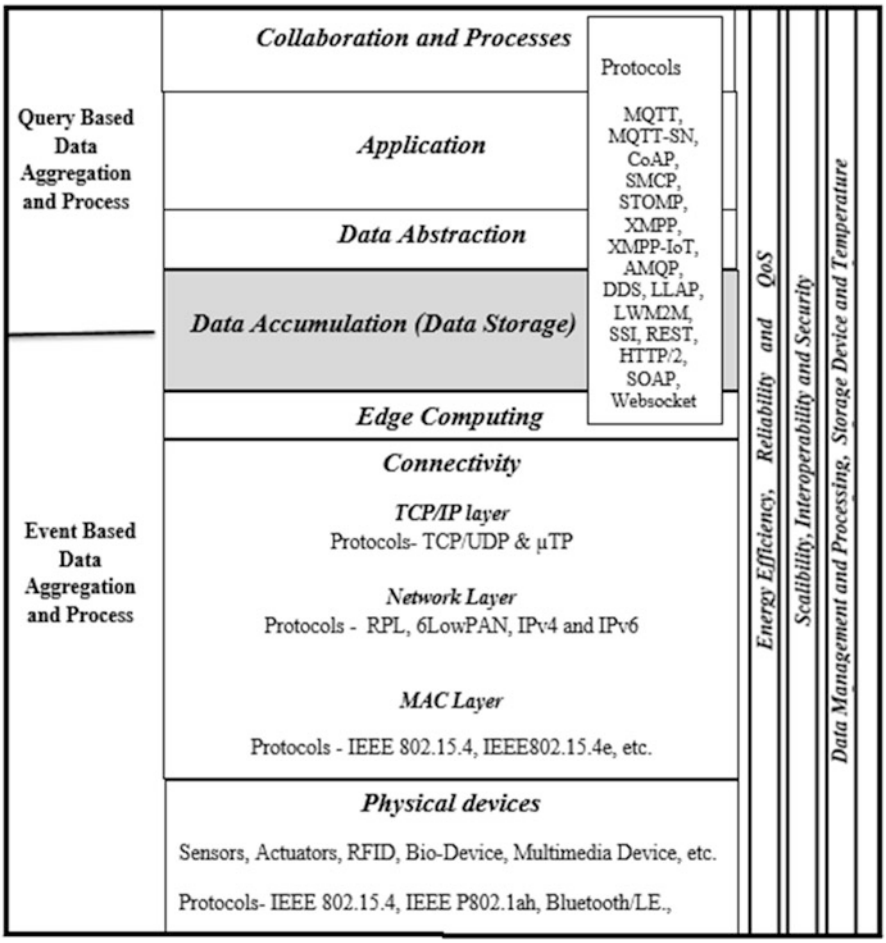



**Fig. 1.1** IoT World forum reference model and its challenges

higher end devices (servers) and lower end devices (sensors). Therefore, connecting this interoperable network environment with efficient network architecture is the major challenge in IoT networks [2]. IoT is the new idea, but the technologies that required for built IoT are well matured, therefore IoT technology is conceivable in different fields. Existence of remote and wired advances with productive processing and correspondence structure encourages IoT to give powerful correspondence between machine-to-machine, machine-to-individual, individual to-machine, and so forth [3]. Internet of Things can be described as smart device network that collects the environment information with the help of IoT devices, for every applications network devices and network requirements vary, thus based on the network requirements and network challenges we designed an effective network architecture in this chapter.

In Fig. 1.1, lower layers manage event related data (occasion related data), whereas higher layers are responsible for question related data (query based data). The middle layer is used to store and recoup the information made by higher layers and lower layers. The major goals in lower layers are energy efficiency, Quality of Service (QoS) and reliability, since they handle constrained devices and low power radio connections [4]. In higher layers, data organization and data get ready are considered as genuine troubles, since they handle a gigantic piece of data. Arranging powerful and compact storage devices and controlling the temperature dispersed from higher end devices (servers) are the noteworthy difficulties in the center layer (storage layer). IoT devices in low power frameworks (IoT, WSN) routinely continue running by limited battery control, obliged preparing speed and irrelevant memory [3, 5]. In Internet, devices are connected to the electricity grid with stable power supply. However, IoT devices are usually battery sourced devices, which are unstable in nature. Web uses stable associations, for instance, Ethernet and SONET/SDH joins, whereas IoT holds low power radio connections (IEEE 802.15.4) that are lossy and precarious in nature, which says components of IoT contrast from consistent Internet.

## 1.2 Network Level Challenges in IoT

In this chapter, we concentrate on challenges of lower layers (Network Level Challenges). This section describes the importance of energy efficiency, reliability and QoS for low power IoT networks.

### *1.2.1 Energy Efficiency*

Efficient energy utilization is one of the primary challenges in wireless smart devices networks (WSN, IoT, etc). Smart devices used in environmental monitoring applications as well as in commercial application are operated by battery power. In forest fire monitoring application (environmental monitoring), if the sensor battery drains out its power, it is very difficult to replace the battery. Similarly, in industrial boiler monitoring (critical monitoring), if the smart device drains out its power, it severely affects the service. Unbalanced energy utilization disturbs the network lifetime, cutback in lifetime of networks severely affects customer as well as service provider. Thus, energy is considered as one of the valuable resources in low power wireless network applications [6]. To utilize energy in efficient way factors that affect energy efficiency need to analyze and to be prevented. Effective physical layer techniques, MAC based power control techniques and network layer optimization techniques can be the effective solution to optimize the energy efficiency in networks.

### *1.2.2 Reliability and QoS*

Low power IoT network utilizes low power radio protocols (e.g. IEEE 802.15.4) to interconnect switches, routers and other IoT devices, which are battery sourced devices. Providing higher duty cycle protocols in resource constrained network environment is impossible, similarly providing main power lines to every IoT device is impossible. Therefore, it is understood that providing effective communication between IoT devices through unstable network links is the major consideration in IoT network design. Due to unstable network links (low power radio links) chance of packet loss is more and dynamic link variations severely affect the data transmission. Sensor devices in remote territory (e.g. landslide monitoring) might be scattered unplanned and they are little in size and constrained in battery. In such network environment, providing reliable data transfer with low duty cycle protocols is the challenging task in wireless network research. Routing process consumes huge amount of energy, for reliable data transfer routing is the key element, since efficient routing design is the major requirement in low power IoT networks [6]. Data transmission through unstable links leads to packet loss, which severely affects the network reliability and QoS. Thus, an effective network design is needed to achieve better reliability and QoS.

## 1.3 Factors that Affect the Network Performance

In low power wireless network, various factors affect the network performance; in this chapter, some of the major factors that affect the energy efficiency, reliability and Quality of Service (QoS) are described. Figure 1.2 describes the major factors that affect the network performance.

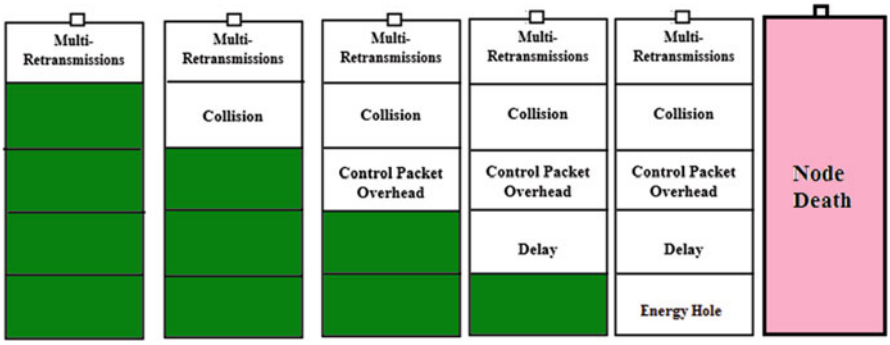



**Fig. 1.2** Factors that affect the network performance

### 1.3.1 Energy Hole (Node Overload)

The smart devices which are close to the base station convey enormous measure of information activity. Since smart devices (sensor devices) which are close to the base station forward the sensed data of its own as well as the forwarded data of other sensors, and as a result of huge data load, nodes drain out its power in short span of time [6]. When every node near to base station coverage area drains out its battery source, communication to the base station will be blocked, which leads to network re-initialization, this problem is referred to as energy hole issue. Energy hole problem severely affects the network performance, cost and time (re-installation time). Thus, balancing energy consumption and avoiding node overload can be the active solution to avoid energy hole issue.

### 1.3.2 Multi-Retransmissions

In low power wireless networks, lossy and unstable links are used to connect the low power smart devices (battery sourced device). And quality of radio links that are used in low power network environment will vary frequently. Data transmission in unstable links leads to packet loss, which are the prime reason for data re-transmission [6]. Data re-transmission affects the network reliability as well as energy efficiency. Thus, avoiding re-transmission decidedly improves the network performance. Data-re-transmissions can be avoided by transmitting data in stable links (reliable links).

### 1.3.3 Collision

Collision is a noteworthy issue in remote system; it influences the execution and lifetime of the wireless network. Amid the information transmission from transmitter node and receiver node in specific channel, new entry of transmission signal from another node in same channel prompts collision. Collision gives rise to increase in packet re-transmission, this prompts energy wastage and huge network congestion. Increase in collision increases the network latency and degrades the network reliability and energy efficiency. In wireless network, TDMA is the suitable technique to avoid collision, but TDMA increases control overhead as well as energy utilization. Thus, an effective MAC based optimization technique is required to avoid the network collision [7].

### 1.3.4 Control Packet Overhead

Extra data (control packets) required for specific protocol to establish association and correspondence is referred to as packet overhead. These control packets are specified as CTS, RTS, RREQ, RREP, etc. based on protocols. These packets are the prime reason for establishing the communication, and with the aid of this information network connectivity is maintained. Therefore, control packets are fundamental need for establishing connection, but it should not exceed its limit, which severely affects the energy efficiency of the network. Avoiding excess control packet usage (huge control overhead) can improve the energy efficiency with better network connectivity [8].

### 1.3.5 Delay

Delay is one of the major reasons that affects the QoS of the network. Data transfer in unstable links, packet loss, re-transmissions, unbalanced buffer usage are the major reasons for increase in end-to-end delay. Thus, preventing above factors will highly prevent end-to-end delay [6].

### 1.3.6 Motivation

From the above discussions, it is seen that energy efficiency, reliability and QoS are the major requirements in low power IoT networks. In low power wireless network, huge amount of energy is consumed by communication unit. Therefore, optimizing communication unit highly improves the network performance. There are various techniques such as node placement technique, MAC based power control technique, MAC based scheduling technique, routing technique, TCP/IP based optimization techniques etc. to optimize the communication unit of the network. All these techniques satisfy specific network requirements, when compared to utilizing the features of single optimization technique. Integrating the features of various optimization techniques in single network architecture highly improves the network performance. In wireless network research, cross layer protocols such as EQSR Ben-Othman and Yahya [9], HAN Larzon et al. [10], XPL Vuran and Akyildiz [11], and Breath Park et al. [12] are developed by combining the features of various optimization techniques to obtain better network performance. From this observation, we designed various network architectures that integrate the features of various optimization techniques to achieve application specific network requirements.

## 1.4 Envisioned Network Architecture for Low Power IoT Networks

This section describes the various envisioned network architectures that are proposed for IoT applications, proposed network architectures are designed by integrating various optimization techniques in single network architecture. In this chapter, network architectures are developed for some of the popular IoT applications.

### *1.4.1 E-Health*

Smart health applications have gained lot of attention in many health care sectors. Availability of intelligent biomedical devices, effective middle ware services and low power communication technologies (low bandwidth radio protocols) made the possibility of accessing medical information of patient, even they are in remote area. Now-a-days because of mobile telemedicine system, patients continue their regular activities [13]. IoT technology supports various e-health applications, such as wearable device and its data accessing (automatic diagnosis system), remote monitoring of elderly patients' health condition, and immediate responding in emergency situations. Smart health systems in hospitals collect the information from ECG measuring devices, EEG measuring devices, X-ray devices and store it in their database, this health information of patients can be accessed by health experts and based on the health history from smart health systems medical experts proceed the treatment [13]. Because of smart health systems, accurate health information of patients is obtained and effective health care service is provided to them.

E-health application is one of the critical monitoring applications. Constant monitoring of patients with chronic conditions and reporting to medical officer is the major task that is practiced in e-health applications. The prime components required for health care system are communication enabled devices, data acquisition boards, wireless routers and efficient server that maintain the health-related data. All these components will be connected by low power radio links and most of the wireless devices (wireless routers and data acquisition boards are operated by battery powered devices. In such network scenario, reliability is the major requirement, if data loss occurs it severely affects the e-health service. The layers that involve in reliability are TCP/IP layer, network layer and MAC layer. Therefore, we integrated TCP\IP layer and routing layer to improve the reliability of the network. Buffer level information of each node is observed in route discovery process (control packet transmission and reception). Based on the control packet information data transmission is done with respect to the buffer level of nodes. This highly balances the data traffic and improves the network reliability. In critical monitoring applications (monitoring the health conditions of patients) reliable data transfer is the major challenge. When packet loss occurs, then ACK packets are required for re-transmissions. Multi-retransmissions in network severely affect the

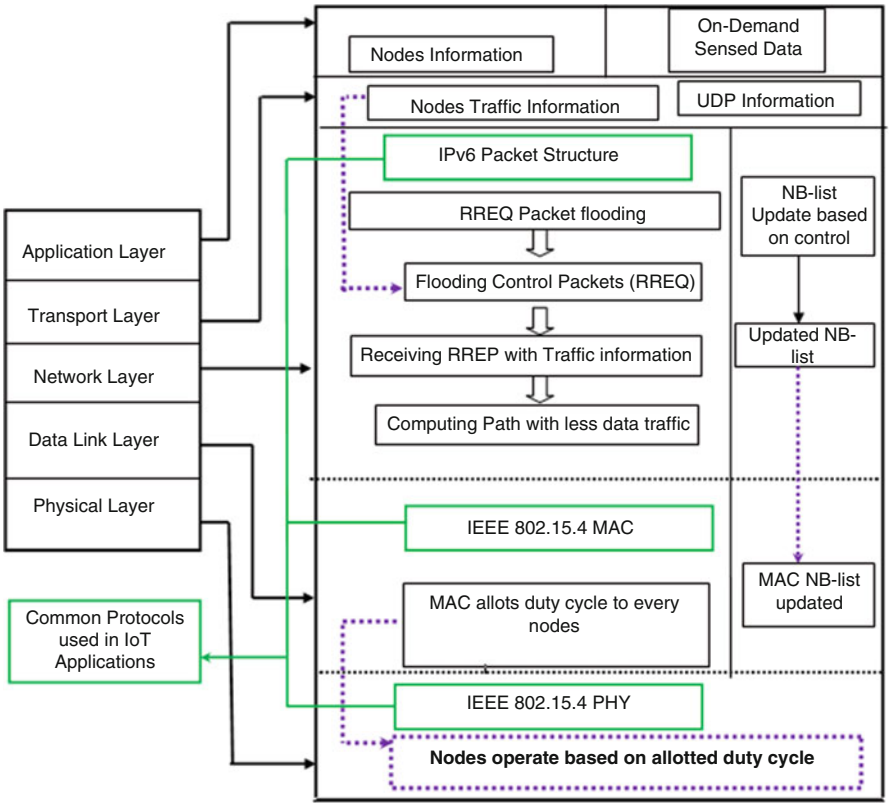



**Fig. 1.3** Reliable network architecture for e-health applications

network reliability and degrade the health care service. Thus, we propose the reliable network architecture by integrating the TCP/IP layer and network layer.

Figure 1.3 describes reliable network architecture that is suitable for e-health application, it is the operational flow of proposed network model. In this architecture, features of TCP/IP technique and routing technique are integrated. Initially traffic information of network is attained by the TCP/IP technique and later routing technique utilizes the traffic information and finds reliable path for data transmission.

### *1.4.2 Environmental Monitoring*

Environmental monitoring applications collect the environment related information (temperature, humidity, smoke, fire, etc.) and transmit it to base station. Environmental monitoring includes landslide monitoring, forest fire monitoring, flood monitoring, temperature monitoring, snowfall monitoring, irrigation monitoring,

tsunami monitoring, wild life monitoring, industrial smoke monitoring, pedestrian monitoring, etc. All these applications are situated in remote area, thus devices used in these applications are battery operated. When the battery drains out its energy, battery replacement need to be initiated, frequent battery replacement in such harsh environment is impossible. In flat based network scenario, the data traffic is from sensor nodes to base station, therefore the nodes near to base station hold bulk amount of data traffic, when nodes near to base station are burdened (data traffic overload), then they drain out its power in short span of time [14]. Thus, balanced energy utilization is the major requirement in environmental monitoring applications.

Based on this observation, we designed energy efficient network architecture that integrates routing and node placement technique. The reason of integrating these two techniques for environmental monitoring applications is they are considered as the effective techniques, to improve the energy efficiency of the network. In node placement technique density of nodes are varied based on data traffic, which balances the data traffic load and prevents energy hole issue (quick node death). In routing technique energy and reliability related parameters are included to compute energy efficient and reliable path. Residual energy is considered for energy monitoring and expected transmission count (ETX) is considered to find the reliable links, therefore effective combination of these two parameters finds energy efficient and reliable route. Node placement technique takes care of data traffic, routing technique monitors energy consumption and finds reliable links. Thus, all these features are included in single network architecture to achieve balanced energy utilization in energy constrained network environment. In this network architecture, routing technique and MAC nased power control technique are both integrated in network initializing phase. The idea at the heels of this work is utilizing features of node placement technique and routing technique in single network architecture to improve the energy efficiency of the network.

Figure 1.4 describes the energy efficient network architecture for environmental monitoring applications, it is the operational flow of proposed network architecture. In this architecture, node placement technique manages the data traffic and routing finds energy efficient and reliable path, these features are integrated in single network architecture. In network setup phase node placement is implemented and in network initialization phase routing is implemented.

### *1.4.3 Industrial Automation*

Industrial automation is making industrial systems automated by the aid of computing assisted technology and communication technology. Problems in many industrial equipment are diagnosed and reported by the help of wireless enabled industrial automation systems [3]. Of late, many industries started talking about wireless enabled automation systems in industries. Many discrete manufacturers are concentrating on implementing wireless infrastructure in industries to monitor and

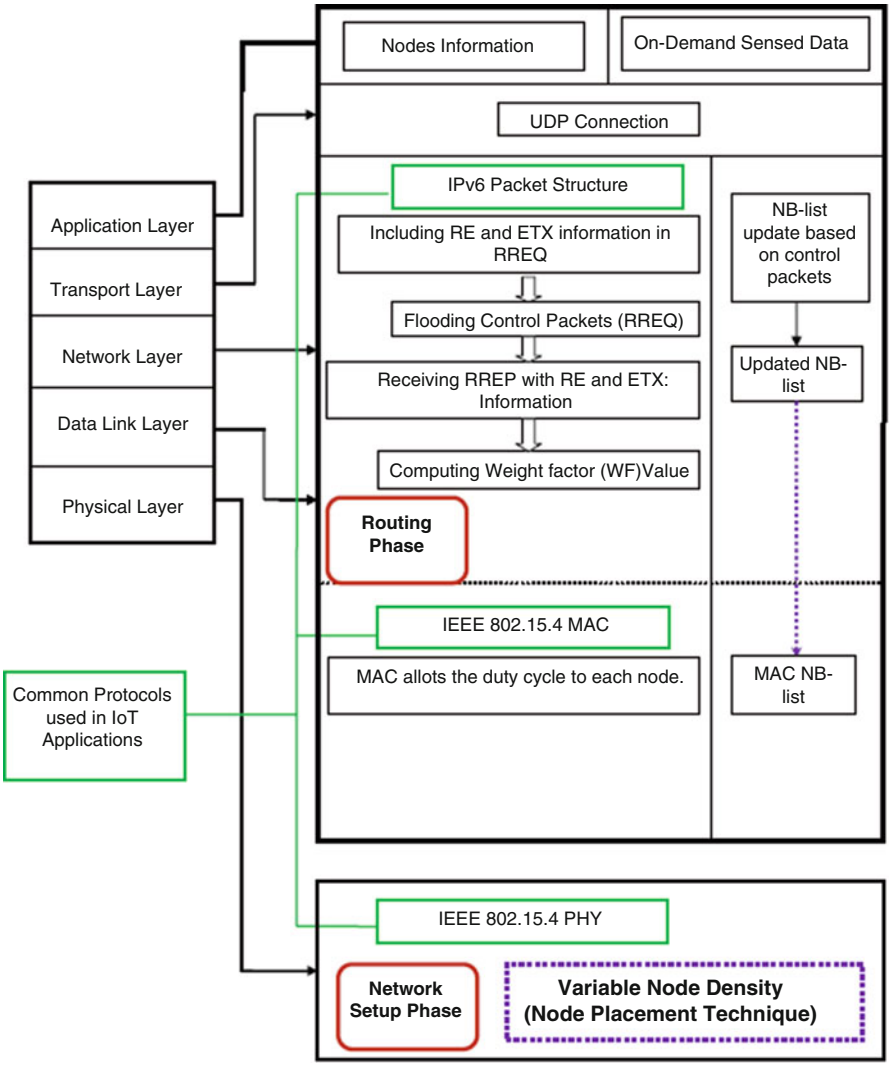



**Fig. 1.4** Energy efficient network architecture for environmental monitoring applications

maintain the plants. In many industries, parameters such as pressure, temperature and flow are observed, whereas equipment related information such as equipment conditions and efficiency of the equipment are not monitored [15]. Thus, an effective wireless network infrastructure for monitoring production as well as equipment conditions is the major requirement. Implementing wired network infrastructure is also possible in industrial environment, but designing, wiring, cost of the components, etc. are difficult in wired network environment, therefore implementing wired network infrastructure is very expensive [15], whereas wireless technology

is flexible in all aspects such as implementation, cost, scalability, and mobility. Wireless enabled industrial automation system provides sensing as well as actuation.

Industrial activity monitoring is also the critical monitoring applications (e.g. industrial boiler monitoring). Monitoring the working conditions of industrial instruments is the prime task in industrial monitoring applications. Therefore, reliability and QoS need to be maintained in effective way in this application. For this application, we propose a network architecture that integrates network layer and MAC layer to improve the QoS and reliability of the network. Based on the routing information transmission range of nodes is varied by MAC based power control technique. Based on this integration every node obtains its optimum transmission range. In this network architecture, routing technique and MAC based power control technique are both integrated in network initialization phase. The reason of integrating routing and MAC based power control technique is reliability and QoS related information are observed from routing technique. Based on this routing information MAC adjusts its transmission power, which says every node achieves its reliable transmission power.

Figure 1.5 describes the QoS aware network architecture that is suitable for industrial automation applications. In most of the industrial applications QoS aware data transmission is the major challenge. In proposed network architecture, QoS related information for every node is achieved by routing technique. Later, MAC based power control technique utilizes the received signal strength and adjusts the transmission and reception power, to achieve QoS aware data transfer.

### *1.4.4 Smart Grid*

Power grid is the electrical grid that delivers electricity to various infrastructures (office, houses, industries, huge apartments, etc.). Distributing electricity from power plants to consumers in smarter way (efficient way) is referred to as smart grid. Electric power consumption of every consumers is measured by the electric meters, thus implementing smart meters in power generation as well as power distribution systems performs sensing as well as actuation to enhance the grid operation [16]. Smart devices collect the electricity related information between power generation and distribution systems and it sends these information to analytical tools for balanced utilization of power, as well as it helps in identifying issues in power distribution systems and power wastage in power distribution systems [16]. For such network scenario, power distribution should be maintained by reliable network infrastructure. Thus, we come with reliable cross layer design that integrates routing technique and MAC based power control technique to improve the network reliability. In routing mechanism, reliability related parameter called received signal strength (RSSI) is included for route discovery process, later MAC based power control technique utilizes the routing information and adjusts its transmission power based on the routing information. In this network architecture transmission ranges are adjusted based on reliability (RSSI) of the network. Thus, every node obtains

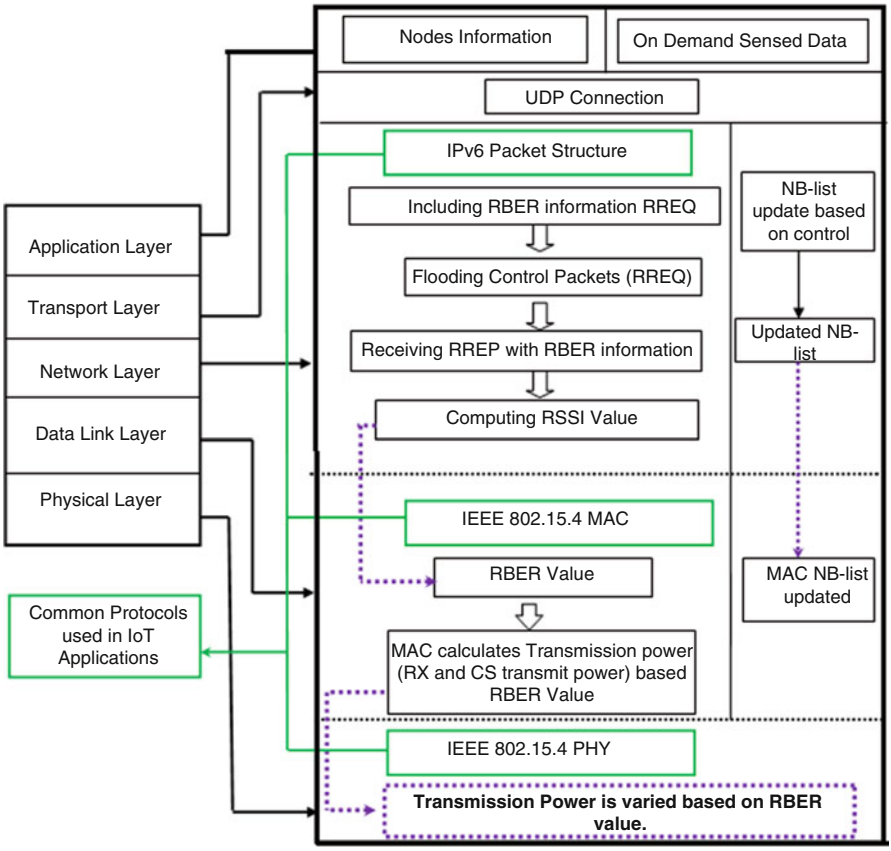



**Fig. 1.5** QoS aware network architecture for industrial automation applications

its own capable transmission range to achieve reliability of the network. Figure 1.6 describes the reliable network architecture for smart grid applications. Reliability is the prime need for smart grid applications. In proposed network architecture, RSS (received signal strength) information for every node is attained by routing technique. Later, MAC based power control technique utilizes the received signal strength and adjusts the transmission and reception power, to achieve reliable data transfer. Table 1.1 describes the features of proposed network architectures.

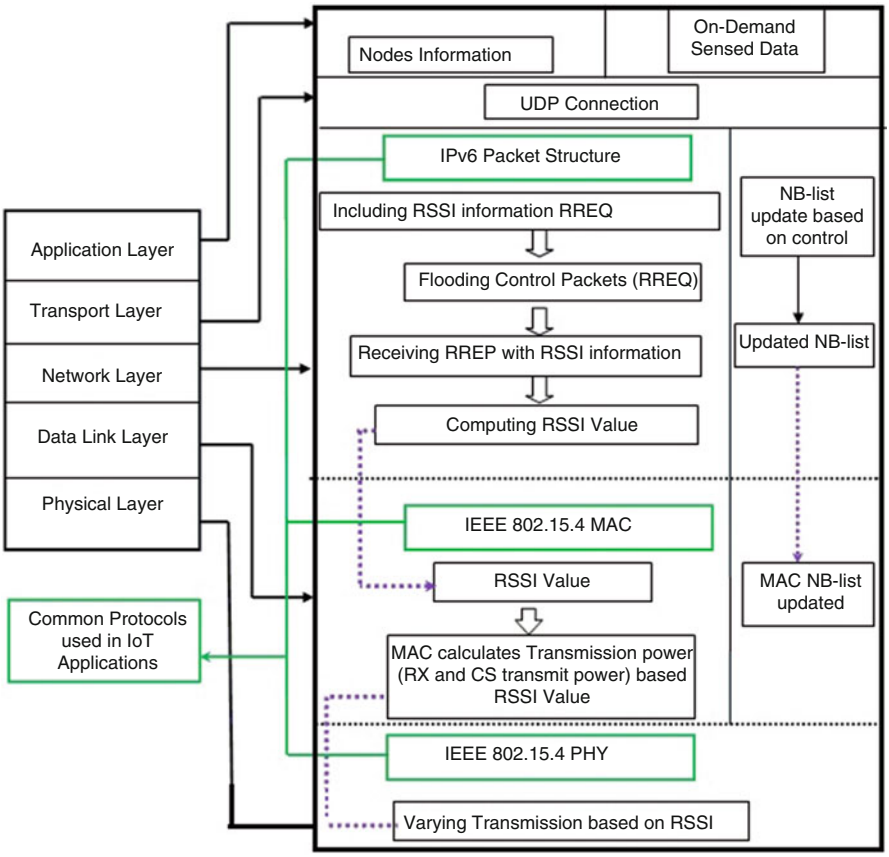



**Fig. 1.6** Reliable network architecture for smart grid applications

**Table 1.1** Difference between proposed network architectures

| Applications | Challenges | Integration |
|---|---|---|
| E-health | Reliable data transfer | Network layer and TCP/IP layer are integrated |
| Environmental monitoring | Network lifetime and reliable communication | Network layer and physical layer are integrated |
| Industrial automation | QoS aware data transfer | Network layer and MAC layer are integrated |
| Smart grid | Reliability | Network layer and MAC layer are integrated |

## 1.5 Suitability of Proposed Network Architectures for IoT Scenario and Network Assumptions

IPv6 address, IEEE 802.15.4, RPL (routing protocol for low power and lossy networks) and 6LoWPAN are included in network architecture to obtain suitability of proposed architectures with real-time IoT applications.

The basic assumptions of proposed network architectures are as follows:

- Sensor nodes are deployed in random manner.
- All the nodes are battery sourced.
- Nodes are aware of the routing information (energy related information and reliability related information).
- Base station is not limited by energy.

## 1.6 Conclusion

Internet of Things is practiced by numerous applications. Every application has specific network level issues based on their network environment. Most of the IoT applications are situated in remote environment, where battery operated devices are possible devices to collect environmental information and low power radio links are the possible medium to connect low power devices. Therefore, providing active communication in resource constrained network environment is the major challenge in low power IoT network. This chapter elaborates the resource constrained nature of IoT network and describes the issues that commonly occur in IoT network. We analyzed the major network level challenges that are faced by various IoT applications. Based on these challenges, we designed network architectures that satisfy application specific network requirements. The proposed network architectures are constructed by integrating the features of various layers together. These envisioned network architectures give the clear idea to optimize the low power IoT networks.

## References

1. The Internet of Things ITU internet reports, November 2005
2. Lee GM, Park J, Kong N, Crespi N, Chong I (2012) The internet of things – concept and problem statement. Internet Research Task Force
3. Vasseur J-P, Dunkels A (2010) Interconnecting smart objects with IP. Elsevier, New York
4. IoT World Forum Reference Model. http://www.iotwf.com/resources
5. GilKo J, Terzis A, Dawson-Haggerty S, Culler DE, Hui JW, Levis P (2011) Connecting low-power and Lossy networks to the internet. IEEE Commun Mag 49(4):96–101
6. Boukerche A (2008) Algorithms and protocols for wireless sensor networks. Wiley-IEEE Press, Hoboken

7. Rajendran V, Obraczka K, Garcia-Luna-Aceves JJ (2003) Energy-efficient, collision-free medium access control for wireless sensor networks. In: Proceedings of ACM sensor system 03, Los Angeles, California
8. Jones CE, Sivalingam KM, Agrwal P, Chen JC (2001) A survey of energy efficient network protocols for wireless networks. Wireless Networks, pp 343–358
9. Ben-Othman J, Yahya B (2010) Energy efficient and QoS based routing protocol for wireless sensor networks. J Parallel Distrib Comput 70:849–857
10. Larzon LA, Bodin U, Schelen O (2002) Hints and notifications for wireless links. In: Wireless communications and networking conference, WCNC2002, pp 635–641
11. Vuran MC, Akyildiz IF (2010) Xlp: a cross-layer protocol for efficient communication in wireless sensor networks. IEEE Trans Mob Comput 9:1578–1591
12. Park P, Fischione C, Bonivento A, Johansson KH, Sangiovanni- Vincent A (2011) Breath: an adaptive protocol for industrial control applications using wireless sensor networks. IEEE Trans Mob Comput 10:821–838
13. Maglogiannis I (2009) Introducing intelligence in electronic healthcare systems: state of the art and future trends, artificial intelligence an international perspective. Springer, Heidelberg, pp 71–90
14. Ferdoush S, Li X (2014) Wireless sensor network system design using raspberry Pi and Arduino for environmental monitoring applications. Procedia Comput Sci 34:103–110
15. Noble D (2017) Forces of production: a social history of industrial automation. Routledge, Abingdon
16. Farhangi H (2010) The path of the smart grid. IEEE Power Energy Mag 8:18–28

# Chapter 2
# A Measurement Study of Campus WiFi Networks Using WiFiTracer


**Chengwei Zhang, Xiaojun Hei, and Brahim Bensaou**

**Abstract** Highly dense and large-scale WiFi networks have been widely deployed in public areas to provide cost-effective high-speed wireless Internet access for mobile end users. This emerging practice has been leading to a severe spectrum usage overlap and channel interference between colocated WiFi networks. To understand the characteristics of highly dense WiFi networks, we conduct a measurement study of campus WiFi networks in this chapter. First, we instrument an Android App to sense WiFi access points (APs) to characterize WiFi networks in campus areas, including WiFi spectrum and channel usage, AP density, network distribution, and so on. Our measurement results demonstrate that a large number of WiFi APs have been widely deployed on campus, and about 80% of the total APs occupy the 2.4 GHz band, whereas the remainder part are the higher frequency 5 GHz APs, commonly used by public WiFi networks. The spectrum overlap and channel interference in the 2.4 GHz band is much more severe than that in the 5 GHz band. Then, extra WiFi connection measurements are conducted at selected areas with well-deployed campus WiFi networks, to understand WiFi connection characteristics while pedestrians are moving around in the coverage of the WiFi networks. By harvesting data from voluntary Android smart phone users, the connection setup time composed of Authentication–Association (AA) time, handshake time, and IP acquisition time is found to be generally affected by various factors, such as AP density, RSSI levels, etc. To achieve load balancing with reduced interference and higher WiFi network performance, this field measurement study may provide guidelines to design the next generation software-defined WiFi networks.

C. Zhang · X. Hei (✉)
Huazhong University of Science and Technology, Wuhan, China
e-mail: heixj@hust.edu.cn

B. Bensaou
The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_2

## 2.1 Introduction

WiFi networks have been providing a cost-effective high-speed wireless network access in the past decades. WiFi-based wireless local area networks are widely deployed on Edges of Internet for convenient user access due to the following three benefits: (1) simple technical implementation, (2) low-cost network construction, and (3) high-bandwidth wireless links [1]. Although only a limited number of user clients are supported by a single access point (AP) for the WiFi original design, WiFi network with multiple APs, such as a hotzone [2], has been increasing for supporting Internet access with a large number of clients in a relative large area [3, 4]. WiFi networks serving as the major network components have been envisioned for constructing smart city and even smart country [5].

The communication and entertainment paradigm in people's daily life has been gradually reshaped by the rapid penetration of smart phones. A variety of micro sensors have been integrated in modern smartphones, including accelerometers, gyroscopes, magnetometers, light sensors, global navigation satellite system (GNSS) as well as Bluetooth and WiFi transceiver modules [6, 7], which can cooperate with monitoring, positioning, and navigating applications. Due to the pervasive usage of smart phones, together with the cooperative sensing capability and users' mobility, smart phones have been evolving from ordinary mobile devices into measurement enablers [8, 9]. Users can carry smartphones around during their daily lives for measuring, collecting, and preprocessing data of user activities with powerful sensor and microprocessors embedded in smartphones. Recently, various research projects and applications deeply relying on smartphones and user mobilities have emerged for different purposes, such as smartphone-based indoor position system [10], recording physiological indexes of mobile users [11], monitoring user behavior [12], tracking the air quality of the urban environment [13], and so on. Mobile measurement applications running on smartphones carried by a large number of users, which can perform measurements individually and conduct analysis collaboratively, from the mobile crowd sensing (MCS) measurement [14, 15]. The MCS can fully exploit the limited resources of individual smart phones, and conveniently deploy real and randomized measurement experiments rather than well-planned experiments in a large scale [14, 15]. Lane et al. [16] and Xiao et al. [17] studied the data transmission efficiency and energy consumption problem of MCS. In particular, recent measurements [18, 19] have shown that significant energy has been consumed by wireless transceiver modules (WiFi and 3G/4G) of mobile devices during data transmission. These field measurement studies have greatly pushed forward the innovation of mobile cloud transmission systems [20, 21] to shift the heavy energy-hungry services for mobile Apps to the remote clouds, instead of local mobile devices.

In this chapter, we are motivated to conduct a measurement study to characterize the spectrum interference and the connection bottleneck on campus WiFi networks using a MCS approach. We developed a MCS platform for tracking the channel usage and connection bottleneck of campus WiFi APs. An augmented Android

measurement tool, named WiFiTracer, can automatically probe, maintain, and upload detected WiFi APs' information through smart phones from volunteers. For the large-scale WiFi measurement construction, student participants as anonymous users have been invited to perform random movement in various ways (driving, jogging, and walking) on the campus with the measurement APP running on smartphones for abundant measurement data collection. During this crowd sensing measurement process, we conducted various experiments to quantify the connection time for a public campus WLAN. The major results from these experiments are summarized as follows:

1. We first summerized the WiFi dataset on our campus area. Our results show that there are considerable and high-density WiFi APs maintained on the campus area. Over 10,000 WiFi APs and more than 7000 distinct WiFi networks have been detected.
2. We characterized a public campus WiFi network quantitatively. Our results show that more than 70% measurement areas have been covered by this public campus WiFi WLAN. We quantified the interference of this public campus WLAN with its nearby private WiFi networks. Extra experiments were conducted to compare campus WiFi networks deployed indoors and outdoors. Measurement results also show that the dynamic frequency selection (DFS) feature of WiFi APs is not enabled in the general circumstance.
3. We conducted the WiFi connection setup time on the public campus WiFi networks during the MCS measurement process. The connection setup time deviates significantly on the different mobile devices, ranging from the tens of milliseconds to tens of seconds.

In this chapter, we first introduce the concept of MCS and recent research progress. Then, we propose and implement a crowd sensing measurement platform. Next, we dissect the WiFi connection setup process for public WiFi networks. Afterwards, we report our measurement results mainly on two aspects: channel interference and connection time. Finally, we conclude this chapter. The measurement results demonstrate the necessity of a configurable and manageable software-defined WiFi network that can dynamically adjust WiFi channels based on the current network states to achieve load balancing with reduced interference and improved WiFi network performance.

## 2.2 WiFi Measurement Platform

Public campus WiFi networks are widely deployed at public locations through the campus; for understanding the characteristics of WiFi networks, like channel usage, AP density, connection time, etc., we are motivated to propose a general WiFi sensing measurement framework using the MCS way as shown in Fig. 2.1 to provision the data transmission and sharing of measurement results. The measurement platform can conveniently cooperate with particular WiFi measurement modules to inspect various metrics of WiFi networks.

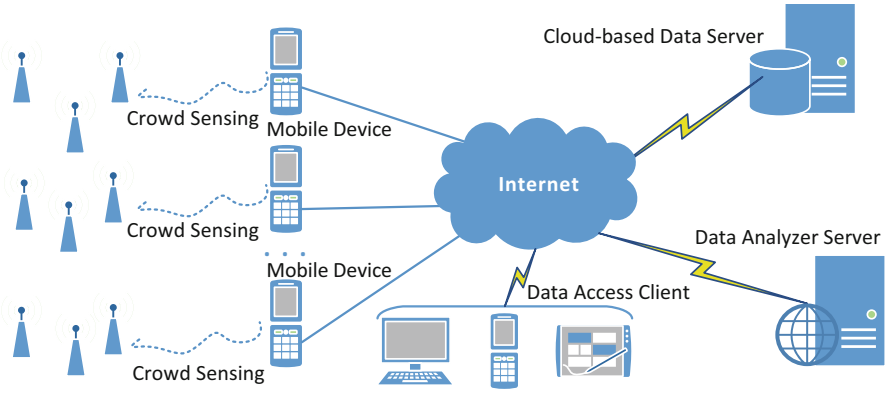



**Fig. 2.1** WiFi measurement architecture using mobile crowd sensing (MCS)

### *2.2.1 Measurement Framework Overview*

The proposed MCS platform, consisting of three major modules including data acquisition, collection, and analysis, is constructed in Fig. 2.1. Smartphones equipped with WiFiTracer can successfully turn these typical mobile devices into WiFi measurement tools for the nearby WiFi information harvest and preprocessing. WiFi measurement data has been collected and formatted locally on mobile devices by data acquisition module. Preprocessed data from varied mobile devices will be collected and analyzed as a repository hosted on a cloud platform by the data collection module. When the volunteers complete the measurements, Android app or service running on the framework will automatically upload the local results to the server with timestamps and user tags. The reward module is used to share the available WiFi information as an incentive for participants. The server provides information about available WiFi networks close to the end users and the possible WiFi network access at the current location, which can be displayed to users through web pages or the client app. Increasing individual users are willing to contribute the measured data and acquire better network connection performance potentially based on the incentive crowd sensing way.

### *2.2.2 WiFiTracer Architecture*

WiFiTracer is an Android mobile app which can run on different Android smartphones to explore and aggregate the WiFi network information from the WiFi client side. Its software architecture is constructed with four major layers as shown in Fig. 2.2, including application interface, task module, system libraries, and Android system control. Each layer implements the individual function and works collaboratively to transform a normal mobile phone into a general portable measurement device.

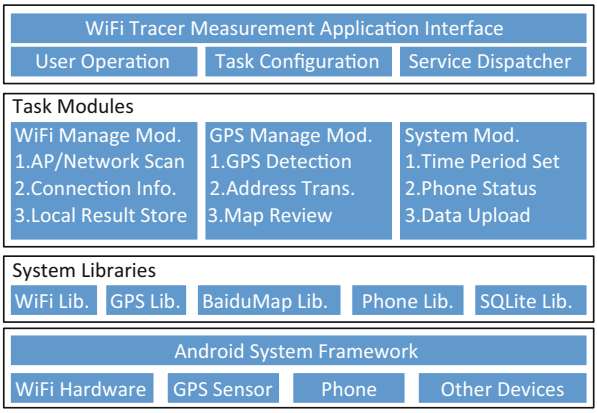



**Fig. 2.2** The software architecture of WiFiTracer

- **Android system control layer:** It provides the necessary physical device drivers and interfaces for the Android applications and services. Through this layer, the high-level applications can apply a general unified way to access and manage various sensors, such as WiFi transceiver, GPS sensor, and so on.
- **System library layer:** The system dependence libraries, including WiFi Lib., GPS Lib. et al., extend the management for the Android resources. Those third-party libraries can provide additional functions and data sources than those generic ones. The BaiduMap library, as an extra map provider library, offers more accurate GPS location service and various data visualization methods on the map.
- **Task module layer:** This layer, designed as the main module of WiFiTracer, consists of three individual components: (1) WiFi management module, (2) GPS management module, and (3) application configuration module. The WiFi and GPS management components are used to access the corresponding drivers and obtain the raw measurement results from the low-level Android devices, such as WiFi basic information and GPS raw data. The configuration component offers a flexible interface for higher layers and supports multidimensional measurement tasks of WiFiTracer.
- **User interface layer:** This layer provides a friendly operation graphic interface for end users. A user can configure parameters of the tool, schedule tasks, and manage low-level sensors of the tool.

#### 2.2.2.1 Measurement Sample

WiFi APs may be sensed repeatedly for multiple times during the measurement process; therefore, the same WiFi APs may be tagged with different timestamps and locations during measurement. For the purpose of describing the measurement

results conveniently, a measurement sample by a single smartphone can be summarized as follows:

$$o^i_{bssid} = \{timestamp(ts), round, bssid, ssid, channel, rssi, capabilities, gps\}. \tag{2.1}$$

$o^i_{bssid}$ represents the measurement data of WiFi AP which appears for the $i$-th time. During each sensing round, $ts$, $round$ and $gps$ represent the general round information of WiFi measurement, where $ts$ represents the accurate sensing time on the smartphone, $round$ records the time number of measurement, and $gps$ stores the current measurement location. Each WiFi AP's basic information, including BSSID (as $bssid$), SSID (as $ssid$), the channel frequency (as $channel$), the signal strength RSSI (as $rssi$), and the security mechanism (as $capabilities$), once sensed during the measurement process, will be obtained and merged with the general round information to formulate a unique and complete measurement sample set $o^i_{bssid}$. Therefore, the whole dataset of WiFi APs' bssid is detected on MCS measurement process as $S$; the results of the same WiFi APs (using $bssid$ as the identifier) compose independent result set $O_{bssid}$:

$$O_{bssid} = \{o^i_{bssid}, i \in [1, +\infty], bssid \subseteq S\}. \tag{2.2}$$

MCS allows using different devices to harvest WiFi network information. The measurement result sets of different devices can be expressed in Eq. (2.3). The results of each device can be formed as an independent measurement result set which can be identified by the unique device ID. The measurement data collected by WiFiTracer are uploaded to the remote server through the Internet and then are analyzed afterwards.

$$OD_{device} = \{\{id, deviceid, O_{bssid_i}\}, i \in [1, +\infty], bssid_i \subseteq S\}. \tag{2.3}$$

### 2.2.3 Measurement Sampling Procedure

With the supplement of the MCS mechanism, WiFiTracer can cooperate with various Android mobile devices for WiFi measurement. The tool implements an optimized scanning procedure as shown in Algorithm 1, which can significantly improve the efficiency and accuracy of measurements and avoid unnecessary multiple sensing rounds on the same locations.

WiFiTracer tracks the dynamics of WiFi APs periodically (such as 10 s) while the user is in moving states. During the WiFi measurement process, WiFiTracer obtains and computes the distances between the current measurement location and the previous measurement location. Once the computed distance is larger than a threshold (10 m as default), the tool will activate the scanning process to detect the nearby WiFi APs tagged with timestamps and GPS coordinates to form the measurement metadata. Results are then stored locally in the Android SQLite database, and will be uploaded in the cloud repository for further analysis.

**Initialization:** Smartphone, WiFi transceiver, and GPS sensor initialized ;
**Data**: $deviceinfo \leftarrow$ deviceinformation, $origpos \leftarrow$ currentGPSposition,
**Data**: $period \leftarrow$ userconfiguration, default : 10s,
**Data**: $mindistance \leftarrow$ userconfiguration, default : 10m ;
**Result**: WiFi measurement dataset on variant mobile devices: $O_{deviceid}$ ;
**while** *scanning service is not stopped* **do**
  **if** $scanperoid = peroid$ **then**
    $currentpos \leftarrow$ currentGPSposition ;
    **if** $distance(origpos, currentpos) > mindistance$ **then**
      activate the WiFi scanning process; scan the WiFi APs nearby ;
      $scantime \leftarrow$ currenttimestamp, $scancount \leftarrow$ currentscancounter ;
      record $scanresult : (bssid, ssid, frequency, rssi, capabilities)$ ;
      build entity: $o^i_{bssid} : (scantime, scancount, deviceinfo, scanresult, currpos)$ ;
      store the dataset $O$ and upload to the remote server ;
      terminate the current service ;
    **else**
      exit the current service, start a new round timer for scanning ;
      **continue ;**

    $origpos \leftarrow currpos, scancount = scancount + 1$ ;
    start a new round timer for scanning ;
    **continue ;**
  **end**
**end**
User terminates the application, and stop all the functions.

**Algorithm 1:** Sketch of the WiFiTracer sensing procedure

## 2.3 Sensing Result Analysis

MCS supports various mobile devices in collaborative measurement and each mobile device becomes a distinct end-point measurement tool. The measurement device cooperating with user's mobility translates the whole experiment to a randomized distributive measurement process, and the data storage and computation on the cloud offers convenient data sharing and analysis among all measurement clients. The main campus of our university has been chosen as the main experiment area to launch the WiFi measurement. Well-performed Android smartphones, such as HUAWEI Honor7, ZTE Nubia Z7, etc., have been carefully selected as measurement devices to operate WiFiTracer tool, and all devices can perform smoothly on WiFi standard frequencies in both 2.4 and 5 GHz bands for WiFi standard protocols such as 802.11 a/b/g/n. Participants as anonymous users have been invited to perform random movement in various ways (driving, jogging and walking) on the campus with the measurement APP running on smartphones for abundant measurement data collection.

Participants were requested to perform randomized movement on the main roads with a relative low speed ($\leq$20 km/h) during each measurement process and almost took 1.5–2 h to traverse the whole campus measurement areas. In order to cover the whole measurement areas with sufficient and accurate WiFi metadata, the entire measurement experiments have been lasted for around 30 days and the total measurement time is up to almost 100 h. Due to different WiFi networks have variant radiation coverages, the proposed experiments assumed that most of indoor WiFi APs and networks were visible on the main roads and could be obtained by the WiFiTracer.

**Table 2.1** Measurement dataset

| Metric | Amount |
|---|---|
| Scan times | 20,210 |
| Data samples | 534,210 |
| Independent areas by GPS | 13,065 |
| Number of distinct WiFi APs | 11,380 |
| Number of distinct WiFi networks | 7483 |
| Number of 2.4 GHz APs | 10,390 |
| Number of 5 GHz APs | 1988 |
| Number of public WiFi APs | 2893 |

### *2.3.1 Basic WiFi Dataset*

By Eq. (2.2), each WiFi AP detected by smartphone applications can be presented as an independent measuring result set which records the location information and current signal strength (RSSI). The values of WiFi's RSSI have a variance relation to the distance from the measurement node to the AP [22], which implies that a smaller distance leads to a stronger signal. It is possible that we can choose the APs' results with the maximized RSSI value and utilize the GPS information to estimate the real AP locations.

The distribution of WiFi APs is primarily on roads or near roads because the measurements were conducted along the main roads of the campus covered by WiFi APs with intensive quantities. Table 2.1 shows the WiFi sensing measurement dataset that over 10,000 independent WiFi APs have been successfully sensed and most of them are private. Private WiFi networks constructed by independent APs can provide small range network access with passwords or other authentications. With dense deployment of WiFi networks, it has become an emerging problem for WiFi networks to reduce the spectrum interference from other WiFi APs.

### *2.3.2 General Analysis of WiFi Networks*

#### 2.3.2.1 Heatmap of WiFi APs' Distribution

Figure 2.3 shows the heatmap of the WiFi APs distribution, where red areas suggest high density of WiFi APs deployed. Demonstrate that there is a strong correlation between the high-density WiFi networks. The circled areas $1 \sim 6$ are official areas, teaching areas, and living areas where people spend most of daily time. Covered with orange colors are focused on crossroads or intersections of roads. One reason is that the intersections are the connections of different roads which potentially have more chances to measure than normal areas during the random movement under crowd sensing measurements; the other one is that WiFi APs usually are deployed with buildings. Hence, Fig. 2.3 indicates the WiFi density near intersections heavier than normal areas.


**Fig. 2.3** WiFi AP heatmap

### 2.3.2.2 WiFi Channel Usage

Densely deployed WiFi APs would potentially result in severe spectrum interferences in WiFi channels. Figure 2.4 depicts channel usages of WiFi networks for both 2.4 GHz band and 5 GHz band. Results demonstrate that 2.4 GHz band is the main working band occupied almost 80% in the sensing dataset while 5 GHz band only accounts for 20%. Private WiFi networks seldom work on 5 GHz band, and over 95% 5 GHz APs are used for public WiFi networks. IEEE WiFi standard organization encourages more than 15 channels in 5 GHz band for high-speed WiFi networks, whereas part of them are allowed to use in different countries, such as only channels 149–151, 161, and 165 are permitted as legitimate channels in China.

Figure 2.4 shows various channel usages in percentages. Results demonstrate that channels 1, 6, and 11 in 2.4 GHz are the most popularly used channels among all channels. The reasonable explanation is that WiFi manufactures usually set the 2.4 GHz WiFi appliances default in these independent channels to avoid the adjacent channel interference in practical applications. However, even these three channels are completely independent of each other in the spectrum, and due to the fact that WiFi users rarely change these default channel settings, the overuse of these channels would greatly increase the co-channel overlaps and interferences. On the contrary, channels in 5 GHz are completed isolated from each other and result none of adjacent channel interferences.

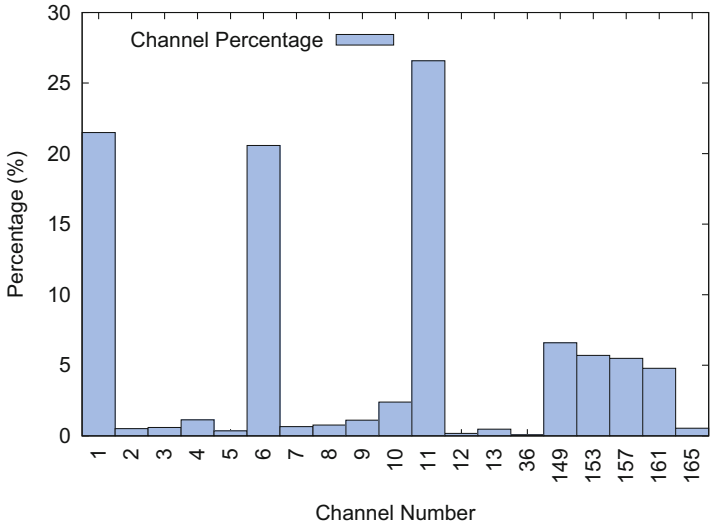



**Fig. 2.4** WiFi channel utilization

#### 2.3.2.3 WiFi AP Hardware Legality

Variant WiFi devices have been widely put into commercial use for the simplicity and convenience of WiFi network. Figure 2.5 presents top 10 WiFi manufacturers used in measurement areas. Wireless devices made by TP-Link dominates over 20%, for their high price–quality ratio of small home routers. Over 35% WiFi devices marked with "Unknown" cannot find the corresponding manufactures through the registered manufacturer information provided by IEEE Standard Association (ISA) [23]. Two reasons can explain why there are so many "Unknown" devices. One is that the manufacturer list is not updated in time by ISA; the other is that some factories do not register in ISA at all and produce WiFi devices illegally and privately.

Table 2.2 shows channel usages of "Unknown" WiFi devices, where the percentage of channel 11 is nearly double of channels 1 and 6. Therefore, it can be inferred that these anonymous manufactures choose the highest channel 11 in 2.4 GHz to avoid the interferences with other commercial products' channels. However, joint consideration with Fig. 2.4, channel 11 has the highest utilization among all the channels for its overuse by a large quantity of "Unknown" WiFi devices, which could potentially result much severer co-channel interference than others.

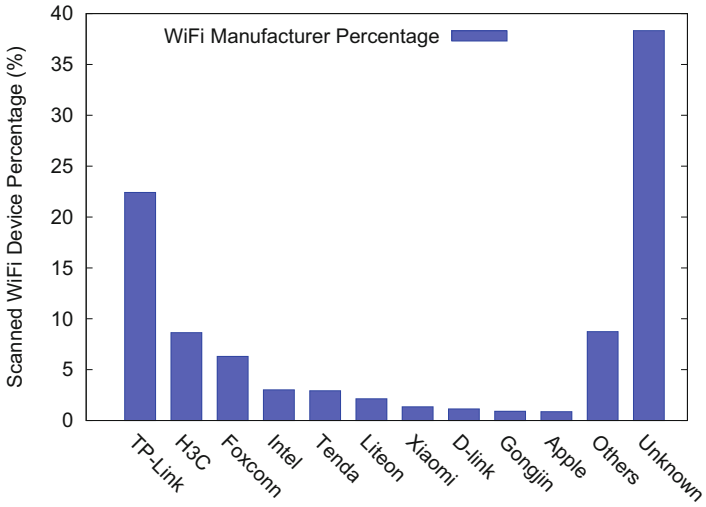



**Fig. 2.5** WiFi device usage of different manufacturers

**Table 2.2** Channel usage of "Unknown" WiFi devices

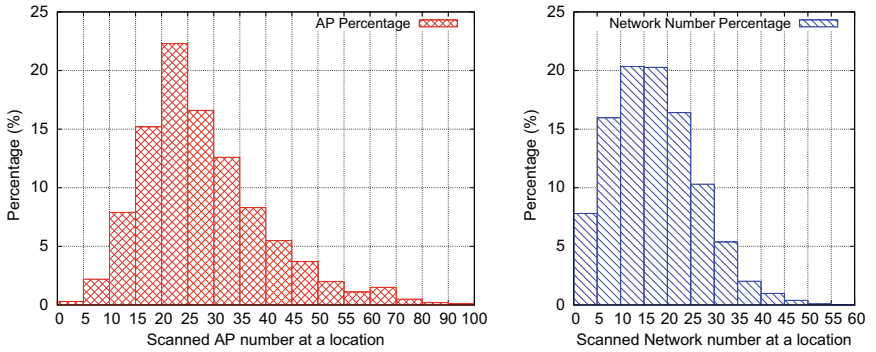
| Channel | Number | PCT.(%) |
|---|---|---|
| 1 | 958 | 24.0 |
| 6 | 983 | 24.7 |
| 11 | 1912 | 48.0 |
| Others | 130 | 3.3 |

#### 2.3.2.4 Density of WiFi APs and Networks

AP densities at distinct measurement locations are shown in Fig. 2.6. Results in Fig. 2.6a indicate that over 15 individual WiFi APs have been detected in almost 90% measurement areas and over 100 independent APs have been scanned in some extremely high-density areas. Under the Extended Service Set(ESS) model, independent APs may construct a wide-range WiFi network with the same identified network name. Therefore, densities of WiFi networks would exhibit different characteristics from densities of WiFi APs in the same areas. Figure 2.6b demonstrates independent WiFi networks at various measurement locations with percentages. Over 10 independent WiFi networks have been detected in about 80% measurement areas. Results from densities of WiFi APs and networks illustrate the approximation to the normal distributions.

#### 2.3.2.5 Utilization in 5 GHz Band

Figure 2.4 illustrates that WiFi channel utilization meets the 80/20 rule, 80% for 2.4 GHz band and 20% for 5 GHz band. Table 2.3 shows that only 5% of 5 GHz




(a) (b)

**Fig. 2.6** Density of WiFi APs and distinct networks. (**a**) Density statistics of WiFi APs. (**b**) Density statistics of WiFi networks

**Table 2.3** Usage of 5 GHz WiFi APs

| Type | Private APs | Public APs | Total |
|---|---|---|---|
| Number | 100 | 1888 | 1988 |
| PCT.(%) | 5 | 95 | 100 |

APs are used to construct private WiFi networks, even they have better performance and less interferences than 2.4 GHz APs; whereas 95% ones are used by public WiFi networks for high quality access.

### *2.3.3 Characterizing Public Campus WiFi Networks*

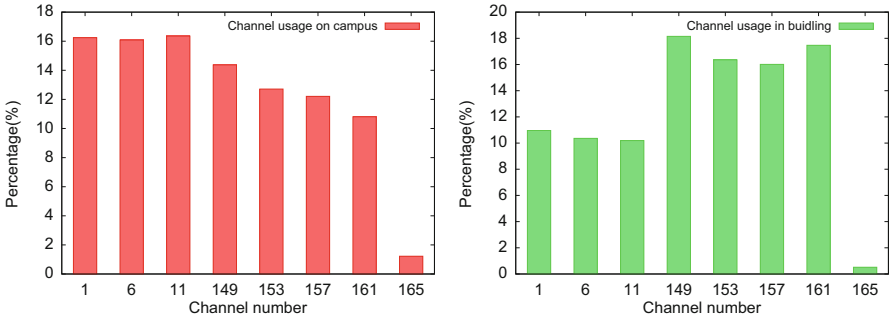
Through the enhanced scanning measurements for public WiFi networks, more than 7000 distinct public campus WiFi APs have been successfully scanned and recorded. Considering the maximum signal strength (RSSI) received at GPS locations, distributions and coexistent characteristics between public and private WiFi networks can be merged on the real areas by heatmaps, measurement results depicted in Fig. 2.7 show that public networks and private networks appear to be complementary.

#### 2.3.3.1 Indoor vs. Outdoor Channel Usage

Figure 2.8 shows the usage of channels in campus (outdoor/indoor) WiFi networks differentiate distinctly from private networks. The numbers of the occupied 2.4 GHz and 5 GHz bands are similar and the channel bands are distributed evenly except that channel 165 has fewer WiFi APs. We infer that public WiFi networks adopted the balanced AP deployment strategy to make frequency bands evenly distributed

(a) (b)

**Fig. 2.7** Spatial spread density statistics of WiFi networks. (**a**) Density of public WiFi. (**b**) Density of private WiFi




(a) (b)

**Fig. 2.8** Channel spatial distribution of campus WiFi channels. (**a**) Channel usage (outdoors). (**b**) Channel usage (indoors)

and reduce co-channel interference in the same network. In the 2.4 GHz band, public WiFi APs commonly select channels 1, 6, and 11 which are completely independent from each other and the other channels are not occupied to avoid the adjacent channel interferences within WiFi networks. However, in the 5 GHz band, the uniform deployment appears to be enabled without considering the adjacent channel interference. Considering the coverage and penetrability of 5 GHz WiFi networks, the proportion of 5 GHz WiFi APs is lower than the outdoor 2.4 GHz WiFi APs while the proportion of 5 GHz WiFi APs is higher than the indoor 2.4 GHz APs for providing higher access rate and better access quality.

Figure 2.9 shows RSSI CDFs of public campus WLAN under the indoor and outdoor environments. Figure 2.9a shows the RSSI CDF from every measurement record and results have obviously shown that the indoor signal strength is stronger than outdoors. Due to certain coverage of WiFi APs, a WiFi AP RSSI information can be measured frequently in the coverage areas. To make a deeper comparison, we refined the maximal RSSI from the observation set $OD$; results shown in Fig. 2.9b

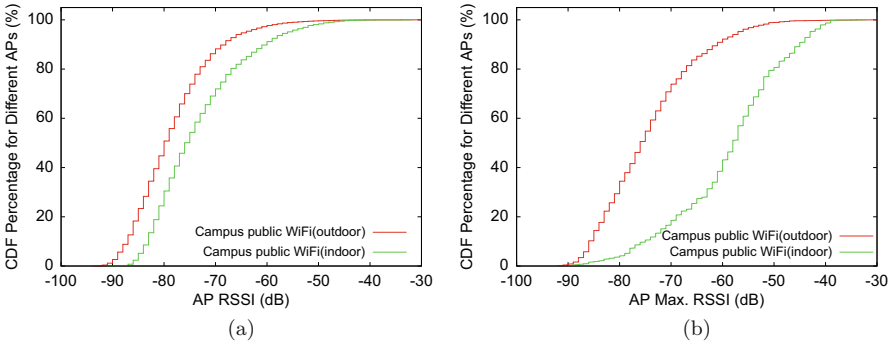



**Fig. 2.9** RSSI comparison of indoor and outdoor APs. (**a**) All RSSI. (**b**) Max. RSSI

also present that the indoor AP RSSI is much greater than outdoors. Hence, results prove that deployments of most WLAN APs are dependent on the buildings. Current mobile devices choose WiFi APs mainly relied on the current AP RSSI, so we can infer from the measurement that it is much easier to obtain access to available WiFi APs indoors rather than outdoors, and we also can infer that interferences indoors are more serious than outdoors.

#### 2.3.3.2 Indoor vs. Outdoor Interference of Public WiFi Networks

For more insights of WiFi networks, extra measurement experiments were also conducted indoors. Figure 2.10 presents the comparison of WiFi AP density results between the outdoors and the indoors. As shown in Fig. 2.10a, over 60% outdoor areas are covered with public WiFi APs ranged from 20 to 60, and nearly 20% areas are covered with over 60 ones. The adjacent channel interference among WiFi networks is shown in Fig. 2.10a by public WiFi network density with the green curve. As the previous analysis in Sect. 2.3.2.2, the default frequencies of most WiFi APs usually are configured on three independent channels (1, 6, and 11) in 2.4 GHz instead of other channels. Therefore, further public WiFi APs in the same areas would potentially generate additional co-channel interferences as shown in Fig. 2.10a with the red curve.

Figure 2.10b shows that 60% indoor areas are covered with public WiFi APs ranging from 40 to 100, which is doubled with the outdoor result. The extreme AP density at several indoor locations is over 200. Therefore, more serious co-channel interference has been discovered indoors through the comparison of Fig. 2.10. Figure 2.10 shows that public campus WiFi networks suffer the co-channel interferences from themselves rather than external private WiFi networks.

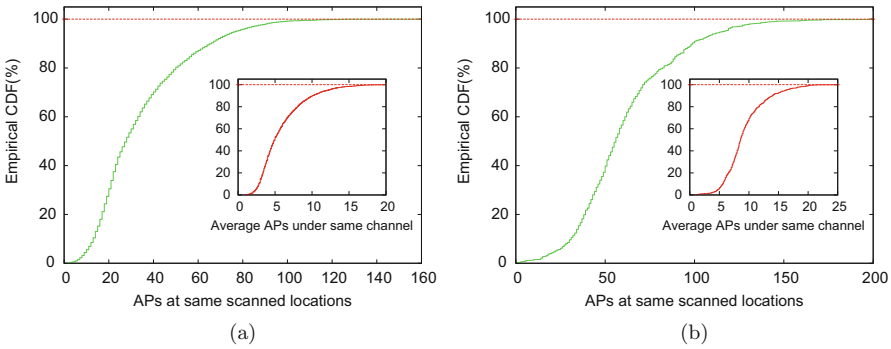



**Fig. 2.10** CDF of public campus WLAN density. (**a**) WiFi AP density CDF (outdoors). (**b**) WiFi AP density CDF (indoors)

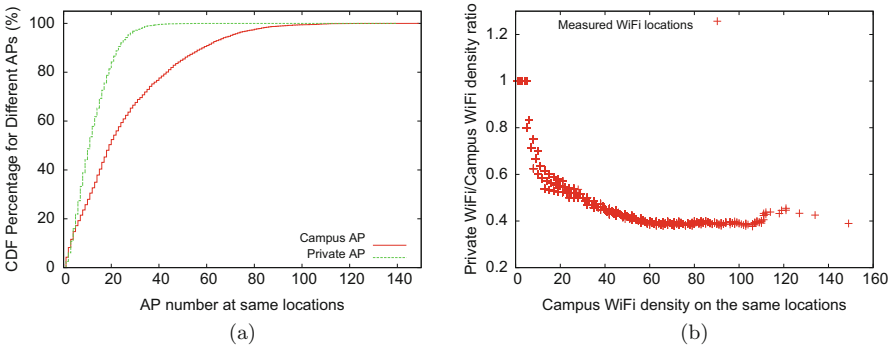



**Fig. 2.11** Public campus WLAN vs private WLAN. (**a**) CDF of private and public APs. (**b**) Density ratio of private/public APs

### 2.3.3.3 Interference of Hybrid WiFi Networks

Figure 2.11 presents a comparison of densities between public networks and private networks in measurement areas, where public and private networks construct hybrid WiFi networks. Figure 2.11a shows that public campus WiFi density is higher than private WiFi density on the average and 80% measurement areas are covered with less than 20 private APs. Compared between the private and public WiFi APs, densities of the public WiFi networks are almost doubled in the same locations. It can be inferred from results that public WiFi networks are not only affected from external private networks but also from the internal networks themselves.

To differentiate the network interference of hybrid network in various areas, we defined the **Relative Density Ratio** as private AP's density divided by public AP's density at a measurement location for further analysis. The defined ratios with public WiFi APs in the hybrid network areas is shown in Fig. 2.11b. Results indicate

that about 90% measurement areas obtain the ratio value smaller than 1, which implies that the density of public APs is higher than the number of private APs at the same locations. Results demonstrate that public campus WiFi networks suffer from the potential interferences not only from those private networks but also from themselves due to their high-density deployments.

#### 2.3.3.4 Dynamic Frequency Selection Detection

To reduce the impact of spectrum interference, WiFi APs may adopt the DFS feature which can dynamically adjust the WiFi transmitting frequency based on the channel utilization of the WiFi APs in the neighborhood to avoid the busy channels and select the appropriate working channel. As shown in Fig. 2.12, we tracked the number of channels utilized by the same AP in our dataset. The results show that WiFi APs enabling DFS account for only 20% over all the measured dataset and about 80% percentage of WiFi APs do not change channel numbers at all. There might be two reasons for this observation: one is that these devices may not support DFS; the other one is that many users enable the DFS without configuring the WiFi devices appropriately so that AP devices stay with the default factory settings.

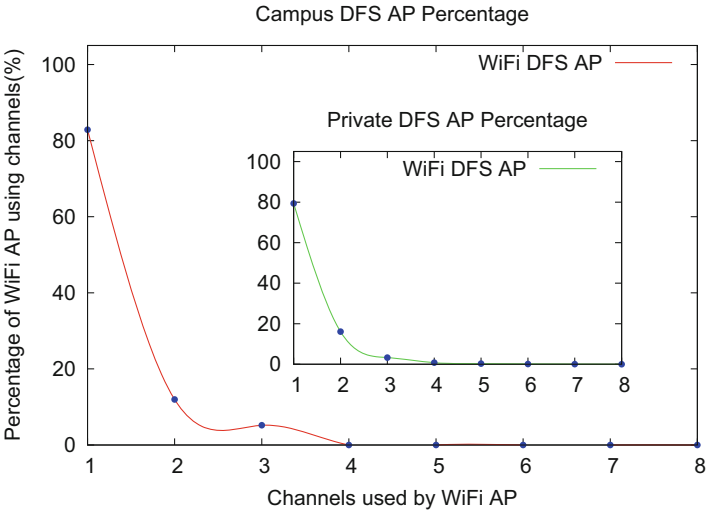



**Fig. 2.12** Dynamic frequency selection (DFS) technology used in WiFi APs

## 2.4 Characterization of WiFi Connection Time

During the WiFi sensing measurement process, the WiFi connection monitor module will cooperate with WiFiTracer to record the overall connection process when the public campus WiFi networks are available to use. Sensing results in Fig. 2.7 strongly recommend us that the connection measurement experiments should be conducted in the following distinct areas, such as 1–3 areas on the map shown in Fig. 2.7a, where public campus WiFi networks have been densely deployed.

### *2.4.1 WiFi Connection Dataset*

Based on results analyzed from the WiFi sensing dataset, the measurement areas can be determined for conducting the connection measurement experiments. Students invited as participants are required to load the connection monitor module in WiFiTracer and to move around in measurement areas during their daily lives. By over 2 months' measurement, a WiFi connection dataset has been successfully collected for various data of connection procedures, discussed previously in Sect. 2.2.

Table 2.4 presents a data summary of connection experiments conducted in chosen areas covered with well-deployed public campus WiFi networks. Over 70,000 times of connection attempts have been successfully observed from the measurement dataset, and only about 10% attempts have achieved the complete connection procedure and smoothly set up the data communication link between WiFi APs and clients. The dataset not only records the connection measurement results of public campus WiFi networks but also contains private WiFi networks' connection information for daily usages of participants.

From statistics in Table 2.4, the number of WiFi BSSIDs is much larger than WiFi SSIDs, which can be inferred that actual public WiFi networks extend the network coverage with multiple APs under the same SSID (recognizable network name) by WiFi ESS (extended service set) technique. Once successfully connected to a WiFi

**Table 2.4** Connection measurement dataset

| Metric | Amounts |
|---|---|
| Measure duration | Over 2 months |
| Phone models | 10 |
| Platforms | Android |
| Total connected WiFi SSIDs | 69 |
| Total connected WiFi BSSIDs | 2255 |
| Campus WiFi SSIDs | 3 |
| Campus WiFi BSSIDs | 1643 |
| Observed successful connections | 7289 |
| Observed connection attempts | 70,516 |

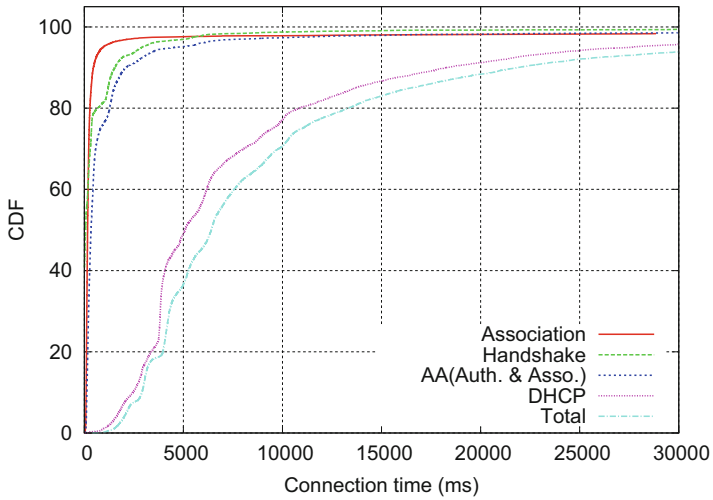



**Fig. 2.13** CDF of the total connection setup time

network, the basic network information, like SSID, username, and password, will be automatically recorded and stored locally at WiFi client side in Android system. Due to WiFi networks normally recognized by SSIDs, this feature will engage the WiFi client to trigger the connection procedure automatically once the previous connected networks with the same SSIDs becoming available, which is quite applicable for connection measurements.

## *2.4.2 Characterizing Successful WiFi Connections*

### 2.4.2.1 Overall of WiFi Connection Time

Figure 2.13 shows the CDFs of successful connection setup times for chosen WiFi networks, composed of AA times, handshake times, and IP acquisition (DHCP) times. Results demonstrate that the DHCP time is much larger than the other phases in the connection setup procedure, and occupies most of the connection time among all connection aspects. Furthermore, the AA time and handshake time only dominate a very small portion of the complete connection setup time, normally under 10%, and present a quite different trend from the connection time. About 80% successful connections can be completed within 10 s, which is considerably acceptable for mobile end users, and the main factor to influence the total connection time is the DHCP phase, which exhibits the similar variation tendency with the curve of connection setup time.

Figure 2.14 presents a close observation on the small portions of the connection setup procedure, which consists of the association phase, AA phase, and handshake

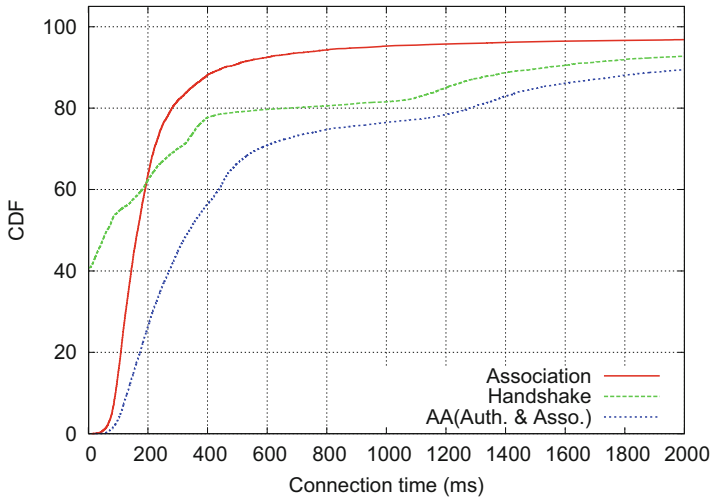



**Fig. 2.14** CDF of association, AA, and handshake times

phase, to demonstrate the detailed interactions during WiFi connections. Results reveal that these minor time phases are quite short, but vital for the connection setup, and some can be completed instantly without user's awareness.

Nearly 90% of association times are under 400 ms and about 80% of AA times are completed in 1000 ms. Note that the WiFi handshake mechanism, utilized by WiFi APs and clients to identify each other based on some security specifications (WPA or WPA2), is optional for the connection setup. If the WiFi network is open and free accessible for WiFi clients without any security and authentication, the handshake time will always be 0 ms. About 40% of handshake times remain 0 in the dataset, due to some WiFi networks configured as "[ESS]" for totally free access without any security mechanism. Currently, some public WiFi networks utilize extra authentication portals, such as web access control servers or popular Android tools like WeChat [24], to verify the clients' intentions of WiFi connection, which does not encrypt the wireless communication lines and totally does not need any handshake procedure. It seems that the omitted handshake phase would shorten the overall connection process; however, in real environments, this connection approach needs the manual operations of WiFi clients and substantially lengthens the connection time than the normal ones.

#### 2.4.2.2 Differentiate WiFi Connection Time by Various Devices

Figures 2.15 and 2.16 show the dissections of the connection time under the variant mobile devices. Figure 2.15 shows that the connection time range varies from several milliseconds to several seconds, and all the measured devices have a median

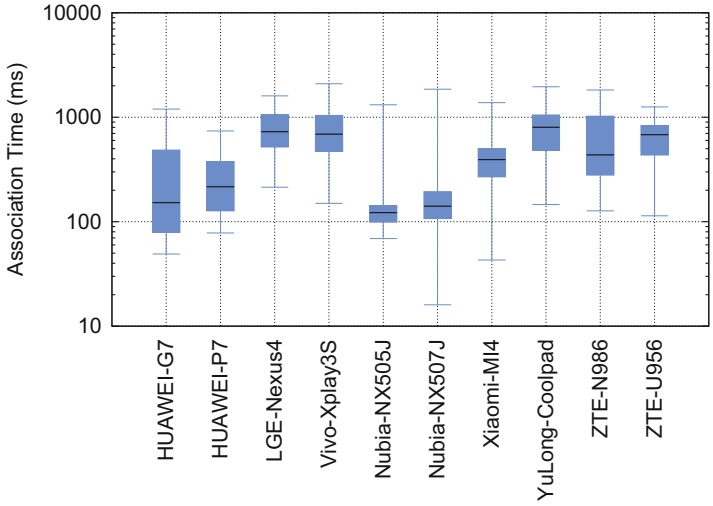



**Fig. 2.15** Minimum, 25th, 50th, 75th percentiles, and maximum of AA time

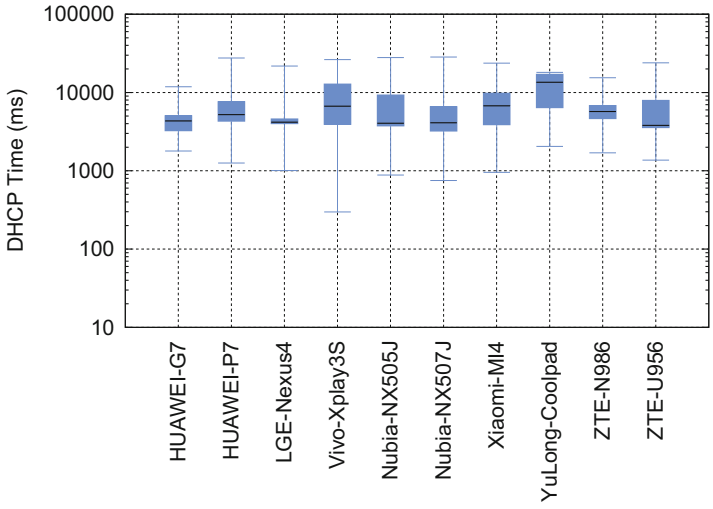



**Fig. 2.16** Minimum, 25th, 50th, 75th percentiles, and maximum of DHCP time

association time smaller than 1 s. 6 phone models have a 75th percentile of less than 500 ms and it is less than 2 s among all the measured phone models.

Figure 2.16 shows that 6 phone models have a median percentile of IP acquisition time ranging from 3 to 5 s, and for 2 phone models, the time is greater than 6 s. And, for one model it was greater than 10 s. Measurement results show that the IP acquisition time is mainly distributed in the range of [2, 10] s. From these two figures, the IP acquisition time is the dominated part and greatly affects the overall connection time.

## 2.5 Related Work

In this section, we review the measurement studies on WiFi networks. Spectrum interference of WiFi networks have been examined in passive sniffing in a few studies. In [25], Rose and Welsh designed the Argos system which deployed 26 stationary devices around a city to sense wireless devices using passive sniffing. Argos is the first city-level wireless network inspection system that can detect, measure, and analyze the performance of wireless devices and networks, including network type, data traffic, and application type, etc. However, due to the specialized devices and fixed deployment in Argos, it is quite expensive and has low flexibility in the large deployment and measurement. Applying a similar carrier sensing mechanism, in [26] Paul et al. studied the interference of WiFi networks for detecting misbehaving WiFi nodes. Van Bloem et al. studied the effect of different interference sources on WiFi network, such as audio and video transmitter, microwave, and Bluetooth [27]. The results showed that the audio and video transmission have a serious impact on WiFi networks and lead to poor network performance.

Active measurement has also been applied in WiFi measurement. Sommer and Barford utilized an Internet measurement web site to the clients and collected more than 300 million user measurement results in 15 different areas [2]. They compared various performance issues between WiFi networks and cellular networks in distinct areas. Their results show that there exists notable room for improving and optimizing the deployment for these two kinds of networks. In [28], Seneviratne et al. investigated the WiFi connection setup time in a lab environment. They examined the whole WiFi connection procedure between smart phones and WiFi APs. They found that the WiFi connection time is strongly impacted by the DHCP message transmission and proposed a scheme to accelerate the DHCP process for reducing the WiFi connection time. Farshad et al. utilized an Android App, RF Signal Tracker, to measure WiFi APs and networks in a typical European city (Edinburgh) in [29]. Participants took buses as the carriers on the main roads in the city and run the measurement tool in smartphones to characterize the urban WiFi distributions and features using the MCS. WiGLE [30] aggregated the WiFi measurement data collected by the war-driving measurement tool and constructed the wireless network mappings on the Google map which was also visualized on a web site. The results showed that WiFi networks have been hugely increased and densely deployed in recent years, and WiFi devices have been experiencing potential channel and spectrum interference in high-density deployment areas.

In this chapter, we designed and implemented a MCS platform to conduct a comprehensive WiFi measurement study with the focus on two aspects in both spectrum interference and connection establishment during mobility in a real campus network environment. We classified the interferences between the private and public WiFi network. In particular, we inspected the overall WiFi connection procedure by dissecting the detailed steps in the connection process with tracking connection state transitions. We also analyzed the reasons for unsuccessful connections based on our measurement data. Similar to [29], our results confirm that WiFi network deployments have been increasingly dense and causing consequences.

## 2.6 Conclusion

In this chapter, we conducted a measurement study of increasingly densely deployed WiFi networks in a campus area based on the MCS mechanism. Due to no planning, large-scale, and high-density deployment of WiFi networks, our measurement results show that current WiFi networks have various problems in frequency interferences. The campus WiFi networks suffer from the potential interferences not only from private networks but also from themselves for the high-density deployment in the main channels. With the growth of the public WiFi networks' deployment density, the intra-network interference is becoming dominating, and the aggregate interference becomes more severe. By supporting the measurement tools using the MCS way on Android systems, we chose measurement areas with well-deployed public campus WiFi networks to investigate the characteristics of the connection setup time of WiFi networks. The WiFi connection setup is the prerequisite condition for the WiFi connection and data transmission. Our measurement results showed that the connection time deviates significantly on different mobile devices.

Our overall measurement results showed that the current WiFi network deployments are largely unplanned and disordered and lead to the significant performance degradation due to inter-/intra-interference, the competition, and sharing of channels. It may not be sufficient to solve these emerging problems in densely deployed WiFi networks with the standard 802.11 protocols. Motivated by the findings in this chapter, we will design and develop software-defined WiFi network infrastructure and protocols to mitigate the interference and mobility to enhance the performance and manageability of WiFi networks [31–33]. Our preliminary study have demonstrated the feasibility of constructing a software-defined WiFi network testbed and there exists the trade-off between performance and programmability of software-defined APs [34].

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China (no. 61370231), in part by the Fundamental Research Funds for the Central Universities (nos. 2016YXMS303 and 2017KFYXJJ190).

## References

1. Zhang C, Qiu D, Mao S, Hei X, Cheng W (2015) Characterizing interference in a campus WiFi network via mobile crowd sensing. In: 11th international conference on collaborative computing (CollaborateCom), pp 173–182
2. Sommers J, Barford P (2012) Cell vs. WiFi: on the performance of metro area mobile connections. In: ACM SIGCOMM IMC, pp 301–314
3. Suiy K, Zhou M, Liu D, Ma M, Pei D, Zhao Y, Li Z, Moscibroda T (2016) Characterizing and improving WiFi latency in large-scale operational networks. In: 14th annual international conference on mobile systems, applications, and services (MobiSys), pp 347–360

4. Shi J, Meng L, Striegel A, Qiao C, Koutsonikolas D, Challen G (2016) A walk on the client side: monitoring enterprise WiFi networks using smartphone channel scans. In: IEEE INFOCOM
5. Gao Y, Dai L, Hei X (2017) Throughput optimization of multi-BSS IEEE 802.11 networks with universal frequency reuse. IEEE Trans Commun 65(8):3399–3414
6. Goel U, Wittie M, Claffy K, Le A (2016) Survey of end-to-end mobile network measurement testbeds, tools, and services. IEEE Commun Surv Tutorials 18(1):105–123
7. Guo B, Wang Z, Yu Z, Wang Y, Yen N, Huang R, Zhou X (2015) Mobile crowd sensing and computing: the review of an emerging human-powered sensing paradigm. ACM Comput Surv 48(1):7:1–7:31
8. Khan WZ, Xiang Y, Aalsalem MY, Arshad Q (2013) Mobile phone sensing systems: a survey. IEEE Commun Surv Tutorials 15(1):402–427
9. Lane N, Miluzzo E, Lu H, Peebles D, Choudhury T, Campbell A (2010) A survey of mobile phone sensing. IEEE Commun Mag 48(9):140–150
10. Zhuang Y, Syed Z, Georgy J, El-Sheimy N (2015) Autonomous smartphone-based WiFi positioning system by using access points localization and crowdsourcing. Pervasive Mob Comput 18:118–136
11. Gao C, Kong F, Tan J (2009) HealthAware: tackling obesity with health aware smart phone systems. In: IEEE ROBIO, pp 1549–1554
12. Vu L, Nguyen P, Nahrstedt K, Richerzhagen B (2015) Characterizing and modeling people movement from mobile phone sensing traces. Pervasive Mob Comput 17:220–235
13. Mun M et al (2009) PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In: ACM MobiSys
14. Ganti R, Ye F, Lei H (2011) Mobile crowdsensing: current state and future challenges. IEEE Commun Mag 49(11):32–39
15. Lane ND, Eisenman SB, Musolesi M, Miluzzo E, Campbell AT (2008) Urban sensing systems: opportunistic or participatory? In: ACM HotMobile
16. Lane ND et al (2013) Piggyback crowdsensing (PCS): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In: ACM conference on embedded networked sensor systems (SenSys)
17. Xiao Y, Simoens P, Pillai P, Ha K, Satyanarayanan, M (2013) Lowering the barriers to large-scale mobile crowdsensing. In: ACM HotMobile
18. Shu P, Liu F, Jin H, Chen M, Wen F, Qu Y (2013) eTime: energy-efficient transmission between cloud and mobile devices. In: IEEE INFOCOM
19. Zhang T et al (2015) eTrain: making wasted energy useful by utilizing heartbeats for mobile data transmissions. In: IEEE ICDCS
20. Liu F et al (2013) Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. IEEE Wirel Commun 20(3):14–22
21. Liu F, Shu P, Lui JC (2015) AppATP: an energy conserving adaptive mobile-cloud transmission protocol. IEEE Trans Comput 64(11):1
22. Gualda D, Urena J, Garcia J, Garcia E, Ruiz D (2013) RSSI distance estimation based on genetic programming. In: International conference on indoor positioning and indoor navigation (IPIN), pp 1–8
23. IEEE standards, WiFi MAC producer query. http://standards.ieee.org/develop/regauth/oui/public.html/
24. Tencent Inc. Wechat: connecting people with chat, calls and more. http://www.wechat.com/
25. Rose I, Welsh M (2010) Mapping the urban wireless landscape with Argos. In: ACM SenSys, pp 323–336
26. Paul U, Kashyap A, Maheshwari R, Das S (2013) Passive measurement of interference in WiFi networks with application in misbehavior detection. IEEE Trans Mobile Comput 12(3):434–446
27. van Bloem JW, Schiphorst R, Kluwer T, Slump C (2012) Interference measurements in IEEE 802.11 communication links due to different types of interference sources. In: 2012 8th international conference on wireless communications, networking and mobile computing (WiCOM), pp 1–6

28. Seneviratne S, Seneviratne A, Mohapatra P, Tournoux PU (2013) Characterizing WiFi connection and its impact on mobile users: practical insights. In: ACM MOBICOM, pp 81–87
29. Farshad A, Marina M, Garcia F (2014) Urban WiFi characterization via mobile crowdsensing. In: IEEE network operations and management symposium (NOMS)
30. Wigle: Wireless geographic logging engine. http://wigle.net/
31. Nsunza WW, Hei X (2017) Design and implementation of a smart home router based on Intel Galileo gen 2. In: 12th EAI international conference on testbeds and research infrastructures for the development of networks & communities (TRIDENTCOM)
32. Chen Z, Fu D, Gao Y, Hei X (2017) Performance evaluation for WiFi DCF networks from theory to testbed. In: The 16th IEEE international conference on ubiquitous computing and communications (IUCC)
33. Kang J, Hei X, Song J (2017) A comparative study of Zynq-based OpenFlow switches in a software/hardware co-design. In: International workshop on network optimization and performance evaluation (NOPE)
34. Zahid T, Hei X, Cheng W, Ahmad A, Pasha M (2018) On the tradeoff between performance and programmability for software defined WiFi networks. Wirel Commu Mobile Comput 2018:12 pp

# Chapter 3
# People as Sensors: Towards a Human–Machine Cooperation Approach in Monitoring Landslides in the Three Gorges Reservoir Region, China


**Zhenhua Li, Guoxuan Cheng, Wenming Cheng, and Hongbo Mei**

**Abstract** Landslides are serious geologic hazards which have occurred in most countries and can cause significant loss of life and damage to property. The loss and damage may be avoided to some extent by monitoring and early warning systems for landslides. Currently, the most popular method to detect landslides is the wireless sensor network. In this paper, a human–machine cooperation system is proposed, which not only employs 500 sensor sets to collect data in the conventional way but also mobilizes over 6000 people to inspect landslides and gather data by simple tools daily, to take advantage of human wisdom and mobility to remedy the weakness of fixed sensors, which could not move, observe, think, and make decisions. For its 12 years of application in the Three Gorges Reservoir Region, China, the system has successfully predicted most threats which take place nearly 100 times each year.

## 3.1 Introduction

Landslides are common geologic hazards worldwide which heavily threat human life and property. On October 9, 1963, a huge landslide caused approximately 270 million cubic meters of rock and debris to fall into the Vajont reservoir, Italy, as a result a wave overtopped the dam and destroyed many villages in the Piave valley, with the loss of over 2000 lives.

For the detecting landslides, there are two types of observation method: remote sensing and ground-based monitoring. The former uses satellite, unmanned aerial vehicle (UAV), and even balloon to actively or passively monitor the surface

Z. Li (✉) · G. Cheng · H. Mei
China University of Geosciences, Wuhan, China
e-mail: zhli@cug.edu.cn

W. Cheng
The Headquarters of Prevention and Control for Geo-Hazard in the Three Gorges Reservoir Area, Yichang, China



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_3

displacement through optical, radar, and LiDAR technologies [2–4, 8]; the latter utilizes sensor networks to detect the whole status of landslide, on both surface and underground, such as surface deformation [1, 11], deep displacement [7], groundwater level [12], rainfall [10], reservoir level (if there is a reservoir around) [6], etc., or some or all of them [5, 9].

To take advantages of the above modern technologies, the government in the Three Gorges Reservoir area constructs a multilevel, three-dimensional, and full-time monitoring system to integrate Remote Sensing Monitoring System, Global Positioning Satellite Monitoring System, and Comprehensive Monitoring System, including over 500 sensor sets to collect data of absolute displacement, relative displacement, groundwater level, pore water pressure, soil pressure, meteorology, earthquake, earth sound, and human activities, and, in addition, mobilizes over 6000 persons to collect data by simple tools and inspect landslides daily, in an economic way, to take advantage of human wisdom and mobility to remedy the weakness of fixed sensors, which could not move, observe, think, and make decisions.

## 3.2 Project Description

The Three Gorges Reservoir region has been vulnerable to landslides, because of its complicated terrain and climate. More than 70 geological disasters had taken place in this area from 1982 to 2000, killing over 400 persons. Landslides have seriously damaged the social and economic life in the reservoir area (Fig. 3.1).

However, the Three Gorges Dam Project, the largest hydroelectric dam in the world, though it brings huge benefits such as flood control, power generation, and navigation, etc., exacerbates the scale and frequency of the geological disasters, as the water is pushed to its maximum level periodically. On July 13, 2003, just 1 month after it first impounded up to the 135 m, Qianjiangping landslide occurred in Zigui County, which was responsible for the deaths of 24 people, and destruction of 129 houses and 4 factories. To prevent and respond to risks of the geological disasters, the Chinese government has so far invested over 250 million yuan in monitoring nearly 4000 landslides in this area.

## 3.3 The Sensor-Based Monitor System

The Monitor System is composed of three subsystems: the Remote Sensing Monitoring System, the Global Positioning Satellite (GPS) Monitoring System, and the Comprehensive Monitoring System, to build a multilevel, three-dimensional, and full-time monitoring and early warning platform.

**Fig. 3.1** Damages caused by landslides in the Three Gorges area

### *3.3.1 The Remote Sensing Monitoring System*

As the Three Gorges Dam was built, the storage water level went up from 135 to 175 m. To explore the environment influence of reservoir impoundments, remote sensing data in three stages: before filling, filling to 135 m level, and filling to 175 m level are compared to investigate the changes. And, the remote sensing maps, along with geographic information system and GPS system, became the foundation of the Monitor System (Fig. 3.2).

### *3.3.2 The GPS System*

The GPS Monitoring System adopts a three-layer hierarchical network architecture (see Fig. 3.3). The first layer, A-level, is the GPS control network with 15 control points; the second one, B-level, is the GPS base network with 210 base points; and the last one, C-level, is the GPS monitoring networks with 1070 monitoring points on 127 landslides [13].

**Fig. 3.2** Identify a landslide from remote sensing images

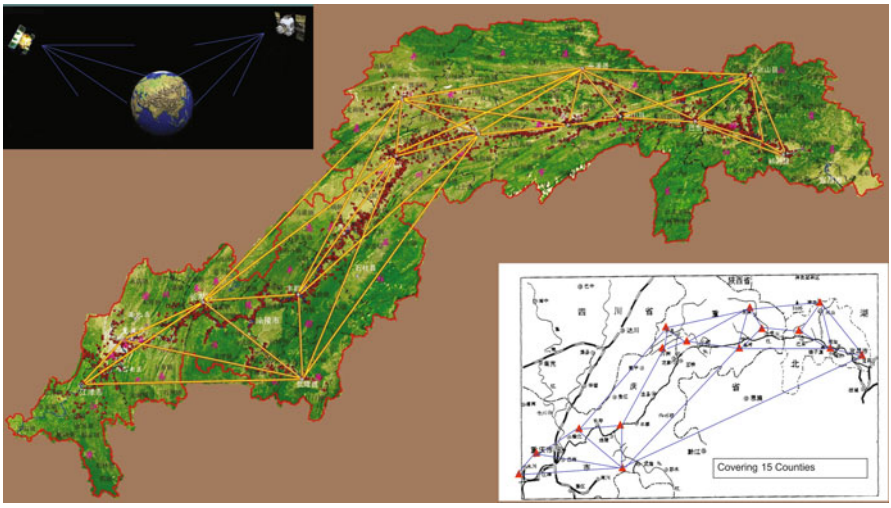



**Fig. 3.3** The GPS monitoring network

GPS monitoring points were built as can be seen in Fig. 3.4. For surface displacement monitoring, GPS technology yields 3D measurements with a high accuracy (in the range of a few millimeters) and high frequency (hours to days).

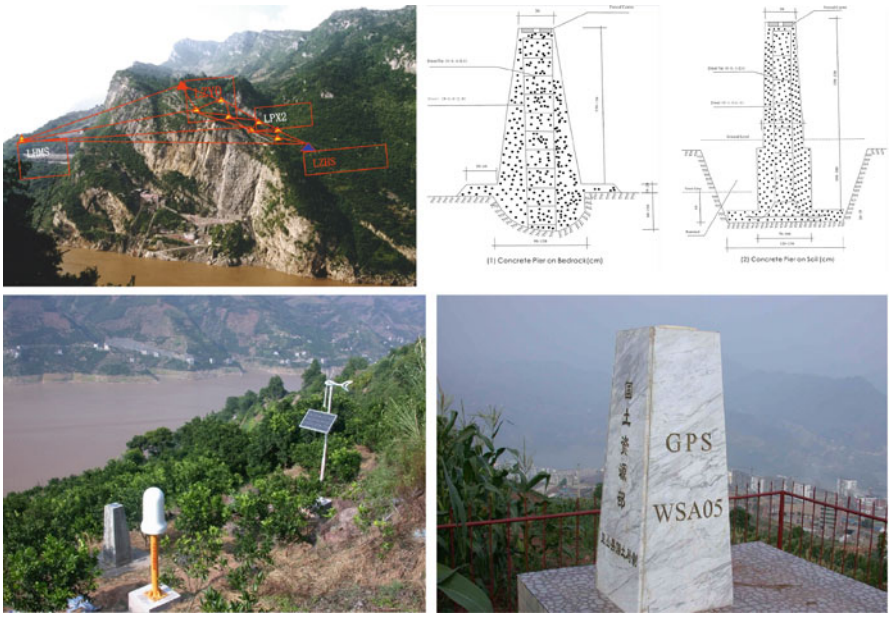



**Fig. 3.4** The GPS monitoring points

### *3.3.3 The Comprehensive Monitoring System*

The above two systems aim to detect surface displacement, the most important measurement. However, there are other factors also significant to landslide monitoring. According to geological conditions in study area, the Comprehensive Monitoring System is developed to monitor the rain gauge, deep deformation (see Fig. 3.5), landslide thrust, and groundwater level (see Fig. 3.6). The first measurement, rain fall, is an important factor to trigger landslide movement, and the latter three are meaningful kinematic parameters of landslide process.

There is still another trigger of geological disaster in this area, the reservoir water level, whose data comes from Changjiang Water Resources Committee.

## 3.4 The Human-Based Monitoring System: People as Sensors

The project employs over 6000 part-time local people to observe landslides and upload results daily through mobile phone. To use people as sensors, there are two obvious advantages in the Mass Monitoring System (Figs. 3.7 and 3.8):


**Fig. 3.5** Detecting rain gauge and deep deformation

**Fig. 3.6** Detecting landslide thrust and groundwater level


**Fig. 3.7** Information bulletin of a landslide observer (the upper figures), mobile APP for reporting observations (the lower-left figure), and warning devices, a gong & a loudspeaker (the lower-right figure)

1. Machines are programmed to follow certain rules, while wise people can adapt to new environments. In fact, there are no sensor networks that can replace people to investigate landslides so far.
2. Besides the investigation of landslides, some measurements can be done more flexibly and cheaply by people. In Fig. 3.9, a scrap of paper, or a tape can be used to detect the movement of landslides.

## 3.5 The Monitoring and Early Warning Platform

To combine the Sensor-based Monitor System and the Human-based Monitoring System and improve the efficiency of disaster response and preparedness, the Monitoring and Early Warning Platform is built as follows (see Fig. 3.10).

The Platform is composed of 4 subsystems and 27 modules, as shown in Fig. 3.11. The subsystems are information system, early warning system, data center, and administration system.

**Fig. 3.8** Penetration at the trailing edge of the landslide (the upper-left figure), crack at the trailing edge of the landslide (the upper-right figure), the steep surface between the failed body and the terrain (the lower-left figure), and road deformation in the front edge of the landslide (the lower-right figure)

## 3.6 Conclusions

The monitoring and early warning system, combined the Sensor-based Monitor System with over 500 sensor sets and the Human-based Monitoring System with 6000 people to monitor nearly 4000 landslides in 26 counties, has been run in the Three Gorges Reservoir Region for nearly 12 years and it has successfully predicted most threats which occur nearly 100 times each year, with no life loss.

The human–machine monitoring system, essentially a combination of Internet of Things and Crowdsourcing, with Chinese characteristics, offers an economical and flexible alternative to monitoring and prediction of geo-hazards, especially for developing countries.

**Fig. 3.9** Scraps of paper to monitor wall crack (the upper figures) and a tape to measure the length of landslide rapture (the lower figures)


**Fig. 3.10** Diagram of the monitoring and early warning platform

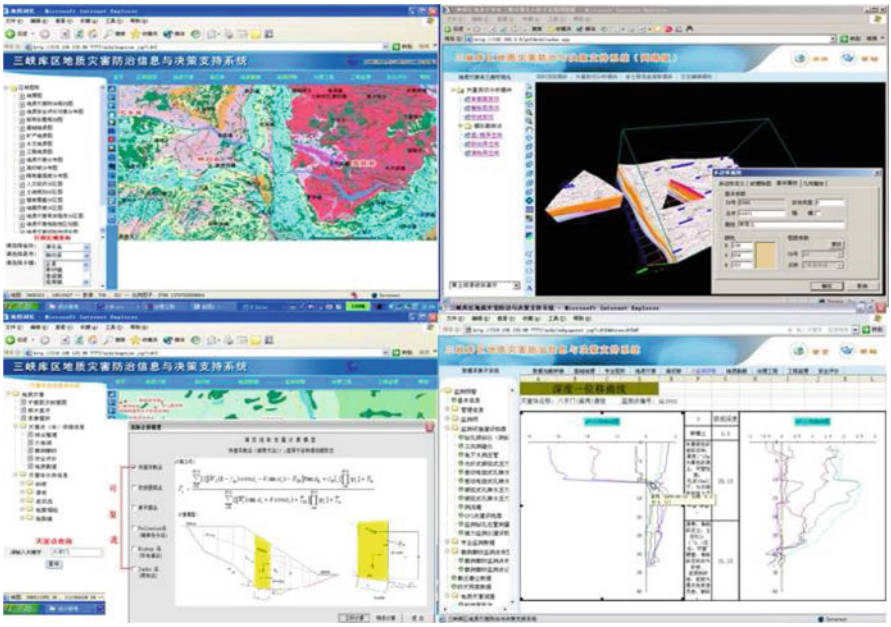


**Fig. 3.11** Snapshots of system UI

## References


1. Benoit L, Briole P, Martin O, Thom C, Malet JP, Ulrich P (2015) Monitoring landslide displacements with the geocube wireless network of low-cost gps. Eng Geol 195:111–121
2. Colesanti C, Wasowski J (2006) Investigating landslides with space-borne synthetic aperture radar (SAR) interferometry. Eng Geol 88(3–4):173–199
3. Dingfa H, Jun Q (1998) GPS-based target monitoring and navigation system for remote sensing-equipped flying balloon. In: Proc. SPIE 3504, Optical remote sensing for industry and environmental monitoring, pp. 131–135. https://doi.org/10.1117/12.319526
4. Jaboyedoff M, Oppikofer T, Abellan A, Derron MH, Loye A, Metzger R, Pedrazzini A (2012) Use of LIDAR in landslide investigations: a review. Nat Hazards 61(1):5–28
5. Ju Np, Huang J, Huang Rq, He Cy, Li Yr (2015) A real-time monitoring and early warning system for landslides in southwest china. J Mt Sci 12(5):1219–1228
6. Kaczmarek H, Mazaeva OA, Kozyreva EA, Babicheva VA, Tyszkowski S, Rybchenko AA, Brykala D, Bartczak A, Skowinski M (2016) Impact of large water level fluctuations on geomorphological processes and their interactions in the shore zone of a dam reservoir. J Great Lakes Res 42(5):926–941. https://doi.org/10.1016/j.jglr.2016.07.024
7. Lami Y, Nocera G, Genon-Catalot D, Lagreze A, Fourty N (2016) Landslide prevention using a buried sensor network. In: 2016 IEEE radio and antenna days of the Indian Ocean (Radio)
8. Maria Mateos R, Azanon JM, Roldan FJ, Notti D, Perez-Pena V, Galve JP, Luis Perez-Garcia J, Colomo CM, Gomez-Lopez JM, Montserrat O, Devantery N, Lamas-Fernandez F, Fernandez-Chacon F (2017) The combined use of PSInSAR and UAV photogrammetry techniques for the analysis of the kinematics of a coastal landslide affecting an urban area (SE Spain). Landslides 14(2):743–754

9. Palis E, Lebourg T, Tric E, Malet JP, Vidal M (2017) Long-term monitoring of a large deep-seated landslide (La Clapiere, South-East French Alps): initial study. Landslides 14(1):155–170. https://doi.org/10.1007/s10346-016-0705-7
10. Ramesh MV (2009) Real-time wireless sensor network for landslide detection. In: 2009 3rd international conference on sensor technologies and applications (Sensorcomm 2009) pp 405–409
11. Vera JE, Mora SF, Cervantes RA (2016) Design and testing of a network of sensors on land surfaces to prevent landslides. In: 2016 IEEE biennial congress of Argentina (Argencon)
12. Xu Q, Liu H, Ran J, Li W, Sun X (2016) Field monitoring of groundwater responses to heavy rainfalls and the early warning of the Kualiangzi landslide in Sichuan Basin, Southwestern China. Landslides 13(6):1555–1570. https://doi.org/10.1007/s10346-016-0717-3
13. Yue W, Xuebin H, Shaoquan X, Wenming C, Yingbing L (2006) Application of GPS technique for geological hazard professional monitoring in TGR area. J Geomatics 31(5):16-17

# Chapter 4
# Two Major Applications in Vehicular Ad Hoc Networks


## Rear-End Collision Warning & Automatic Incidents Detection

**Binbin Zhou, Zhan Zhou, Gang Pan, Shijian Li, Hexin Lv, and Tiaojuan Ren**

**Abstract** Vehicular ad hoc networks (VANETs) have emerged in the past decades as a significant type of networks, which consists of vehicles with sensors to communicate. For its ad-hoc nature, VANETs have great potential in a large number of applications, within which rear-end collision warning and traffic automatic incidents detection are two major applications. Because of the large number of injury and consequent economic loss, rear-end traffic collision has become an important issue and attracted a large number of attentions. In the past decades, there have been lots of efforts paid on this field. Existing work usually employed mathematical approaches or machine-learning approaches. In this study, we develop a collaborative rear-end collision warning algorithm (CORECWA), which is able to estimate and assess traffic risk in a collaborative and real-time way, and further notify drivers the warning message timely. Experiments results have shown that our algorithm outperforms the predominant method, HONDA algorithm. On the other hand, traffic incidents detection has been a critical problem in the past decades, due to the considerable economical cost and inestimable disgruntlement from numerous drivers. We present a support vector machines (SVM)-based approach for automatic incident detection (AID), in which the traffic data are collected by VANETs techniques. We process collected data and utilize traffic variables in the SVM model to confirm whether an incident occurs. Several experiments have been conducted to evaluate our approach's performance, and the results show that our approach could outperform the other two approaches in most cases.

B. Zhou (✉) · G. Pan · S. Li
College of Computer Science and Technology, Zhejiang University, Hangzhou, China
e-mail: bbzhou@zju.edu.cn

Z. Zhou
College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, China

H. Lv · T. Ren
College of Information and Science Technology, Zhejiang Shuren University, Hangzhou, China



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_4

## 4.1 Introduction

Vehicular Ad Hoc Networks (VANETs) have emerged in the past decades as a significant type of networks, which consists of vehicles with sensors to communicate. For its ad-hoc nature, VANETs have great potential in a large number of applications, within which rear-end collision warning and traffic automatic incident detection (AID) are two major applications.

It is well-acknowledged that the increasing traffic accidents can bring growing injury and consequent economic damage. So, it has become a severe social problem worldwide. Among them, rear-end traffic collision has attracted lots of attentions, due to the high-frequent occurrence (almost 30 %) [1, 2]. Therefore, it is urgent to develop rear-end collision warning system for message transferring and notification.

In the past, lots of efforts have been paid on this field [3–21]. All computation methods would take traffic data into consideration, either traffic data from probe vehicles or experimental data. In order to cover the aforementioned drawbacks, we propose a COllaborative REar-end Collision Warning Algorithm (CORECWA), to assess traffic risk in a real-time way and notify drivers the useful information timely. Compared with HONDA algorithm, our method is able to get better performance using a publically available dataset.

Meanwhile, traffic congestion has become a huge and increasingly severe problem worldwide nowadays, due to the growing demand on transportation and constraint resources supported by existing traffic infrastructures. Traffic incidents play a crucial role in the traffic congestion problem. In this way, incidents refer to abnormal events that occur to obstruct the normal satiny traffic flows and affect the utilization of traffic infrastructures, i.e., traffic accidents, interception because of hazard weather conditions. Hence, AID has been proposed and developed in the past decades, and attracted the interest of a number of scholars. Accurate and effective incidents detection could be helpful not only to relieve congestion, improve traffic efficiency, and decrease fuel cost but also to provide reliable information to drivers to reduce their travelling time. Usually, a large amount of traffic data utilized for AID has been a sharp problem. Data collection methods using current detectors (i.e., inductive loops and video cameras) have lots of shortcoming, e.g., the limited detection range and high costs of implementation and maintenance. Hence, we employ sensors nodes, which are widely used in VANETs, to detect, transmit, and fuse traffic data.

We employ a support vector machines (SVM)-based approach to detect the VANET-based incidents. And, we extract the most critical features related to incidents occurrence, such as speed, occupancy, and volume, and then train SVM through various feature combinations. Finally, we conduct experiments to evaluate the proposed approach's performance, and results present that our approach can outperform the relevant state-of-the-art approaches in three well-acknowledged evaluation metrics.

## 4.2 Rear-End Collision Warning

Existing work in rear-end collision warning can be reviewed from two groups: mathematical approaches and machine-learning approaches. There have been lots of mathematical methods for rear-end collision warning problems. Minimum Safety Distance-based methods and Minimum Safety Time-based methods are two well-acknowledged methods. They mainly consider the essential distance or time between two preceding and following cars as thresholds [3–8]. Based on the two methods, there have developed some related methods, including MAZDA [9, 10] and Honda [11]. Perception reaction time has been an important factor in the methods [12, 13]. Its value varies from 0.5 to 2.5 s [13, 14]. They also consider the minimum computed predictable time as a factor in the algorithm design [15, 16]. There are also several machine learning-based studies in this field. Researchers have focused on drivers' behavior to improve traffic collision warning methods [17, 18]. Neural networks have been used to adapt to warning message for drivers by learning behavior models of drivers, using genetic algorithms to optimize related parameters [17]. Wang et al. developed a driving-assistance system, to push collision warning notices. Recursive least square methods can be used to identify and transform drivers' behavior information into the model and also adjust parameters [18]. Furthermore, lots of studies using machine learning-based methods to achieve improvement of reaction time-involved traffic collision warning issues [16, 19–21]. Chang et al. developed a fuzzy-based method for traffic pre-crash reminder using quantum-tuned BPNN-fused heterogeneous data, which are identified and transmitted through vehicle-to-vehicle (V2V) communication [16]. Wei et al. proposed a multilayer perception NN-based approach for minimum safety distance computation using the probe vehicle data [19]. There are also some studies combining Fuzzy logic and MLPNN together to this problem, providing warning notices for multidirectional collision situations [20] and automobiles in highways [21], respectively.

Before processing for different goals for these existing methods, researches need the traffic data first. Generally, they use sensors for the detection of preceding vehicles or other traffic information. The information usually need a real-time transmission to achieve the real-time traffic risk assessment, by V2V communication and vehicle-to-infrastructure (V2I) communication.

To cover these drawbacks, we have proposed an algorithm named as CORECWA that can assess traffic risk in a real-time way and notify drivers the corresponding collision warning notices timely. CORECWA is utilizing some collaboratively collected real-time traffic data, such as position and speed of preceding and following cars, etc. Traffic risk can be evaluated in a real-time way, considering space headway and current speed of the preceding and following vehicles, and also drivers' behavior characteristics, such as perception–reaction time. The results depict that CORECWA could achieve better performance comparing with *HONDA* algorithm.

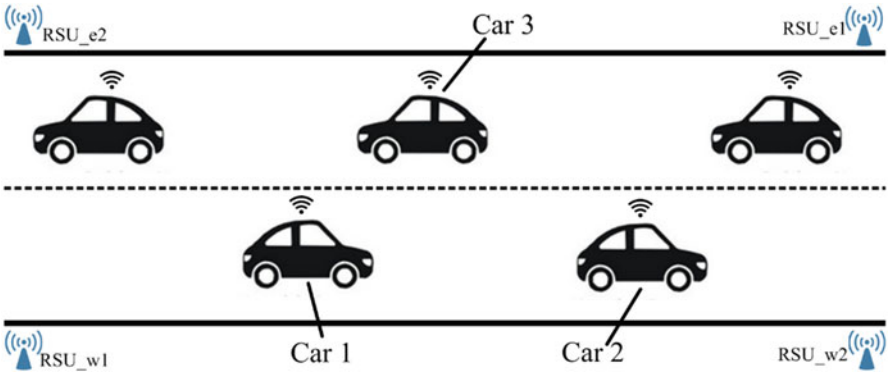



**Fig. 4.1** An example of traffic scenarios

### *4.2.1 Problem Description*

We formulate this rear-end collision warning problem as to notify drivers a warning message when in necessary situation. Therefore, these drivers are able to keep safe distance with appropriate actions.

To formulate this problem, we take an example of traffic scenarios in consideration (Fig. 4.1). From the figure, we can see that a road segment having four roadside units (RSU) has been illustrated, and all cars are able to monitor the surrounding environment and transfer information. All cars in this road segment would be detected by RSUs. For instance, when a car comes into this road from the west, this car would be monitored by RSU_w1 immediately, and then RSU_w2 can receive this information that there is a car with unique car id entering in this road segment.

Considering a rear-end collision warning problem, Car 1 is defined as a following car. We should confirm the corresponding preceding car, and then estimate current traffic risk. We define a threshold to discretize the traffic risk into three levels, under-risk, slight-danger, and emergent situation. Here, TR($V$) is defined as traffic risk of car $V$. The objective is to estimate TR($V$) and obtain a maximum threshold value Thresh($V$) to determine in which risk case drivers must take actions to keep safe, as presented in Eq. 4.1:

$$\text{Max} \quad \text{Thresh}(V) \tag{4.1}$$

There are many relevant factors for the parameter. It is obvious that the distance between preceding–following cars is important. Dst($V$, $\text{RSU}_{i1}$) and Dst($V$, $\text{RSU}_{i2}$) are defined as distance between car $V$ to $\text{RSU}_{i1}$ and $\text{RSU}_{i2}$ of the same direction and road segment, respectively, $i \in \{\text{east}, \text{west}\}$. vlc($V$) is defined as $V'$ speed, and rs($V$) is defined as the corresponding road segment which car $V$ belongs to.

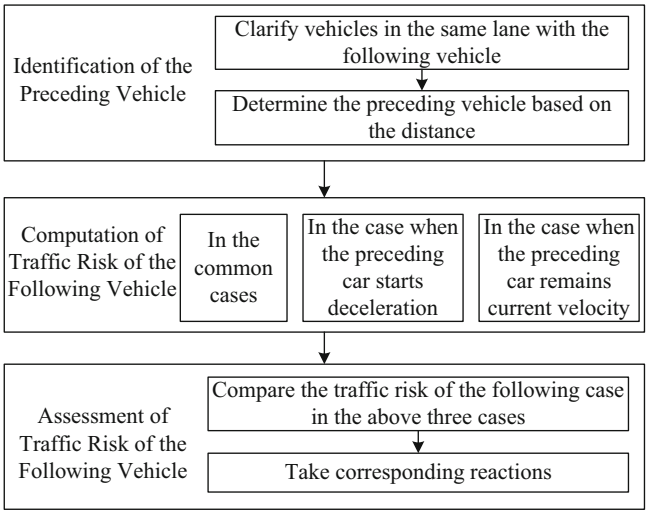



**Fig. 4.2** Process of collaborative rear-end collision warning algorithm (CORECWA)

### *4.2.2 Our Collaborative Real-Time Rear-End Collision Warning Algorithm*

In this section, we propose CORECWA as shown in Fig. 4.2. The algorithm contains three steps: the preceding vehicle identification, traffic risk computation of the following vehicle, and traffic risk assessment of the following vehicle. The first step is to confirm the preceding vehicle of the following vehicle. The second step is to compute traffic risk of following car, and meanwhile estimate the maximum and minimum thresholds of traffic risk. The last step is to assess traffic risk of following car.

#### 4.2.2.1 Identification of the Preceding Vehicle

In this step, the preceding vehicle of one particular following car would be determined. As shown in Fig. 4.1, it is obvious that preceding vehicle of Car 1 is Car 2, not Car 3 which has the shortest distance to Car 1. Therefore, preceding car identification cannot adopt the nearest distance method. The whole process of preceding car identification is shown in Fig. 4.3.

At first, two cars $V_P$ and $V_F$ would be in the same road covered by RSUs of the same roadside. When a car $V_F$ enters into the road from the west, it will send a message to the nearest RSU_w1. After that, RSU_w1 notifies RSU_w2 the event of a newly arriving car. And then, these two RSUs would store this vehicle's information, and broadcast message to cars in the same road to inform them the arrival of $V_F$. At that time, all vehicles in front of $V_F$ would communicate with $V_F$ and estimate distances between them. The distance computation would have some errors, because the location data would have some errors.

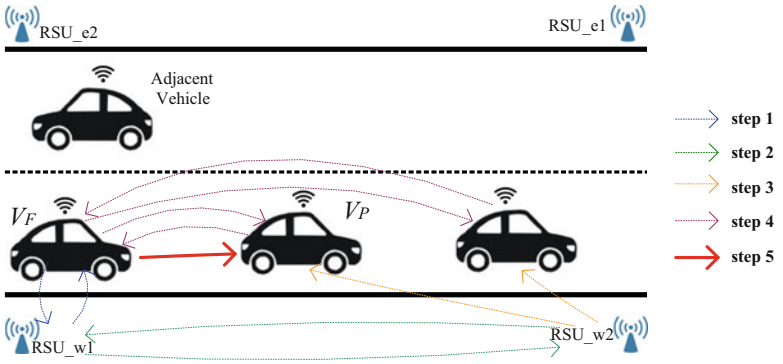



**Fig. 4.3** Process of preceding car identification

At last, $V_F$ will compare all distances between one car in front of $V_F$ and $V_F$, and choose the vehicle with the shortest distance as the preceding vehicle.

#### 4.2.2.2 Computation of Traffic Risk of the Following Vehicle

Then, traffic risk of $V_F$, which is defined as TR($V_F$), can be estimated with the basis of the minimum safety distance. As presented in Fig. 4.4a, there have been two cars $V_P$ and $V_F$ in the same road with speed vlc($V_P$) and vlc($V_F$), respectively, and Dst($V_F$,$V_P$) defined as the distance between two cars is what we want to know, and calculated in Eq. (4.2). After a certain time $T$, as shown in Fig. 4.4b, $V_F$ catches $V_P$ after a running distance of $s(V_F)$ and $s(V_P)$, respectively. With the consideration of perception reaction time, 1.5 s is employed in our method due to the previous work suggesting that 60 % rear-end collision can be eliminated with 0.5 s earlier warning and 90 % rear-end collision can be prevented with 1.5 s earlier warning [22]. Here, $a(V_F)$ and $a(V_P)$ are defined as the acceleration rate. sh($V_F$,$V_P$) is defined as the space headway of $V_F$ and $V_P$.

$$\begin{aligned} \mathrm{Dst}\left(V_\mathrm{F}, V_\mathrm{P}\right) \geq s\left(V_\mathrm{F}\right) - s\left(V_\mathrm{P}\right) &= \left(\mathrm{vl\,c}\left(V_\mathrm{F}\right) - \mathrm{vl\,c}\left(V_\mathrm{P}\right)\right) \times \left(T + 1.5\right) \\ &+ \frac{1}{2}\left(a\left(V_\mathrm{F}\right) - a\left(V_\mathrm{P}\right)\right) \times \left(T + 1.5\right)^2 \end{aligned} \quad (4.2)$$

We also consider two typical traffic scenarios. From that moment on, $V_P$ starts to decelerate until it stops running with a distance of $s(V_P)$ after time $T$. Once $V_P$ stops, $V_F$ just catches $V_P$ with the same velocity vlc($V_F$) and a running distance of $s(V_F)$. In this scenario, Dst($V_F$,$V_P$) would be calculated in Eq. (4.4), and we define this Dst($V_F$,$V_P$) as Thresh_min($V_F$) which refers to the minimum value of Thresh($V_F$).

$$T = \frac{\mathrm{vl\,c}\left(V_\mathrm{P}\right)}{a\left(V_\mathrm{P}\right)} \quad (4.3)$$

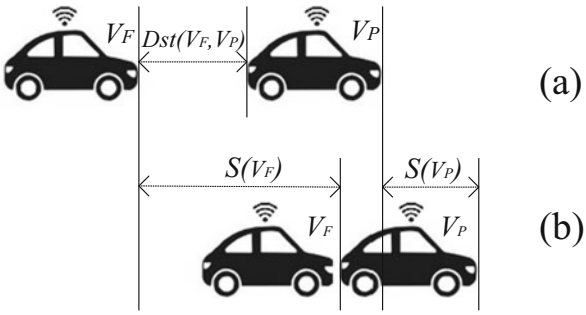



**Fig. 4.4** An example scenario of traffic collision

$$\text{Thresh\_min}\,(V_\text{F}) = (\text{vl c}\,(V_\text{F}) - \text{vl c}\,(V_\text{P})) \times \left(\frac{\text{vl c}\,(V_\text{P})}{a\,(V_\text{P})}\right) + 1.5 + \frac{1}{2}\,(a\,(V_\text{F}) - a\,(V_\text{P}))\left(\frac{\text{vl c}\,(V_\text{P})}{\text{a}\,(V_\text{P})} + 1.5\right)^2 \tag{4.4}$$

As in another typical traffic scenario, $V_\text{P}$ keeps running in the same velocity of vlc($V_\text{P}$) with a running distance of $s(V_\text{P})$, and $V_\text{F}$ just catches $V_\text{P}$ with the same velocity vlc($V_\text{F}$) and a running distance of $s(V_\text{F})$. Under this case, the Dst($V_\text{F}$,$V_\text{P}$) would be calculated in Eq. (4.5), and we define this Dst($V_\text{F}$,$V_\text{P}$) as Thresh_max($V_\text{F}$) which refers to the maximum value of Thresh($V_\text{F}$).

$$\text{Thresh\_max}\,(V_\text{F}) = (\text{vl c}\,(V_\text{F}) - \text{vl c}\,(V_\text{P})) \times (T + 1.5) \tag{4.5}$$

#### 4.2.2.3 Assessment of Traffic Risk of the Following Vehicle

After the computation of Dst($V_\text{F}$,$V_\text{P}$)$T$, Thresh_min($V_\text{F}$), and Thresh_max($V_\text{F}$), we would compare their value, and take actions as show in Eq. (4.6). If Dst($V_\text{F}$,$V_\text{P}$) is larger than Thresh_max($V_\text{F}$), which means that it is a safe traffic situation and there is enough long distance between $V_\text{P}$ and $V_\text{F}$, there is not any warning message or suggestion message should be sent. If Dst($V_\text{F}$,$V_\text{P}$) is larger than Thresh_min($V_\text{F}$) and smaller than Thresh_max($V_\text{F}$), which means that it is in somehow a traffic risk situation with a certain safe degree although. Hence, a warning suggestion message should be put forward. If Dst($V_\text{F}$,$V_\text{P}$) is equal to Thresh_min($V_\text{F}$), that refers to an urgent situation occurs. In this circumstance, we should send a collision warning message timely. Because these three steps run cyclical, the case that Dst($V_\text{F}$,$V_\text{P}$) greater than Thresh_min($V_\text{F}$) would not occur.

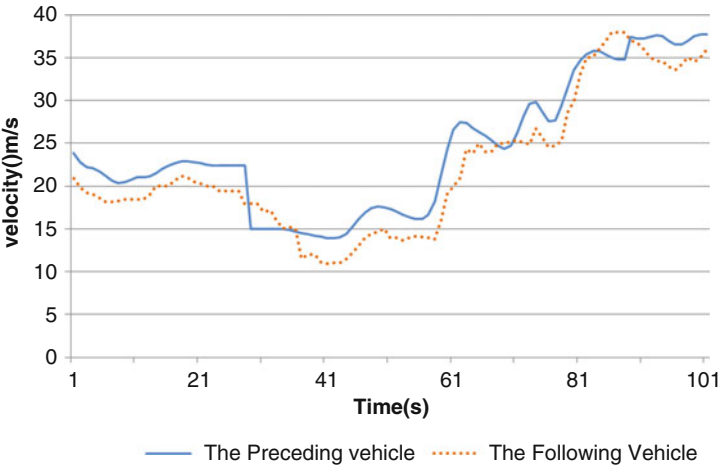



**Fig. 4.5** Velocity trend of two vehicles

$$\text{Dst}\ (V_{\text{F}}, V_{\text{P}}) \begin{cases} = \text{Thresh\_min}\,(V_{\text{F}}), \text{put forward a warning message and take brake;} \\ \geq \text{Thresh\_min}\,(V_{\text{F}})\ \text{and} < \text{Thresh\_max}\,(V_{\text{F}}), \text{put forward a suggestion on message;} \\ > \text{Thresh\_max}\,(V_{\text{F}}), \text{nothing to do;} \end{cases} \tag{4.6}$$

### *4.2.3 Evaluation*

Since a number of traffic data are not easy to collect, we adopt a public trajectory dataset, known as Next Generation Simulation (NGSIM) trajectory data, for experiments evaluation. These trajectory data contain car's speed, acceleration, location information, and so forth. To evaluate our algorithm's performance, we choose a well-acknowledged and popular algorithm *HONDA*.

As shown in Fig. 4.5, we compare the speed trends of preceding and following cars on the basis of trajectory data with sampling rate of 0.1 s. We also present the development trend of acceleration of the pair cars in Fig. 4.6. As presented in these two figures, we can see that several relationships exist between the pair cars in speed and acceleration. For instance, when the preceding vehicle has a sharp deceleration, the following vehicle needs to conduct responsive actions to keep safe.

To validate our method's effectiveness, we present the performance of HONDA method in Fig. 4.7 for comparison. From this figure, we can see that HONDA method is able to estimate traffic risk accurately in several critical durations, e.g., when the acceleration value or deceleration value is larger than 5 m/s$^2$. However,

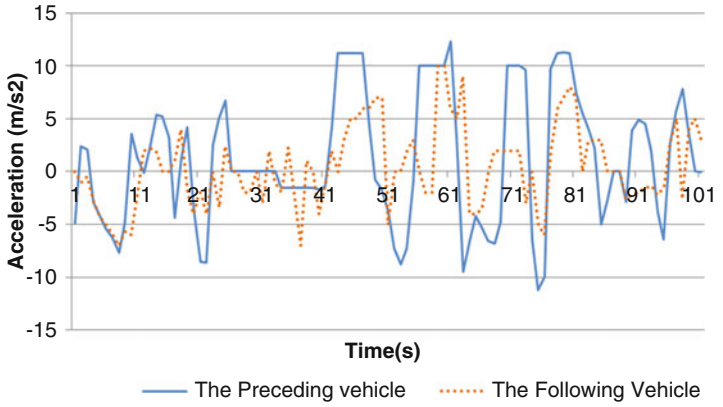



**Fig. 4.6** Acceleration trend of two vehicles

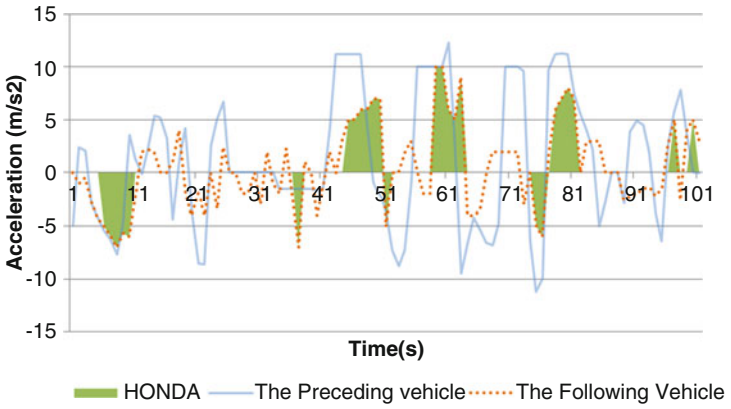



**Fig. 4.7** Performance of HONDA

there also have been some moments that HONDA method cannot detect the risk situation accurately, such that it cannot notify the drivers the necessary information timely, e.g., when the acceleration value or deceleration value is not high.

Our CORECWA algorithm has been presented in Fig. 4.8. It is obvious that our method is able to monitor most of the risk traffic occasions. After that, drivers are able to receive corresponding collision warning information. For instance, when the acceleration value or deceleration value of following car grows sharply, our method can recognize and estimate traffic risk of this following car, with warning decision generation.

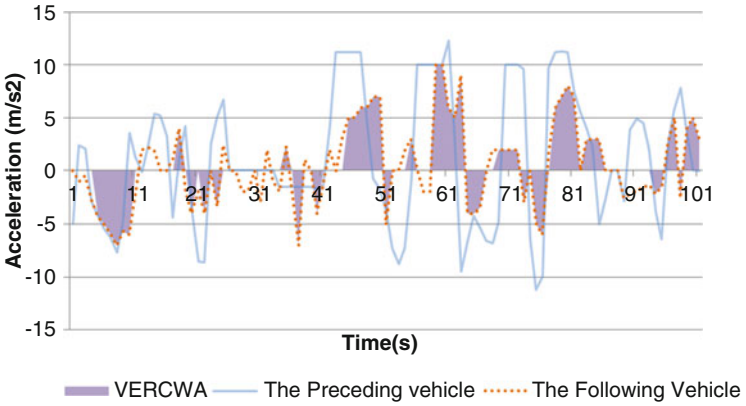



**Fig. 4.8** Performance of CORECWA algorithm

## 4.3 Automatic Incidents Detection

There has been a large amount of studies adopting various techniques for AID in recent years [23–37]. From the perspective of application fields, previous AID approaches are mainly applied in two fields, freeways and urban roads. These two application areas have different traffic characteristics. In freeways, traffic flow would present in a smooth and satiny way with various traffic density, which result in relatively homogenous traffic patterns [19]. On the contrary, traffic flow in urban zones are guided and controlled by traffic signals and traffic police, which would lead to a remarkable difference of traffic pattern compared with in freeways.

From the perspective of detection techniques, previous research generally can be categorized into four groups, machine learning (ML)-inspired algorithms, time series analysis (TSA), other comparative approaches, and hybrid approaches. ML-based methods focus on traffic patterns and estimate the current detected traffic variables whether it is incident-free [24–30]. TSA approaches underline dynamic and abnormal changes of traffic [31–34]. There are also some comparative approaches [35, 36] and hybrid approaches [37, 38].

We choose an SVM-based approach to detect the VANET-based automatic incidents. SVM can be used for data analysis and pattern recognition through its supervised learning models companied with associated learning algorithms [39, 40]. SVM are effective tools in a broad area of classification problems and robust to irrelevant features [41]. We also extract the most critical features related to incidents occurrence, such as speed, occupancy, and volume and then train an SVM. Experiments have been conducted to evaluate the proposed approach's performance, on a publicly available dataset containing real-world traffic data in California, which is used in a wide range of relevant studies. The simulation results present that our approach can outperform the relevant state-of-the-art approaches in three well-acknowledged evaluation metrics.

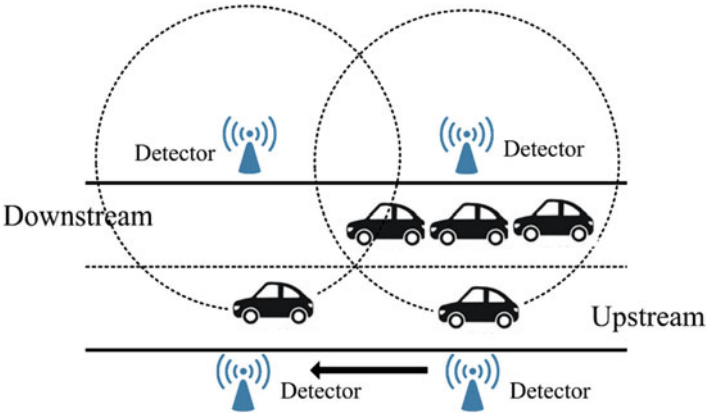



**Fig. 4.9** A road scenario

### *4.3.1 Problem Formulation*

The problem of AID is how to detect the considerable abnormal traffic situation from plentiful and dynamic changing traffic states, with only two results, incident occurred or incident-free. It is similar to the binary classification problem. Our objective is to find the red line to separate these green circles and blue triangles into different sides. In this way, when some traffic variables are detected real-timely and inducing a traffic situation deviation with regular traffic patterns, we can utilize the red line to confirm which side these traffic variables should take place in.

To model this problem, we would consider a detector-equipped freeway road scenario (see Fig. 4.9) which is divided into several segments due to detector's detection range. We assume that when each vehicle comes into a road segment, the corresponding detector can sense its existence successfully.

We represent traffic variables in different segments as vectors $x(i)$, $i = 1,2,3,\ldots,N$, defined as a road segment label. Each $x(i)$ has its own final result, defined as $y(i)$, $y(i)$ $\{-1,1\}$. Our objective is to find a function $F$ in the following expression:

$$F : \xi \rightarrow y \tag{4.7}$$

### *4.3.2 Our Automatic Incidents Detection Approach*

In this section, we propose our AID approach based on the above-established model. The work flow of our approach has been depicted in Fig. 4.10. There are four steps: data collection, data preprocess, data utilization in SVM model, and situation determination.

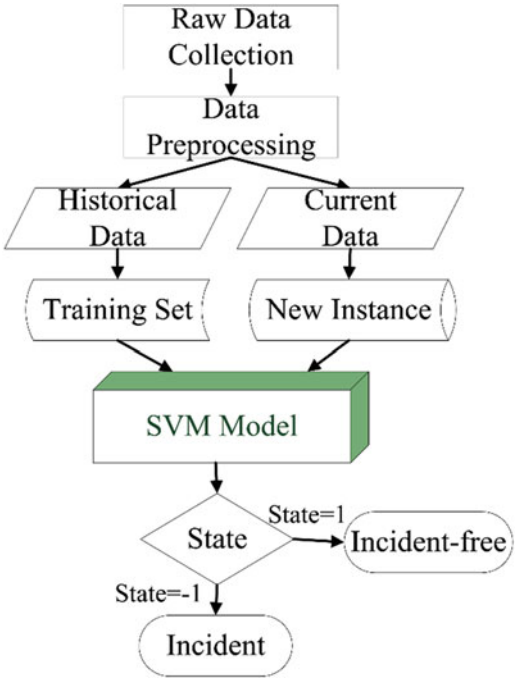



**Fig. 4.10** Work flow of our support vector machines (SVM)-based automatic incident detection (AID) approach

When detecting traffic data, sensors equipped on roadside are usually the popular choices due to their convenient deployment and maintenance, such as wireless sensors. After the real-time traffic data is collected, they need to be preprocessed in order to adapt to SVM model. In this model, when an incident happens in a segment, traffic volume of this segment and following segments would grow rapidly, with tangible reduction in the segments ahead. Similar change trends would occur on segment occupancy. In terms of average traffic flow speed, the speed of this segment and following segments would decrease obviously, with distinct improvement in the segments ahead. Hence, we decide to treat both traffic volume difference and speed difference between current segment and segment ahead as input variables for the SVM model, which means that the data preprocess part should finish this job when receiving all the traffic data collected. Moreover, we treat the historical data as training data, and the real-time detected data as a new instance. The detailed mechanism of the SVM model would be presented in the following. Based on the output of the SVM model, we can confirm whether an incident happens.

Based on the analysis mentioned above, vector $x(i)$ has two elements, traffic volume difference between segment $i$ and segment ahead $i + 1$, defined as tvd($i$, $i + 1$), and speed difference between segment $i$ and segment ahead $i + 1$, defined as sd($i$, $i + 1$).

$$X(i) = (\text{t vd}\,(i, i+1)\,, \text{sd}\,(i, i+1))^T, \quad i = 1, 2, 3, \ldots, N \tag{4.8}$$

The objective is find a maximum-margin hyperplane $\omega \cdot \xi + b = 0$, which divides the variables with $y(i)$ equal to 1 from those with its value equal to $-1$. This problem can be transferred to its corresponding dual problem. And, its purpose is to find the optimal $\alpha^* = \left(\alpha_1^*, \alpha_2^*, \ldots \alpha_N^*\right)^T$.

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \ \left\langle X_i, X_j \right\rangle - \sum_{i=1}^{N} \alpha_i \\ &\text{st.} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\ &0 \le \alpha_i \le C, i = 1, 2, 3, \ldots, N \end{aligned} \tag{4.9}$$

After optimal solution is obtained, and

$$\omega^* = \sum_{i=1}^{N} \alpha_i^* y_i \xi_l \tag{4.10}$$

$$b^* = y_i - \sum_{i=1}^{N} y_i \alpha_i^* \ \left\langle \xi_l, \xi_\varphi \right\rangle \tag{4.11}$$

Thus,

$$F\left(\xi\right) = \text{sign}\left(\omega^* \cdot \xi + b^*\right) \tag{4.12}$$

The well-known sequential minimal optimization (SMO) algorithm is employed for this problem with cost $O(n^{2.3})$ in training and cost $O(v)$ in testing, where n is defined as the number of data instances and $v$ is defined as the number of support vectors [42].

## *4.3.3 Experiments and Analysis*

### 4.3.3.1 Experiment Data Preparation and Evaluation Metrics

The traffic dataset used for experiments is derived from the publicly available I-880 database from the Freeway Patrol Service Project in California, USA [43, 44]. This dataset includes the traffic data we demand for, such as traffic volume and speed. And, they also include abundant incident events, almost 45 lane-blocking incidents [27].

The most common and widely acknowledged evaluation metrics for AID are detection rate (DR), false alarm rate (FAR), and mean time to detect (MTTD). DR is defined as the proportion of correctly found traffic incidents in all traffic incidents, presented in Eq. (4.13). FAR is defined as the proportion of false decisions in all incident-free cases, and presented in Eq. (4.14). MTTD is defined as the average value of each period cost from the moment a traffic incident happens to the moment the traffic incident detected, and presented in Eq. (4.15), where N is defined as the total incident number.

$$\mathrm{DR} = \frac{\#\text{of correctly detected incidents}}{\#\text{of all incidents}} \tag{4.13}$$

$$\mathrm{FAR} = \frac{\#\text{of false decisions}}{\#\text{of all incident-free cases}} \tag{4.14}$$

$$\mathrm{MTTD} = \sum_{i=1}^{N} \frac{T_{\mathrm{detection}}(i) - T_{\mathrm{incident}}(i)}{N} \tag{4.15}$$

#### 4.3.3.2 Experimental Design and Analysis

In automatic incident detection problems, we would prefer higher DR, lower FAR, and shorter MTTD, which leads to a multipurpose problem. The three goals are difficult to achieve optimal solution simultaneously. A higher DR may cause higher FAR and longer MTTD. Hence, we evaluate the performance separately, DR versus FAR and MTTD versus FAR, respectively. Since our approach is based on SVM and select different features in the training stage, we adopt two representative related works [4, 18] as comparative approaches.

Figure 4.11a presents the detection rate comparison between an SVM baseline approach [4], an SVM approach with speed variable [18], and our proposed approach. From the figure, we can observe that our approach can outperform the other two approaches in most cases. When the FAR is lower than 0.5 %, the SVM baseline approach presents the best performance. With the increasing incident number, all three methods have witnessed higher FAR, companied with higher DR. At that time, our approach obtains the best performance and expands the difference from the other two approaches.

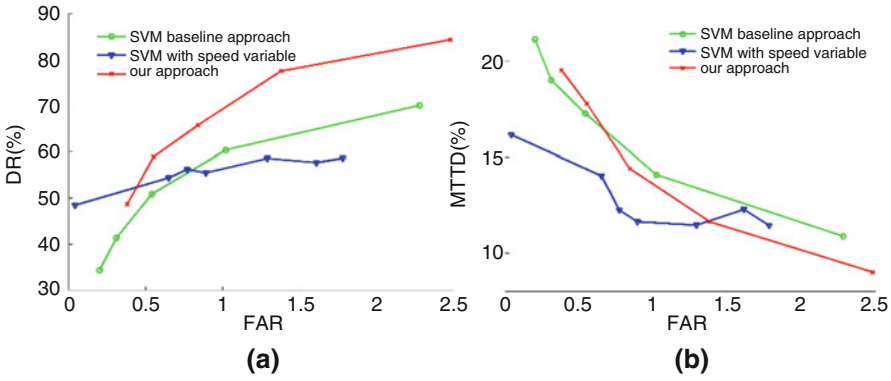



**Fig. 4.11** Performance comparison. (**a**) DR comparison, (**b**) MTTD comparison

Figure 4.11b presents the mean time-to-detect comparison between the three approaches. From the figure, we can notice that the three approaches have different performances when with different FAR. When the FAR is lower than 1.4 %, the approach from [18] achieves the best performance with much lower MTTD. When the FAR is higher than 1.5 %, our approach can outperform the other two approaches.

## 4.4 Conclusion

VANETs have been a popular network comprised of vehicles which can communicate with each other. This kind of network can provide substantial potential for vehicles' applications, to help vehicle be more safe, smart, and flexible. Rear-end traffic collision warning and traffic incidents automatic detection are two significant applications in VANETs.

For the purpose of rear-end collision relief, we proposed a collaborative method for rear-end collision warning problem, named as CORECWA. This method is able to provide drivers the useful and timely traffic risk information, with the utilization of surrounding traffic data detected and collected in a real-time way. The traffic data used here includes location information and speed information of cars. Using these data, our method is able to recognize the preceding car of one specific following car, and then estimate current traffic risk collaboratively, taking the following factors into considerations, such as velocity of two cars and behavior data of drivers (e.g., perception–reaction time). At last, we conduct experiments with the utilization of a public traffic dataset, to verify our proposed method's effectiveness. Experiment results show that our method can have better performance compared with a well-acknowledged method HONDA.

To detect traffic incidents automatically, we have presented an AID approach based on SVM with appropriate features, with traffic data detected by VANET techniques in a real-time manner. After several experiments conducted based on a real-world dataset, we confirm that our features selected can be beneficial for incidents detection, with higher detection rate and low mean time-to-detect with a certain level FAR, compared with two representative related works. In the future, we will optimize our work to further improve the detection rate, and we would pay efforts to optimize current approach in order to apply into urban areas.

**Acknowledgments** This work is supported in part by Public Welfare Technology Applied Research Program of Zhejiang Province, China (2014C33108, 2015C33028, 2015C33074, and 2015C33083), Research Fund of the Education Department of Zhejiang, China (Y201534553), and Zhejiang Provincial Natural Science Foundation of China (LZ15F020001, LY13F010013, LY15F030004, and LY14F020008).

## References


1. National Highway Traffic Safety Administration 2012 traffic safety facts. [Online] Available at: http://www-nrd.nhtsa.dot.gov/Pubs/812032.pdf/
2. Li L, Lu G, Wang Y, Tian D (2014) A rear-end collision avoidance system of connected vehicles. 2014 IEEE 17th international conference on intelligent transportation systems (ITSC), Oct 8–11, Qingdao, China
3. Chang T-H, Chou C-J (2009) Rear-end collision warning system on account of a rear-end monitoring camera. In: Intelligent vehicles symposium, 2009 IEEE, pp 913–917
4. Lu G et al (2013) A car-following model based on quantified homeostatic risk perception. Math Probl Eng 2013
5. Sharizli AA, Rahizar R, Karim MR, Saifizul AA (2014) New method for distance-based close following safety indicator. Traffic Inj Prev 16(1):190–195
6. Kusano KD, Gabler H (2011) Method for estimating time to collision at braking in real-world, lead vehicle stopped rear-end crashes for use in pre-crash system design, Paper published at SAE international
7. Oh C, Oh J, Min J (2009) Real-time detection of hazardous traffic events on freeways. Transp Res Rec J Transp Res Board 2129(1):35–44
8. Saccomanno F, Cunto F (2008) Comparing safety at signalized intersections and roundabouts using simulated rear-end conflicts. Transp Res Rec J Transp Res Board 2078(1):90–95
9. Seiler P, Song B, Hedrick JK (1998) Development of a collision avoidance system, Proceedings of 1998 SAE conference, no.98PC417
10. Doi A, Butsuen T, Niibe T, Yakagi T, Yamamoto Y, Seni H (1994) Development of a rear-end collision avoidance system with automatic braking control. JSAE Rev 15(4):335–340
11. Fujita Y, Akuzawa K, Sato M (1995) Radar brake system. Proceedings of the 1995 annual meeting of ITS America, 1, pp 95–101
12. Olson PL, Sivak M (1986) Perception-response time to unexpected roadway hazards. Hum Factors 28(1):91–96
13. Yi ZZ, Erik KA, Karl G (2006) A new threat assessment measure for collision avoidance systems. IEEE intelligent transportation systems conference, Toronto, Canada
14. Layton R, Dixon K (2012) Stopping sight distance, Kiewit Center for Infrastructure and Transportation, Oregon Department of Transportation
15. Yang H, Ozbay K, Bartin B (2010) Application of simulation-based traffic conflict analysis for highway safety evaluation. 12th WCTR, Lisbon, Portugal
16. Chang BR, Tsai HF, Young CP (2009) Intelligent data fusion system for predicting vehicle collision warning using vision/GPS sensing. Expert Syst Appl 37(1):2439–2450
17. Onken R, Feraric J (1997) Adaptation to the driver as part of a driver monitoring and warning system. Accid Anal Prev 29:507–513
18. Wang J et al (2013) An adaptive longitudinal driving assistance system based on driver characteristics. IEEE Trans Intell Transp Syst 14:1–12
19. Wei Z, Xiang S, Xuan D, Xu L (2011) "An adaptive vehicle rear-end collision warning algorithm based on neural network", ICCIC 2011, Part VI. CCIS 236:305–314
20. Nijhuis J, Neu Ber S, Spaanenburg L, Heller J, Sponnemann J (1992) Evaluation of fuzzy and neural vehicle control, Computer systems and software engineering, proceedings
21. Lemelson JH, Pedersen RD (1999) GPS vehicle collision avoidance warning and control system and method, Patent No. US 5983161 A
22. National Transportation Safety Board (2001) Special investigation report: highway vehicle- and infrastructure-based technology for the prevention of rear-end collisions. NTSB number SIR-01/01
23. Luk J, Han C, Chin D (2010) Freeway incident detection–technologies and techniques. Austroads, Sydney, p 72
24. Jin X, Cheu R, Srinivasan D (2002) Development and adaptation of constructive probabilistic neural network in freeway incident detection. Transp Res 10C(2):121–147

25. Srinivasan D, Sharma V, Toh KA (2008) Reduced multivariate polynomial-based neural network for automated traffic incident detection. Neural Netw 21(2/3):484–492
26. Yuan F, Cheu RL (2003) Incident detection using support vector machines. Transp Res Part C Emerg Technol 11(3/4):309–328
27. Chen S, Wang W, VanZuylen H (2009) Construct support vector machine ensemble to detect traffic incident. Expert Syst Appl 36(8):10976–10986
28. Zhang K, Taylor MAP (2006) Towards universal freeway incident detection algorithms. Transp Res Part C Emerg Technol 14(2):68–80
29. Zhang K, Taylor MAP (2006) Effective arterial road incident detection: a Bayesian network based algorithm. Trans Res Part C Emerg Technol 14(6):403–417
30. Xie K, Li X, Wang X, Xie G, Wen J, Cao J, Zhang D (2017) Fast tensor factorization for accurate internet anomaly detection. IEEE/ACM Trans Networking 25(6):3794–3807. https://doi.org/10.1109/TNET.2017.2761704
31. Tang SM, Gao HJ (2005) Traffic-incident detection-algorithm based on nonparametric regression. IEEE Trans Intell Transp Syst 6(1):38–42
32. Teng H, Qi Y (2003) Detection-delay-based freeway incident detection algorithms. Transp Res Part C Emerg Technol 11(3/4):265–287
33. Wei W, Shuyan C, Gaofeng Q (2007) Comparison between partial least square and support vector machine for freeway incident detection. In: 10th international IEEE conference on intelligent transportation systems. Doubletree Hotel, Seattle
34. Wang W, Chen S, Qu G (2008) Incident detection algorithm based on partial least squares regression. Transp Res Part C Emerg Technol 16:54–70
35. Payne HJ, Helfenbein ED, Knobel HC (1976) Development and testing of incident detection algorithms. Transp Res Rec 2., Report No. FHWA-RD-76-20, TRIS:316
36. Collins JF, Hopkins CM, Martin JA (1979) Automatic incident detection TRRL algorithms HIOCC and PATREG, TRRL Supplementary Report, No. 526, Crowthorne, Berkshire, UK
37. Lu L, Chen S, Wang W, Zuylen Z (2012) A hybrid model of partial least squares and neural network for traffic incident detection. Expert Syst Appl 39:4775–4784
38. Wang J, Li X, Liao S, Hua Z (2013) A hybrid approach for automatic incident detection. IEEE Trans Intell Transp Syst 14:1176–1185
39. Gakis E, Kehagias D, Tzovaras D (2014) Mining traffic data for road incidents detection. 2014 IEEE 17th international conference on intelligent transportation systems (ITSC), pp 930–935
40. Wen HM, Yang ZS, Jiang GY, Shao CF (2001) A new algorithm of incident detection on freeways. In: Proceedings of IEEE international conference on vehicular electronics, pp 197–202
41. Cheu RL, Ritchie SG (1995) Automated detection of lane-blocking freeway incidents using artificial neural networks. Transp Res Part C Emerg Technol 3(6):371–388
42. Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classif 10(3):61–74
43. Petty KF (1997) Incidents on the freeway: detection and management., Ph. D. dissertation. Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Berkeley
44. Al-Deek H, Fawaz Y, Noeimi H, Petty K, Rydzewski D, Sanwal K, Skabardonis A, Varaiya P. Online. Available: http://ipa.eecs.berkeley.edu/~pettyk/FSP/1995

# Chapter 5
# Concurrency and Synchronization in Structured Cyber Physical Systems


**Jitender Grover and Ram Murthy Garimella**

**Abstract** Cyber Physical System (CPS) involves the integration of the Cyber World and the Physical World. Structured Cyber-Physical Systems such as Wireless Sensor Networks (WSNs), Medical Monitoring, Smart Grids, Process Control Systems, Robotics Systems, Auto-Pilot Avionics, etc. are subjected to intense research and development. Efforts are underway to build various special purpose CPSs, without any disciplined effort to innovate theoretical principles required in their analysis and design. In the past few years, control theorists began to intensify research efforts to model and analyse special purpose, structured CPSs. When Concurrent Cyber Physical Systems evolve on multiple time scales, synchronizing such systems is an important problem. The literature on dynamical systems shows that the modeling of concurrent systems received limited interest. It is clear that if concurrent CPSs need to be analysed then taking Concurrency and Synchronization into account is essential. This chapter represents one interesting solution to this problem, i.e. Tensor State Space Representation (TSSR) for modeling and analysis of Concurrent Cyber Physical Systems. Furthermore, the synchronization and temporal semantics are absolutely essential for the control of CPSs evolving on Multiple Time Scales. Thus, the concepts for handling concurrency and time-scales are inevitable in the design and implementation of Cyber Physical Systems. This chapter basically presents an analytical approach to design, monitor, and maintain dynamical concurrent systems. Along with Concurrency and Synchronization, importance and management of Temporal Semantics are also presented in this chapter. Effective methods like Temporal Division of Labor and Edge Computing are proposed to deal with the issue of sensors' data fusion in multi-level servers. An Agent-based Framework is proposed for Reliable and Fault-tolerant CPS architecture. The type of modeling and analysis presented in the chapter will help in designing and implementing linear and distributed Cyber Physical Systems in a better way.

J. Grover (✉) · R. M. Garimella
Signal Processing and Communication Research Center (SPCRC), International Institute of Information Technology, Hyderabad, India
e-mail: jitender.grover@research.iiit.ac.in; rammurthy@iiit.ac.in



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_5

## 5.1 Introduction

Cyber Physical System (CPS) [1] involves the integration of the Cyber World and the Physical World. The evolution of Cyber Physical Systems began with the embedded systems [2]. Embedded systems are information processing systems that are embedded [3] into a larger product like machines or devices [4, 5]. Some examples of embedded systems can be found in the automotive sector when it comes to safety systems like anti-lock braking system (ABS) to regulate the brake force or the automatic four-wheel drive to increase traction or the airbag system that has a sensor that notices the crash and an actuator triggers the release of the airbag. Another example can be the traffic lights where pedestrians press a button to demand green light. So the term embedded system is used for a hardware and software system which is connected with the outside world through sensors and actuators to handle a distinct task [6, 7]. Embedded systems are followed by networked embedded systems. In the networked embedded system, a number of embedded systems get connected with each other and integrated into a wider context [8]. For example, a car, for instance, can be seen as an integrated control and information system where ABS, climate control, gearbox, speed control information and the motor control get connected to a system. Another example would be autonomous aviation. When the pilot switches on the autopilot, then the engine, the position and the speed of the plane need to be controlled. Also the actuators trigger adjustments to meet the settings of the autopilot [9, 10]. These early form of Cyber Physical Systems can be seen as networked embedded systems that are connected with each other through the internet. In this way, the physical world and the virtual world are merging as shown in Fig. 5.1. That's why they are called Cyber Physical Systems [11]. So, for example, cars in a city which can be seen as embedded systems share physical data like speed or distance with other cars through the internet, we could use these data to make road transport much more intelligent. It would be possible to improve traffic flow or to reduce accidents.

Cyber Physical Systems consist of physical components [12] and cyber components as shown in Fig. 5.1, that's why we call them cyber physical [13]. Actually, all Cyber Physical Systems are based on an information processing computer system which is embedded into a product like in a car, plane or other devices. These embedded systems do not stand alone anymore, they share their data via a communication network like the Internet with cloud computing [14]. This way the data from many embedded systems can be collected and processed in the cloud [15]. Connected embedded systems can be controlled decentralized via computational unit where data can be processed automatically or by the human–computer interface.

CPSs and IoTs have become the technology of future [16]. It is the base of Industry 4.0. The expansion of connected CPSs with us over the years is shown in Fig. 5.2. In 2005, when Mobile Era started, humans were connected to only/mostly three devices like Mobile, Computer, and Tablet through the Internet or other technologies.

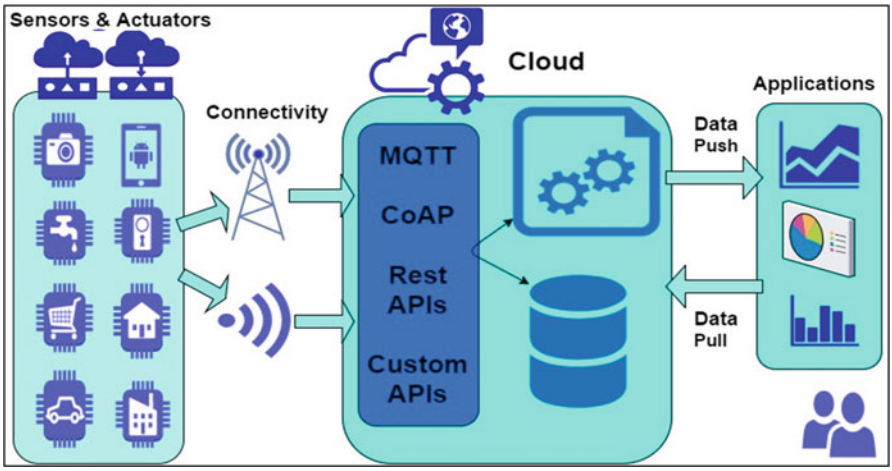



**Fig. 5.1** IoT-Cloud infrastructure

In the IoT-Era, started in 2015, we own as many as 10–15 electronic devices ranging from wearables to non-wearables. Now if we go towards future, there can be more than 100 connected and concurrent Cyber Physical Systems and all over the world 100 Billion concurrent Cyber Physical Systems & sub-systems can be connected in the year 2025 and will keep on increasing over the years. There is a high probability that all the sensor devices won't work individually and will be dependent on other devices as well. For the sake of simplicity, we can consider any number of examples of a big Cyber Physical System like Airplane or Factory, where the Cyber Physical System is the combination of a lot of sub-systems or sensors. All the sensors/sub-systems need to work in tandem to produce an appropriate input for the control server. If there is any error in that input, then the decision taken by the control system will go wrong and can create any big damage to the whole system. Apart from errors, if the sensed data is not synchronized and properly interleaved considering time as a prime factor then it may create a wrong input for the processing systems and feedback results can be severe.

There are many research areas and challenges related to those areas that are associated with Cyber Physical Systems [17, 18]. Some broad areas are shown in Fig. 5.3. Sensors Network (Wired or Wireless), their networking, writing software on different levels, controlling sensor devices with actuators, building information theories for data manipulations or aggregations, and last but not the least writing formal methods in computer science to model and analyse the CPSs in a better way [19] are some major research areas and challenges related to CPSs [20]. All the areas are merged into each other and research required careful effort while solving one issue. For example, while solving issues related to ground level implementation, issues like coordination between different sub-systems [13, 21], synchronization between different sub-systems [22], aggregation of data depending upon the capacity of network, requirement of controlling through feedback loops,

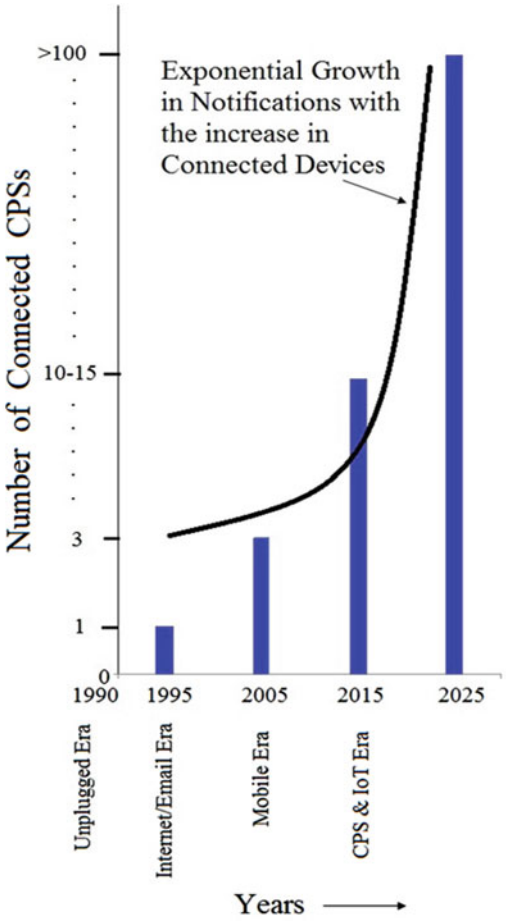



**Fig. 5.2** Expansion of connected devices

delay-tolerance of system in case of real-time system [23] must be taken into consideration as failing in any of the other related part can make the whole system collapse [5]. From the next section onwards, many of the fine-tuned research issues and solutions will be discussed through analytical modeling and intuitive ideas.

For Cyber Physical Systems, there are many fields of applications [24]. Cyber Physical Systems can be used for the energy provision in manufacturing to optimize production processes, for assisted living and also in healthcare, etc. Structured Cyber Physical Systems such as Wireless Sensor Networks (WSNs), Medical Monitoring, Smart Grids, Process Control Systems, Robotics Systems, Auto-Pilot Avionics, etc. [25, 26] are subjected to intense research and development. Efforts are underway to build various special purpose CPSs, without any disciplined effort to innovate theoretical principles required in their analysis and design. In the past few years, control theorists began to intensify research efforts to model and analyse special purpose, structured CPSs [27]. When CONCURRENT Cyber

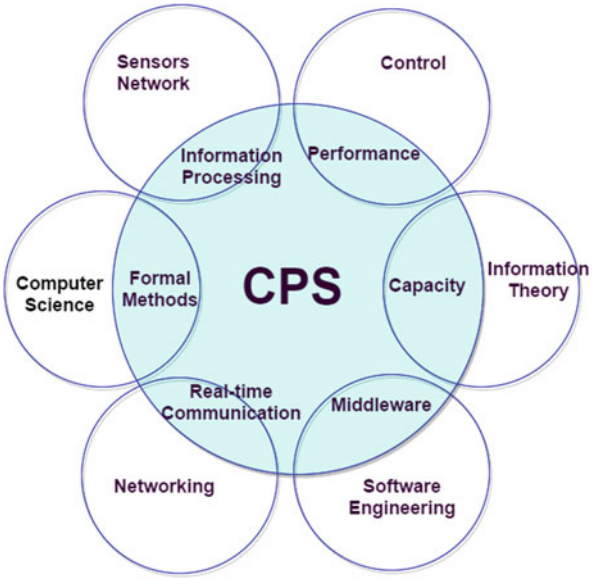



**Fig. 5.3** Formal areas and challenge in CPSs

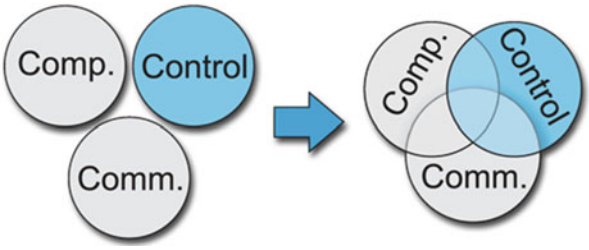



**Fig. 5.4** Joint design of communication, control and compute systems in CPSs

Physical Systems evolve on multiple time scales, synchronizing such systems is an important problem. One interesting solution to this problem is Tensor State Space Representation (TSSR) for modeling and analysis of CONCURRENT Cyber Physical Systems [28]. The literature on dynamical systems [29], modeling of concurrent systems received limited interest. It is clear that if concurrent CPSs need to be analysed, taking CONCURRENCY & SYNCHRONIZATION into account is essential.

The analysis of concurrent and structured CPSs requires tractable models of interconnected dynamical systems [30, 31]. Thus, theoretical efforts (modeling and analysis) related to CPSs are actively pursued by many researchers [32]. Also, design and implementation of prototype CPS are underway at many institutions across the world [33]. The unification of CONTROL, COMMUNICATION and COMPUTATION (Fig. 5.4) functions in multi-dimensional linear dynamical systems have been discovered and formalized. Specifically, optimal control tensors, optimal codeword tensors, optimal logic function tensors are synthesized as the stable states of multi-dimensional neural networks. Furthermore, the synchronization

and temporal semantics are absolutely essential for the control of CPSs evolving on Multiple Time Scales. Thus, the concepts for handling concurrency and time scales are inevitable in the design and implementation of Cyber Physical Systems. The unified theory provides new insights into the global research activity for the design and analysis of concurrent multi-dimensional linear systems evolving on multiple timescales. Also, it will help the research organizations and industries to build special case CPSs.

From systems' viewpoint, we have an interconnected network of linear/nonlinear/hybrid/periodic/aperiodic/distributed/hierarchical dynamical CPSs evolving on multiple time scales [34] (with several feedback loops from control theory viewpoint) [35]. These dynamical systems could be loosely coupled (Buildings' Sensor Networks) or tightly coupled (Wireless Body Area Networks). Some approaches to representing [33] concurrent linear dynamical systems and synchronization of such systems evolving on multiple time scales were discussed in the next section. For instance, a Wireless Body Area Network (monitoring temperature, blood pressure, heart's electrical activity (ECG), brain's electrical activity (EKG), ultrasound, brain imaging-CAT/MRI/FMRI) involves current periodic/aperiodic, linear/nonlinear dynamical systems [36] evolving on multiple time scales. Thus, such a network of Cyber Physical Systems naturally presents important challenges for Smart Hospitals [37, 38]. It is realized that in most such Cyber Physical Systems, in-network distributed computation takes place. Decision fusion takes place at various levels of hierarchical network of Cyber Physical Systems.

Analytical approach to design, monitor, maintain such dynamical systems exists to some extent in the available literature [39]. But, several interesting new challenges naturally arise and need extensive research efforts. For instance, fuzzy logic-based decision support systems present new research challenges. Coming sections in the chapter are going to address some of the important issues related to Concurrency and Synchronization in Structured Cyber Physical Systems.

## 5.2 Concurrent Cyber Physical Systems: Modeling

### *5.2.1 Concurrency in Cyber Physical Systems*

Concurrency is the phenomenon of things to happen in parallel in a system [40]. Cyber Physical System can have many subsystems running in parallel. If they are non-interactive, then there is no need for concurrency control. But if they interact and depend on each other, then serious concurrency control is required. In most Cyber Physical Systems, several physical/physiological processes are concurrently evolving in time on different timescales. For instance, Wireless Body Area Networks (W-BAN) (e.g. sensors monitoring temperature, blood pressure, glucose level, ECG, EKG, etc.) are evolving on different time scales and the information fusion

occurs on different time scales. Coordinating and analysing data generated by such concurrent Cyber Physical Systems is extremely important. Research related to Concurrency Control in Computer Science will provide useful insight into designing and coordinating Concurrent CPSs. It should be kept in mind that several new research issues naturally arise in designing and analysing Concurrent CPSs.

Classifications of Concurrent CPS:

1. Discrete/Continuous Time
2. 1-D/M-D: Input–Output state space representation

Section 5.2.2 explains the solution for maintaining concurrency in Cyber Physical Systems and Sub-systems based on these classifications.

### *5.2.2 Coordination and Maintenance of Concurrent Cyber Physical Systems*

The basic characteristic of Cyber Physical and Embedded Systems is dependency. The key reason is that whatever we do in our information processing, it has an immediate impact on the physical environment. So in case, we do something wrong, we might cause harm to some objects, we might cause harm to people, and that has to be avoided under all circumstances. So, it's not like the office automation where in case there is a problem with some program, we just start the computer again. We have to make sure there will be no harm caused due to CPS.

Cyber Physical and Embedded Systems must be dependable and this comes in a variety of flavours and in a variety of aspects. One of the important aspects is the maintainability which takes into account that systems might fail, and if they fail it should be possible to repair these systems and it shouldn't take too long. So, therefore, we are defining maintainability where maintainability is defined as the probability of a system working correctly t time units after an error occurred. Then as a third aspect, we have availability. Availability is the probability of a system working at a particular time. So, for example, let's suppose that we have a system which very predictively works for 999 h and then fails for 1 h and then works again for 999 h and then fails again. So, such a system would have an availability of 99.9%.

Coordination is must in [41] Concurrent Cyber Physical Systems to make them maintainable and available 24 $\times$ 7. If the concurrent systems or sub-systems lack coordination between them, then the probability of failure increases and on the other hand probability of availability decreases.

### 5.2.3 Design Issues: Concurrent CPSs

Digital integrated circuits are used to realize Cyber Physical System which can be a combinational, sequential or hybrid circuit as per requirement. Combinational integrated circuits can have processing delay at each gate and propagation delay as well. Depending upon the application and its tolerable delay, the design of the system can be verified for meeting the daily requirement of the system. In the sequential Integrated circuit as well, the delay can be calculated at design time and it can be checked that it is meeting the delayed threshold or not? In the sequential circuit, the current output depends upon the past Inputs and outputs and finite state machine associated with such circuits help in finding average tolerable delay. For meeting the requirement of delay and other constants analytical or simulation tools should be used and processor competition power should also be taken into consideration. So we require a tool for digital circuits' performance prediction.

From the present Cyber Physical Systems' implementation, it can be easily observed that temporal semantics are not taken into consideration while designing the software system for any CPS. Most of the available processors run the procedure sequentially so to achieve parallelism in concurrent Cyber Physical Sub-systems, we require careful interleaving of different procedures together [42]. Having multiple hardware processes makes the concurrency task easy but it makes interactions very complex. Involvement of several layers of software, communication between running tasks, separate timing and clock in different processes always make it hard to achieve concurrency on the hardware level. So it's always better that we do this task in software. In software subsystem, it is required to calculate the worst-case execution time. It gives an idea that the system can meet the critical real-time deadlines while designing the Cyber Physical System.

Further, in this section, we are going to discuss modeling method for linearly interacting concurrent Cyber Physical Systems in one/two/multi-dimensions. It is done by stacking the states of concurrent Cyber Physical Systems in different dimensions. Modeling Method can also be generalized for nonlinear concurrent CPSs as well.

### 5.2.4 Modeling Linear Concurrent CPS

Equation (5.1) represents the concurrent physical processes happening at the same time. Their state is represented as a scalar function of time.

$$\{p_i(t) = 1 \leq i \leq N\} \tag{5.1}$$

Equation (5.2) represents these physical processes as an $N$-dimensional vector. It can be seen as the state of all the concurrent coupled scalar valued processes. It is advantageous in examining multiple inputs and outputs in a linear or nonlinear,

time-varying or invariant system.

$$P(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ \vdots \\ p_{N-1}(t) \\ p_N(t) \end{bmatrix} \tag{5.2}$$

If input of systems is represented as $I(t)$, then

$$\frac{d}{dt}P(t) = f_1\,(P, I, t) \tag{5.3}$$

$$Q(t) = f_2\,(P, I, t) \tag{5.4}$$

Equations (5.5) and (5.6) represent the concurrent systems when they are time-varying and Eqs. (5.7) and (5.8) represent when they are time invariant.

$$\frac{d}{dt}P(t) = X_1(t)\ P(t) + X_2(t)\ I(t) \tag{5.5}$$

$$Q(t) = X_3(t)\ P(t) + X_4(t)\ I(t) \tag{5.6}$$

$$\frac{d}{dt}P(t) = X_1\ P(t) + X_2\ I(t) \tag{5.7}$$

$$Q(t) = X_3\ P(t) + X_4\ I(t) \tag{5.8}$$

For the linearly coupled systems, systems are represented as a vector-valued state rather than scalar valued state. $\bar{M}(t)$ (Eq. 5.9) is the *NxN* state matrix of coupled concurrent systems. Equation (5.10) and $\bar{A}(t)$ given in Equation (5.11) is the output matrix linearly coupled concurrent system. Similarly Eqs. (5.12) and (5.13) show the same for time invariant concurrent systems. Here, concurrent processes' state is shown as vector value. The linear coupling matrices $X_1, X_2, X_3, X_4$ are homogeneous for $\bar{M}(t)$.

$$\bar{M}(t) = [S_1(t) : S_2(t) : S_N(t)] \tag{5.9}$$

$$\frac{d}{dt}\bar{M}(t) = X_1(t)\ \bar{M}(t) + X_2(t)\ \bar{I}(t) \tag{5.10}$$

$$\bar{A}(t) = X_3(t)\ \bar{M}(t) + X_4(t)\ \bar{I}(t) \tag{5.11}$$

$$\frac{d}{dt}\bar{M}(t) = X_1\bar{M}(t) + X_2\ \bar{I}(t) \tag{5.12}$$

$$\bar{A}(t) = X_3\bar{M}(t) + X_4\ \bar{I}(t) \tag{5.13}$$

Equations (5.14) and (5.15) show the state space representation of discrete time concurrent Cyber Physical Systems/Sub-systems. Same way state space representation of time-invariant Cyber Physical Systems in discrete time can be generated.

$$\bar{M}\ (x+1) = X_1(x)\ \bar{M}(x) + X_2(x)\ \bar{I}(x) \tag{5.14}$$

$$\bar{A}(n) = X_3(x)\ \bar{M}(x) + X_4(x)\ \bar{I}(x) \tag{5.15}$$

Instead of coupling matrices, 3D arrays can also be used to the present homogeneous linear concurrent systems rather than using coupling matrices $\{X_1, X_2, X_3, X_4\}$. Outer product computation using tensor "$X$" and "$S$" is given below. It will generate a 5D array "$R$".

$$X_{i_1,i_2,i_3}.S_{j_1,j_2} = R_{k_1,k_2,\ k_3,k_4,k_5}$$

Equations (5.16) and (5.17) show the state space representation of heterogeneous linear Concurrent Cyber Physical Systems. They are required to be represented using multi-dimensions rather than single dimension so a matrix or a 3D array or a multidimensional array (tensor) is used to represent such states. Systems can be in discrete or continuous time as well. Here discrete time is taken into consideration.

$$S\ (x+1) = X_1(x)\ S(x) + X_2(x)\ I\ (x) \tag{5.16}$$

$$T\ (n) = X_3(x)\ S(x) + X_4(x)\ I\ (x) \tag{5.17}$$

For multidimensional linear Cyber Physical Systems, Eqs. (5.18) and (5.19) are evolved as shown below. Here $X_1(x)$ is a state coupling multidimensional tensor presented with the order 2r. $S(x+1)$ is the system state at '$x+1$' discrete time index. Similarly $S(x)$ is the state at '$x$' time index. $X_2(x)$ is the input coupling multidimensional tensor with the order represented as '$r+1$'. $T(x)$ is the multidimensional output tensor with the order represented as '$x$ '. $I(x)$ is the varying multidimensional inputs answer with the order 'l'. $X_3(x)$ and $X_4(x)$ are the multidimensional tensors with the orders '$s+r$' and '$s+$ l' respectively, used as tenses to the output dynamics.

$$S_{i_1,\ldots,i_3}(x+1) = X_{(i_1,\ldots,i_{r\,j_1,\ldots,j_r})}(x).S_{j_1,\ldots,j_r}(x) + X_{(i_1,\ldots,i,_{j_1,\ldots,j_l})}(x).I_{j_1,\ldots,j_l}(x) \tag{5.18}$$

$$T_{q_1,\ldots,q_3}(x+1) = X_{(q_1,\ldots,q_{s\,j_1,\ldots,j_r})}(x).S_{j_1,\ldots,j_r}(x) + X_{(q_1,\ldots,q_{s\,j_1,\ldots,j_l})}(x).I_{j_1,\ldots,j_l}(x) \tag{5.19}$$

## 5.3 Synchronization of Concurrent Cyber Physical Systems

### *5.3.1 Networked Cyber Physical Systems: Synchronization*

The approach for synchronizing the Cyber Physical Systems discussed earlier requires complete knowledge of system dynamics. In many [43] CPS, the dynamics are not completely/perfectly known. For synchronization of such systems, the following approach could be utilized.

We need to estimate the maximum delay involved in knowing the output for worst case input (it should be noted that the system could be implemented in software) [44, 45]. The idea is best illustrated by a parallel connection of software/hardware systems. The systems require processing delays (for worst case input) which are all not necessarily equal.

Hence, the system to which the outputs of parallel systems constitute input must receive all the inputs in a synchronized manner. Thus, as shown in Fig. 5.5, delay buffers are necessary. In general, based on the network topology, synchronization is achieved by optimally buffering/delaying the inputs/outputs in an intelligent manner [46, 47]. Such an approach is already utilized in some robotics applications, for instance, Critical Path Method (CPM) is well known. Synchronization of CPS should be concentrated on the following classifications:

- Replicated Decoupled/non-interacting CPSs
- Replicated Coupled/interacting CPSs

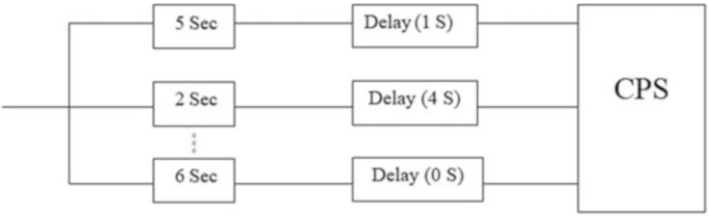



**Fig. 5.5** Delay buffers in parallel systems

### 5.3.2 Clock's Synchronization: Multidimensional CPSs

As most of the physical processes in Cyber Physical Systems happen at multiple time scales [48, 49] the mark distributed coupled linear CPSs on different time scales [50]. Representation of search Cyber Physical Systems and their respective States can be represented as given in Eqs. (5.20), (5.21), and (5.22).

$$\begin{bmatrix} p_1(x) \\ p_2(x) \end{bmatrix} = \begin{bmatrix} r_{11}\ r_{12} \\ r_{21}\ r_{22} \end{bmatrix} \begin{bmatrix} p_1\left(\frac{x}{t_1}\right) \\ p_2\left(\frac{x}{t_2}\right) \end{bmatrix} + \begin{bmatrix} s_{11}\ s_{12} \\ s_{21}\ s_{22} \end{bmatrix} \begin{bmatrix} q_1\left(\frac{x}{u_1}\right) \\ q_2\left(\frac{x}{u_2}\right) \end{bmatrix} \tag{5.20}$$

Dynamics of linearly interacting concurrent Cyber Physical Systems on multiple time scales are represented in Eq. (5.20), where $\{t_1, t_2, u_1, u_2\}$ is a scalar value related to discrete time and happening on separate time scales. The value is >1 and they are coupled.

$$\text{LCM } \{t_1, t_2\} = u \text{ and } x' = \frac{x}{u} \tag{5.21}$$

LCM of $\{t_1, t_2\}$ is calculated in Eq. (5.21).

$$\begin{bmatrix} p_1\left(ux'\right) \\ p_2\left(ux'\right) \end{bmatrix} = \begin{bmatrix} r_{11}\ r_{12} \\ r_{21}\ r_{22} \end{bmatrix} \begin{bmatrix} p_1\left(f_1x'\right) \\ p_2\left(f_2x'\right) \end{bmatrix} + \begin{bmatrix} s_{11}\ s_{12} \\ s_{21}\ s_{22} \end{bmatrix} \begin{bmatrix} q_1\left(f_1x'\right) \\ q_2\left(f_2x'\right) \end{bmatrix} \tag{5.22}$$

After applying Eq. (5.21) to Eq. (5.20), we got Eq. (5.22). A common global clock can be defined using LCM of $\{u, t_1, t_2\}$. It solves the problem of synchronization in Cyber Physical System by defining the system on a single time scale.

## 5.4 Temporal Semantics: Design of Cyber Physical Systems

Design issues related to embedded systems naturally provide insights into the choices of hardware and/or software systems employed in CPS [51, 52]. But various emerging CPSs naturally require time-critical guarantees on the processing of the data generated by the Concurrent Cyber Physical Systems evolving on different time scales [53]. The research related to REAL TIME OPERATING SYSTEMS (RTOS) will provide interesting insights into providing time-critical guarantees on results generated by various sub-systems in CPSs. Thus, the design of SOFTWARE SYSTEMS must necessarily ensure that temporal semantics [54] are taken into account. Existing programming abstractions (paradigms) must necessarily be redesigned to ensure that real-time performance guarantees are provided.

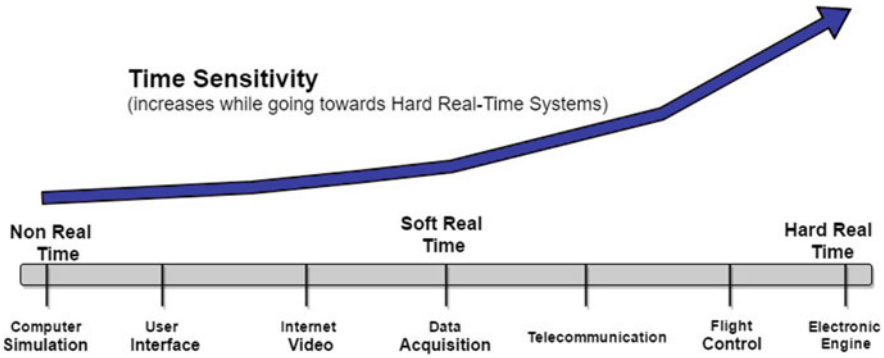



**Fig. 5.6** Real-time and non-real-time CPSs

### *5.4.1 Classification of Cyber Physical Systems (CPSs)*

One possible classification of Cyber Physical Systems (CPS) is [55] (Fig. 5.6)

- Real-Time CPS and
- Non-Real-Time CPS

A *hard real-time system* is one way or the computation has no value whatsoever if the time constraint is not met [56, 57]. Examples are control systems for industrial processes, Air Traffic control, and vehicles' subsystem control, etc. [58, 59]. The *soft real-time system* is the one having the property of timeliness of a computation where the value of computation decreases according to its tardiness. Examples are data acquisition systems, telecommunication, internet video, VoIP, etc. *Non-real-time systems* are the one having no time deadline at all to complete any task. If there is one, then failing to meet that deadline doesn't affect the system at all. One example is a computer simulation.

Considering Real-Time CPSs, deadlines are of two types [60]:

- Those which can be met with software/hardware resources.
- Those which can't be met. This can happen and there is no hope, e.g. Disaster.

Since any CPS involves hardware and software sub-systems [61, 62], it is essential that these sub-systems meet the time-critical deadlines. Otherwise, such real-time CPS will be unable to meet the performance guarantees essential for the operation of such systems.

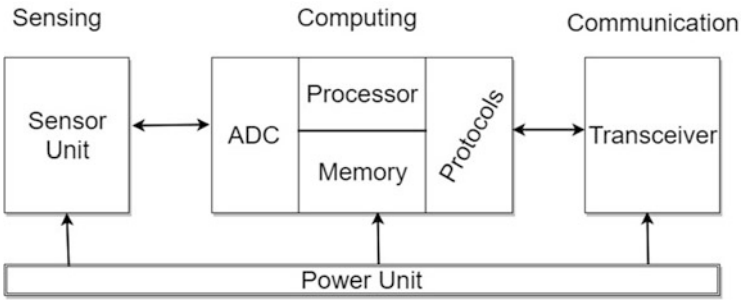



**Fig. 5.7** Sensor node architecture

## *5.4.2 Real-Time CPSs: Temporal Semantics*

### 5.4.2.1 Goal: Optimal Utilization of Software/Hardware Resources to Meet the Time-Critical Deadline

We first consider hardware subsystems. Since any CPS involves the integration of cyber world and physical world, most of the time a sensor/transducer (wired or wireless) is essential. Most sophisticated sensors have (Fig. 5.7) (a) Transducer; (b) Communication unit (transceiver in the case of wireless sensors), (c) Computation unit, (d) Analog-to-Digital Converter, (e) power unit, etc.

The current state of technology determines the speed of A-to-D converter. It imposes units on how fast digital signal is generated from the analog signal. Such technological limitations must be kept in mind when a CPS is designed. In effect, only currently realizable (technology wise) time critical deadlines can be met in current CPS. It should be noted that through "intelligent" utilization of available resources (e.g. time, memory, processors, etc.), sometimes critical deadline can be met. We illustrate an innovative idea which enables meeting temporal deadlines related to computation at the sensor.

We first state the problem of time-critical computation problem [63, 64] at a sensor/embedded system and how it can be solved through intelligent computation. Consider a temperature sensor monitoring a phenomenon and is required to locally compute a function such as mean, min, max, etc. Specifically, minimum fusion function needs to be computed within a "Critical Time", e.g. $T_0$. It is clear that as the number of sensed samples increases, the computation time with existing computation hardware proportionately increases. We propose the idea of TEMPORAL DIVISION OF LABOUR (Fig. 5.8), i.e. as and when the sensed temperature data becomes available locally the fusion function is immediately computed and stored in memory (so that a large number of samples are not accumulated). Thus, proactive intermediate computation of fusion function and storage of local fusion function values can be utilized to efficiently compute the fusion function over a larger timescale. This approach enables the sensor to meet the critical time deadline. In this case, "efficient" utilization of computing resources potentially enables the deadline to be met.

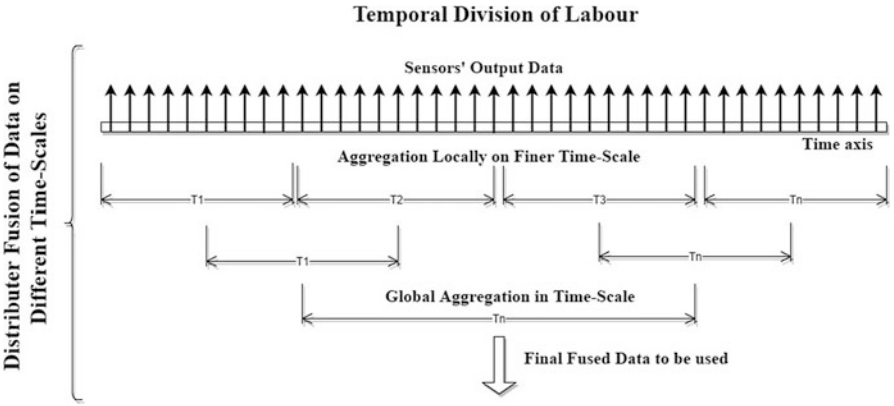



**Fig. 5.8** Temporal division of labour

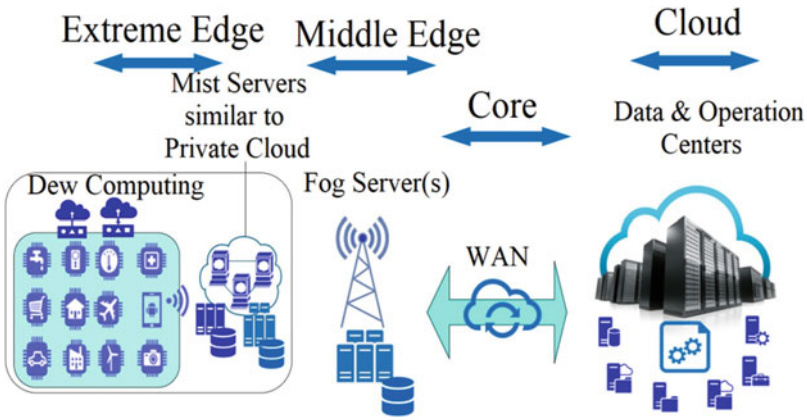



**Fig. 5.9** CPSs with edge computing and cloud

Example: MEAN of $N$ samples -> '$T_0$' Time. Buffer samples on a finer time scale '$d_0$' store the value, i.e. mean on a Finer time scale $Nd_0$ samples on a Coarser timescale takes more time.

### 5.4.2.2 Effective Idea: Edge Computing

If we require to manage real-time (with hard and soft deadlines) and non-real-time (or relaxed real-time computation) computation, the hierarchical architecture of cloud-fog-mist-dew computing can be utilized as shown in Fig. 5.9. This is also called Edge Computing architecture. Edge Computing has emerged as an advanced technology for future CPSs and IoTs. It seems to solve the issues of managing delays in hard real-time applications with ease. Along with this, it is going to resolve the issue of excessive bandwidth requirement in core Internet in coming years.

Edge Computing is the computing in Cloud Servers near to the physical sensing environment [65, 66]. Cloud Servers on Edge are also called Fog Servers or Cloudlets [67–69]. They can be just a few hops away from the sensors so decreases the delay dramatically for real-time applications. Even with the lower configuration than a Cloud Server, Fog Server [70, 71] gives better performance by removing the delays of data transmission in Core Internet [72]. This [73] Cloud Hierarchy can be extended further to the Extreme Edge, where computation is done in Mist Servers situated in the vicinity of the sensors just like a private Cloud. There can be many Mist servers available in the boundaries of an organization/institute. Mist servers further manage the delay-sensitive applications better, for example, managing industrial pipeline issues, fire alarms, etc. [74, 75] Computation in Extreme Edge can be done in Sensor Nodes as well [76]. Processors nowadays have the capability to manage some data locally and can do local computations. This is called Dew Computing. There can be local fusion at a sensor on a finer time scale followed by storage locally or fusion over a coarser time scale [77] on stored samples meeting the critical time deadline. Thus, we have a temporal hierarchical fusion of sensed data [78].

A similar idea is applied for opportunistic Communication [79] and control. For this, we can sense all the free channels at any point of time, snatch the band when available and locally transfer to the intermediate node and then to the destination node. This idea can be applied to Edge Computing as well to transmit data over to the Cloud from Fog Server.

### *5.4.3 Software System: Temporal Semantics*

In the design and analysis of algorithms, various sorting algorithms are thoroughly understood/analysed for time complexity as well as memory complexity [80, 81]. Thus, any application (for instance, sorting marks of students) which uses an efficient sorting algorithm can be predicted for the amount of time taken for generating the output (for worst case input). This conclusion is true for various other algorithm based applications. But there are other software systems (e.g. software switches/routers) for which estimating the time required for getting desired output cannot be easily predicted. Formal computational models of special/general purpose software systems are very useful to determine the time to generate an output given worst case input [81–83]. Software testing approaches could also be adopted for estimating the time complexity.

Most software systems are designed to be modular involving "Functions/Subroutines" which can be tested for estimating the time complexity under worst case input. Most Cyber Physical Systems are based on software and/or hardware systems. Thus, temporal semantics must take into account the architecture involving the integration of software and hardware systems.

It should be kept in mind that extensive testing of distributed software/hardware systems may be prohibitive in terms of time required and may not be possible. Thus, new innovative solutions need to be discovered.

## 5.5 Reliability and Fault Tolerance: Concurrent Cyber Physical Systems

### *5.5.1 Fault Tolerance*

The design of [84] Concurrent Cyber Physical Systems evolving on multiple time scales must be fault tolerant and failure tolerant [85]. Such a design is very critical since failures/faults in sub-systems could be very disastrous in various applications [86]. Fault/failure tolerant system design is investigated by various researchers. Such results provide useful insight in the case of Cyber Physical System design. But the highly interconnected network of Cyber Physical Systems presents various new challenges which are not addressed in the traditional fault-tolerant design.

### *5.5.2 Fault/Failure Localization:*

From a coarse description viewpoint, a graph captures the topology of the interconnected network of Cyber Physical System that potentially spatially distributed and evolving in time on multiple (time) scales.

Diagnostic subsystem embedded into the CPS network must be designed such that fault/failure localization and repair can be done in real-time [87]. In other words, a network of CPSs should be designed for fault/failure tolerance. Well-known approaches such as Design For Testability (DFT), BUILT In Self Test (BIST) in embedded system design provide useful indications/hints.

From a control theoretic viewpoint [88], coupled CPSs involving many feedback loops must be analytically and/or simulation-wise be examined for "stability" issues. In fact, new technologies are required when some of the CPSs are nonlinear. Issues such as positive feedback, resonance must be tested so that small disturbance/noise are not amplified significantly leading to failure of subsystems. The design of the architecture of networked CPS will involve new problems, solutions from a graph/hypergraph theoretic viewpoint.

### 5.5.3 Architectural Considerations: Cyber Physical Systems

It is essential that the architecture of any CPS is able to localize fault (fault detection and localization) and be designed in such a way that the entire system is fault tolerant.

In the case of distributed, networked Cyber Physical System there are at least two categories with respect to the topology:

- Topology of networked CPS is under the control of user: Controlled Deployment
- Topology of networked CPS is not under the control of user: Uncontrolled Deployment

**Example** Consider a wireless sensor network deployed in a building to detect and communicate FIRE in the building. In this case, the user can control the topology/deployment of the wireless network.

Suppose the wireless sensor network is deployed in a forest (using say Helicopter) to monitor and communicate FIRE. The topology/deployment is not under the control of the user. We can also run into distributed networked CPS in which the topology of a portion of the system is under the control of the user and some portion is not under the control of the user. It should be noted that those CPS whose topology is under the control of user are easy to monitor/diagnose and operate.

We are thus naturally led to the design of distributed CPS that are easy to diagnose/monitor and maintain. As in digital telephone networks, the CPS is integrated with an Operations Support System (OSS) for diagnosing/ monitoring/maintaining various subsystems. For such purpose, we propose an overlay network. Such operations support system could be based on modeling abstractions of distributed CPS. The main challenge is that the dynamics of the physical/physiological system could only be partially understood.

In some cases, the dynamics are highly nonlinear and not well understood. One usually resorts to simulations to partially understand the dynamics of such systems. Let us consider the example of Wireless Body Area Network (WBAN). Even though diseases of various organs and their functioning/malfunctioning is understood to a limited extent, well-designed WBAN can monitor the health of various interacting organs in real-time and can provide good diagnostic support to the doctor. The equipment such as ECG machine, MRI scanner can be endowed with software to analyse the 1-D/2-D/3-D signal locally and provide diagnostic help. Also, the equipment could be locally networked to monitor the health of various interacting organs.

In reliability theory applied to fault-tolerant computing systems, various innovative ideas are proposed [89]. Some of those ideas are also applicable to Cyber Physical Systems [90]. Various types of redundancy (e.g. 1-in-*N* redundant systems) are incorporated to make the entire system sufficiently reliable. By means of test inputs fed to various CPS (Wired Communication or Wireless Communication), it could be possible to monitor the health of various subsystems in a large system. Thus Operations Support System can monitor the health of distributed CPS.

The architecture of wired/wireless sensors could be modified to be able to locally compute fusion functions such as Mean, Median, Mode, etc. on a finer timescale. For instance, underwater sensors can be designed to monitor the earthquake/tsunami. Also, existing sensor designs should be modified in such a way that the architecture is tailored to the MAC, routing protocols utilized to network the sensors.

### *5.5.4 Reliability and Fault Management Using Edge Servers*

Internet of Things (IoTs) has emerged as a mechanism for connecting physical world and the cyber world via Internet [91]. So while talking about the reliability and fault tolerance in CPS, the whole mechanism can be represented as IoTs alternatively.

#### 5.5.4.1 Reliability and Backup Policy

One of the important aspects in CPSs is reliability [92]. Reliability refers to the probability of a system working correctly, provided it was working correctly at time t equals zero. That means we are considering the probability over time that the system is working.

To construct a reliable CPS or IoT-Cloud infrastructure the replication of sensed data is very important [93]. The redundancy in data makes sure the feedback for the actuators based on the past sensed data [94, 95]. The backup system can take over the control in case of any failure. For Clouds, we use the concept of AZ (Availability Zone) for backup that can be used in case of any disaster or failure in the Cloud server [96].

Now while using the concept of Edge Computing, the focus is on running CPS Applications from the Edge of the network rather than from the cloud that is so far away from the end sensor devices. So for all the possibilities [97], Fog and Mist level servers should also have backup servers on their levels to run the applications from alternative servers in case of failure of any one of them.

The multi-layered architecture is reliable [98] in the sense that even if there are no alternative servers on the same layer having data replications, there is a guaranty of replication of sensed data on the higher and upper layer. The volume of data for the same sensor may vary on Dew, Mist, Fog and Cloud layers [99] but it would be there on all layers to take over the system in case of absence of any server on any layer.

#### 5.5.4.2 Fault Tolerance and Agent

By definition, Mobile Agent is a special purpose software code that can transfer itself from one machine to another, but practically it is the same code running on all the machines and just data gets transmitted from one system to another. Mobile Agents are very helpful in managing distributed systems as there is no central system to manage them. In this proposed work, the faults [100, 101] in the whole hierarchy will be managed by Mobile Agents.

Here Agent will work as Resource and Network Monitoring Agent. It will share the neighbour information and link state information with other agents on alternative servers [101], if available. Except monitoring, they will also be responsible for assigning the priority to the CPS applications depending upon their delay-tolerance. This priority information will be used at the time of load distribution [102] of a particular server in case of impacted failure or impacted scheduled shutdown activity [103]. It will also help in new path discovery [104] after the load distribution in the case of faults. It will also keep a check on timing intervals of monitoring and backing up the data.

Figure 5.10 shows the recovery process in case of faults and scheduled shutdowns, reactively and proactively in respective case. It is depicting the complete fault tolerance cycle of the hierarchy between Dew, MIST, Fog and cloud. Here the fault tolerance has been achieved by exploiting the capabilities and benefits of an agent, which is basically a software program running on each server. Also at the time of any fault occurrence, the whole responsibility has been assumed/assigned for upper layer agent, not the affected layer's agent.

## 5.6 Agent Working in Different Conditions

1. As per assumption, the reason behind any fault/shutdown can be either any scheduled activity or any sudden activity.
2. Any sudden or shutdown activity comes with two possibilities that either it's going to affect respective server/device or no effect at all on server/device.
3. So as per represented in the graph if the ongoing activity is effectless then for scheduled event agent will move with proactive actions which are basically to send notifications to the all respective whereas for sudden activity agent will look for reactive action.
4. For any sudden fault, the agent will fetch priority index for all applications of the affected server and immediately it will check if any other server available on the same layer [104, 105] (either in same or different availability zone). After that it will do application migration [106] and connection redirection and then will do the load transfer activity as per the sequence given below:

Total load $= C$
Server $S$ can handle (load) $= T$

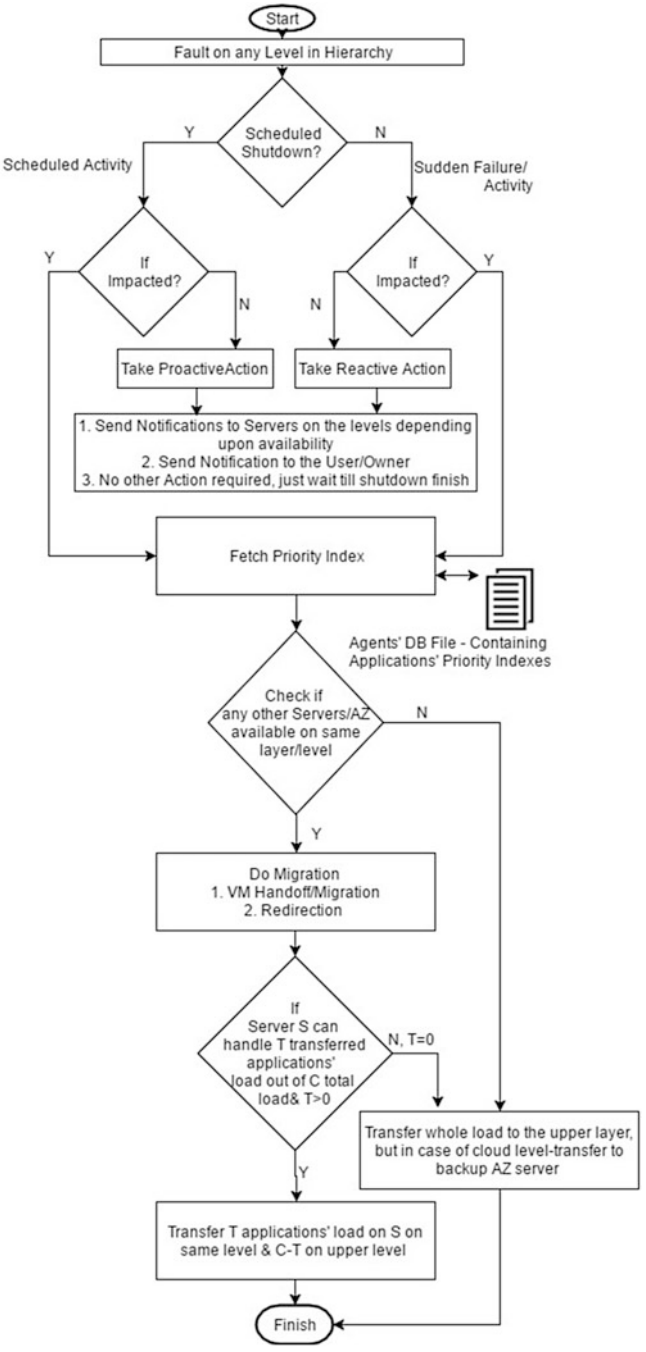



**Fig. 5.10** Agents' working in faults

if $T > 0$,
then

Transfer $T$ (load) $\rightarrow S$
Transfer ($C$–$T$) load $\rightarrow$ upper layer

else if $T = 0$

Transfer whole load $\rightarrow$ upper layer

5. But unfortunately, if no other same layer server is available, then it will point for the upper layer to transfer the whole load.

## 5.7 Conclusion

This book chapter is all about providing appropriate models and solutions for Concurrent Cyber Physical Systems. The chapter takes into consideration the concurrency and the synchronization requirement of the systems. Tensor State Space Representation (TSSR) is used to model linear concurrent Cyber Physical System. The model works for both homogeneous and heterogeneous type of systems. The chapter also defined the idea of delay buffers for synchronizing multiple subsystems and model for clock synchronization of multidimensional Cyber Physical Systems. Importance of Time or Temporal semantics can't be denied in concurrent systems. Importance of time in real-time Cyber Physical Systems is discussed and a time-oriented solution "Temporal Division of Labour" for data fusion on multiple levels of systems is represented. At last, one advanced technology Edge Computing was discussed as a solution to multiple issues. It has been shown that how the hierarchical Cloud can manage temporal data and how it can also deal with reliability, fault tolerance and fault localization issues. All the formal discussions regarding CPSs are very important to have a theoretical understanding of issues and their solutions.

**Acknowledgments** The authors gratefully acknowledge the helpful suggestions from Dr. Shivangi Katiyar, Assistant Professor, CGC Landran. This book chapter is entirely written in MS Word and all the figures are created on website https://www.draw.io/.

## References

1. Lee EA, Seshia SA (2016) Introduction to embedded systems: a cyber-physical systems approach. MIT Press, Cambridge
2. Kopetz H (2011) Real-time systems: design principles for distributed embedded applications. Springer, Berlin
3. Givargis T, Russo S (eds) (2008) Software technologies for embedded and ubiquitous systems: 6th IFIP WG 10.2 international workshop, SEUS 2008, Anacarpi, Capri Island, Italy, Oct 1–3, 2008, Revised Papers (Vol. 5287). Springer

4. Lee J, Bagheri B, Kao HA (2015) A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf Lett 3:18–23
5. Jensen JC, Chang DH, Lee EA (2011) A model-based design methodology for cyber-physical systems. In: Wireless communications and mobile computing conference (IWCMC), 2011 7th international, IEEE, pp 1666–1671
6. Marwedel P (2010) Embedded system design: embedded systems foundations of cyber-physical systems. Springer, Berlin
7. Lee EA (2006) Cyber-physical systems-are computing foundations adequate. In: Position paper for NSF workshop on cyber-physical systems: research motivation, techniques and roadmap (Vol. 2)
8. Baheti R, Gill H (2011) Cyber-physical systems. Impact Control Technol 12:161–166
9. Xiong G, Zhu F, Liu X, Dong X, Huang W, Chen S, Zhao K (2015) Cyber-physical-social system 5n intelligent transportation. IEEE/CAA J Automatica Sinica 2(3):320–333
10. Camarinha-Matos LM, Falcão AJ, Vafaei N, Najdi S (eds) (2016) Technological innovation for cyber-physical systems: 7th IFIP WG 5.5/SOCOLNET advanced doctoral conference on computing, electrical and industrial systems, DoCEIS 2016, Costa de Caparica, Portugal, April 11–13, 2016, Proceedings (Vol. 470). Springer
11. Khaitan SK, McCalley JD (2015) Design techniques and applications of cyberphysical systems: a survey. IEEE Syst J 9(2):350–365
12. Wirsing M, Banâtre JP, Hölzl M, Rauschmayer A (eds) (2008) Software-intensive systems and new computing paradigms: challenges and visions (Vol. 5380). Springer
13. He J, Chen J, Cheng P, Cao X (2014) Secure time synchronization in wireless sensor networks: a maximum consensus-based approach. IEEE Transac Parallel Distrib Syst 25(4):1055–1065
14. Freris NM (2017) A software defined architecture for cyberphysical systems. In: Software defined systems (SDS), 2017 fourth international conference on, IEEE, pp 54–60
15. Lee EA (2007) Computing foundations and practice for cyber-physical systems: a preliminary report. University of California, Berkeley Technical Report UCB/EECS-2007-72
16. Lee EA (2015) The past, present and future of cyber-physical systems: a focus on models. Sensors 15(3):4837–4869
17. Liu Y, Peng Y, Wang B, Yao S, Liu Z (2017) Review on cyber-physical systems. IEEE/CAA J Automatica Sinica 4(1):27–40
18. Siaterlis C, Genge B, Hohenadel M (2013) EPIC: a test bed for scientifically rigorous cyber-physical security experimentation. IEEE Transac Emerg Topics Comput 1(2):319–330
19. Agha G, Danvy O, Meseguer J (eds) (2011). Formal modeling: actors; open systems, biological systems: essays dedicated to carolyn talcott on the occasion of her 70th birthday (Vol. 7000). Springer
20. Ungureanu G, Sander I (2017) A layered formal framework for modeling of cyber-physical systems. In: 2017 design, automation & test in Europe conference & exhibition (DATE), IEEE, pp 1715–1720
21. Andrade HA, Derler P, Eidson JC, Li-Baboud YS, Shrivastava A, Stanton K, Weiss M (2015) Towards a reconfigurable distributed testbed to enable advanced research and development of timing and synchronization in cyber-physical systems. In: ReConFigurable computing and FPGAs (ReConFig), 2015 international conference on, IEEE, pp 1–6
22. Rajkumar RR, Lee I, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th design automation conference. ACM, New York, pp 731–736
23. Kim KD, Kumar PR (2012) Cyber–physical systems: a perspective at the centennial. Proc IEEE 100(Special Centennial Issue):1287–1308
24. Shi J, Wan J, Yan H, Suo H (2011) A survey of cyber-physical systems. In: Wireless communications and signal processing (WCSP), 2011 international conference on, IEEE, pp 1–6
25. Wu FJ, Kao YF, Tseng YC (2011) From wireless sensor networks towards cyber physical systems. Pervasive Mobile Comput 7(4):397–413

26. Hu F, Lu Y, Vasilakos AV, Hao Q, Ma R, Patil Y, Xiong NN (2016) Robust cyber–physical systems: concept, models, and implementation. Futur Gener Comput Syst 56:449–475
27. Derler P, Lee EA, Sangiovanni-Vincentelli AL (2011) Addressing modeling challenges in cyber-physical systems (No. UCB/EECS-2011-17)
28. Lee EA (2008) Cyber physical systems: design challenges. In: Object oriented real-time distributed computing (isorc), 2008 11th IEEE international symposium on, IEEE, pp 363–369
29. Chandhoke S, Hayles T, Kodosky J, Wang G (2011) A model-based methodology of programming cyber-physical systems. In: Wireless communications and mobile computing conference (IWCMC), 2011 7th international, IEEE, pp 1654–1659
30. Tan Y, Goddard S, Perez LC (2008) A prototype architecture for cyber-physical systems. ACM Sigbed Rev 5(1):26
31. Artho C, Legay A, Peled D (2016) Automated technology for verification and analysis. Springer, Berlin
32. Lee EA (2010) CPS foundations. In: Design automation conference (DAC), 2010 47th ACM/IEEE, IEEE, pp 737–742
33. Liu Z, Yang DS, Wen D, Zhang WM, Mao W (2011) Cyber-physical-social systems for command and control. IEEE Intell Syst 26(4):92–96
34. Al-Nayeem A, Sha L, Cofer DD, Miller SM (2012) Pattern-based composition and analysis of virtually synchronized real-time distributed systems. In: Cyber-physical systems (ICCPS), 2012 IEEE/ACM third international conference on, IEEE, pp 65–74
35. Kim M, Stehr MO, Kim J, Ha S (2010) An application framework for loosely coupled networked cyber-physical systems. In: Embedded and ubiquitous computing (EUC), 2010 IEEE/IFIP 8th international conference on, IEEE, pp 144–153
36. Gelenbe E, Wu FJ (2013) Future research on cyber-physical emergency management systems. Future Internet 5(3):336–354
37. Sztipanovits J, Koutsoukos X, Karsai G, Kottenstette N, Antsaklis P, Gupta V, Wang S (2012) Toward a science of cyber–physical system integration. Proc IEEE 100(1):29–44
38. Conti M, Das SK, Bisdikian C, Kumar M, Ni LM, Passarella A, Zambonelli F (2012) Looking ahead in pervasive computing: challenges and opportunities in the era of cyber–physical convergence. Pervasive Mobile Comput 8(1):2–21
39. Mosterman PJ, Zander J (2016) Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems. Softw Syst Modeling 15(1):5–16
40. Garimella RM (2014) Concurrent cyber physical systems: tensor state space representation. In: Control & Automation (ICCA), 11th IEEE international conference on, IEEE, pp 1232–1237
41. Xiao K, Ren S, Kwiat K (2008) Retrofitting cyber physical systems for survivability through external coordination. In: Hawaii international conference on system sciences, proceedings of the 41st annual, IEEE, pp 465–465
42. Balani R, Wanner LF, Friedman J, Srivastava MB. Lin K, Gupta RK (2011) Programming support for distributed optimization and control in cyber-physical systems. In: Proceedings of the 2011 IEEE/ACM second international conference on cyber-physical systems, IEEE computer society, pp 109–118
43. Eidson JC, Lee EA, Matic S, Seshia SA, Zou J (2012) Distributed real-time software for cyber–physical systems. Proc IEEE 100(1):45–59
44. Fink J, Ribeiro A, Kumar V (2012) Robust control for mobility and wireless communication in cyber–physical systems with application to robot teams. Proc IEEE 100(1):164–178
45. Deng X, Yang Y (2013) Communication synchronization in cluster-based sensor networks for cyber-physical systems. IEEE Transac Emerg Topics Comput 1(1):98–110
46. Yadav A, Waghmare A, Sairam AS (2014) Exploiting node heterogeneity for time synchronization in low power sensor networks. In: Contemporary computing and informatics (IC3I), 2014 International conference on, IEEE, pp 828–832
47. Yokoyama T, Matsubara A, Yoo M (2015) A real-time operating system with gnss-based tick synchronization. In: Cyber-physical systems, networks, and applications (CPSNA), 2015 IEEE 3rd international conference on, IEEE, pp 19–24

48. Shih CS, Yang CM, Cheng YC (2015) Data alignment for multiple temporal data streams without synchronized clocks on iot fusion gateway. In: Data science and data intensive systems (DSDIS), 2015 IEEE international conference on, IEEE, pp 667–674
49. Lee EA, Matic S, Seshia SA, Zou J (2009) The case for timing-centric distributed software invited paper. In: Distributed computing systems workshops, 2009. ICDCS Workshops' 09. 29th IEEE international conference on, IEEE, pp 57–64
50. Subramanian V, Umbarkar A, Doboli A (2012) Decentralized event detection using distributed interrupts in Cyber Physical Systems. In: Systems conference (SysCon), 2012 IEEE international, IEEE, pp 1–6
51. Lee EA (2009) Computing needs time. Commun ACM 52(5):70–79
52. Tan Y, Vuran MC, Goddard S (2009) Spatio-temporal event model for cyber-physical systems. In: Distributed computing systems workshops, 2009. ICDCS Workshops' 09. 29th IEEE international conference on, IEEE, pp 44–50
53. Broman D, Derler P, Eidson J (2013) Temporal issues in cyber-physical systems. J Indian Inst Sci 93(3):389–402
54. Eidson J, Lee EA, Matic S, Seshia SA, Zou J (2010) A time-centric model for cyber-physical applications. In: Workshop on model based architecting and construction of embedded systems (ACES-MB), pp 21–35
55. Georg H, Muller SC, Dorsch N, Rehtanz C, Wietfeld C (2013) INSPIRE: integrated co-simulation of power and ICT systems for real-time evaluation. In: Smart grid communications (SmartGridComm), 2013 IEEE international conference on, IEEE, pp 576–581
56. Anwar F, D'souza S, Symington A, Dongare A, Rajkumar R, Rowe A, Srivastava M (2016) Timeline: an operating system abstraction for time-aware applications. In: Real-time systems symposium (RTSS), 2016 IEEE, pp 191–202 IEEE
57. Bui D, Lee E, Liu I, Patel H, Reineke J (2011) Temporal isolation on multiprocessing architectures. In: Proceedings of the 48th design automation conference. ACM, New York, pp 274–279
58. Liu K, Lee VCS, Ng JKY, Chen J, Son SH (2014) Temporal data dissemination in vehicular cyber–physical systems. IEEE Trans Intell Transp Syst 15(6):2419–2431
59. Sun H, Liu J, Chen X, Du D (2015) Specifying cyber physical system safety properties with metric temporal spatial logic. In: Software engineering conference (APSEC), 2015 Asia-Pacific, IEEE, pp 254–260
60. Weiss M, Chandhoke S, Melvin H (2015) Time signals converging within cyber-physical systems. In: Frequency control symposium & the European frequency and time forum (FCS), 2015 joint conference of the IEEE international, IEEE, pp 684–689
61. Dai W, Vyatkin V, Pang C, Christensen JH (2015) Time-stamped event based execution semantics for industrial cyber-physical systems. In: Industrial informatics (INDIN), 2015 IEEE 13th international conference on, IEEE, pp 1263–1268
62. Eidson JC, Stanton KB (2015) Timing in cyber-physical systems: the last inch problem. In: Precision clock synchronization for measurement, control, and communication (ISPCS), 2015 IEEE international symposium on, IEEE, pp 19–24
63. Petnga L, Austin M (2013) Ontologies of time and time-based reasoning for MBSE of cyber-physical systems. Procedia Comput Sci 16:403–412
64. Broman D, Zimmer M, Kim Y, Kim H, Cai J, Shrivastava A, Lee EA (2013) Precision timed infrastructure: design challenges. In: Electronic system level synthesis conference (ESLsyn), 2013, IEEE, pp 1–6
65. Patel M, et al (2014) Mobile-edge computing introductory technical white paper. White paper, Mobile-edge Computing (MEC) industry initiative
66. Glikson A, Nastic S, Dustdar S (2017) Deviceless edge computing: extending serverless computing to the edge of the network. In: Proceedings of the 10th ACM international systems and storage conference. ACM, New York, p 28
67. Gao Y, Hu W, Ha K, Amos B, Pillai P, Satyanarayanan M (2015) Are cloudlets necessary? School of Computer Science, Carnegie Mellon University, Pittsburgh Technical Report CMU-CS-15-139

68. Datta SK, Bonnet C, Haerri J (2015) Fog computing architecture to enable consumer centric internet of things services. In: Consumer electronics (ISCE), 2015 IEEE international symposium on, IEEE, pp 1–2
69. Chen S, Zhang T, Shi W (2017) Fog computing. IEEE Internet Comput 21(2):4–6
70. Bonomi F, Milito R, Natarajan P, Zhu J (2014) Fog computing: a platform for internet of things and analytics. In: Big data and internet of things: a roadmap for smart environments. Springer, Berlin, pp 169–186
71. Dastjerdi AV, Buyya R (2016) Fog computing: helping the internet of things realize its potential. Computer 49(8):112–116
72. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39
73. Wan J, Xia M (2017) Cloud-assisted cyber-physical systems for the implementation of industry 4.0. Mobile Netw Appl 22:1–2
74. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing. ACM, New York, pp 13–16
75. Wen Z, Yang R, Garraghan P, Lin T, Xu J, Rovatsos M (2017) Fog orchestration for internet of things services. IEEE Internet Comput 21(2):16–24
76. Yannuzzi M, van Lingen F, Jain A, Parellada OL, Flores MM, Carrera D, Olive A (2017) A new era for cities with fog computing. IEEE Internet Comput 21(2):54–67
77. Satyanarayanan M, Simoens P, Xiao Y, Pillai P, Chen Z, Ha K, Hu W, Amos B (2015) Edge analytics in the internet of things. IEEE Pervasive Comput 14(2):24–31
78. Mahmud R, Buyya R (2016) Fog computing: a taxonomy, survey and future directions. arXiv preprint arXiv:1611.05539
79. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE communications surveys & tutorials
80. Sorensen A, Gardner H (2010) Programming with time: cyber-physical programming with impromptu. In: ACM Sigplan notices (Vol. 45, No. 10, pp 822–834). ACM
81. Raman V, Donzé A, Maasoumy M, Murray RM, Sangiovanni-Vincentelli A, Seshia SA (2014) Model predictive control with signal temporal logic specifications. In: Decision and control (CDC), 2014 IEEE 53rd annual conference on, IEEE, pp 81–87
82. Wang H, Zhou X, Dong Y, Tang L (2009) Modeling timing behavior for cyber-physical systems. In: Computational intelligence and software engineering, 2009. CiSE 2009. International conference on, IEEE, pp 1–4
83. Tan F, Wang Y, Wang Q, Bu L, Zheng R, Suri N (2013). Guaranteeing proper-temporal-embedding safety rules in wireless cps: a hybrid formal modeling approach. In: Dependable systems and networks (DSN), 2013 43rd Annual IEEE/IFIP International conference on, IEEE, pp 1–12
84. Wan J, Yan H, Suo H, Li F (2011) Advances in cyber-physical systems research. KSII Transac Internet Info Syst (TIIS) 5(11):1891–1908
85. Saha I, Roy S, Ramesh S (2016) Formal verification of fault-tolerant startup algorithms for time-triggered architectures: a survey. Proc IEEE 104(5):904–922
86. Crenshaw TL, Gunter E, Robinson CL, Sha L, Kumar PR (2007) The simplex reference model: limiting fault-propagation due to unreliable components in cyber-physical system architectures. In: Real-time systems symposium, 2007. RTSS 2007. 28th IEEE international, IEEE, pp 400–412
87. Abdelwahed S, Kandasamy N, Gokhale A (2007) High confidence software for cyber-physical systems. In: Proceedings of the 2007 workshop on automating service quality: held at the international conference on automated software engineering (ASE). ACM, New York, pp 1–3
88. Wang X, Hovakimyan N, Sha L (2013) L1Simplex: fault-tolerant control of cyber-physical systems. In: Proceedings of the ACM/IEEE 4th international conference on cyber-physical systems. ACM, New York, pp 41–50
89. Zhang Y, Xie F, Dong Y, Zhou X, Ma C (2013) Cyber/physical co-verification for developing reliable cyber-physical systems. In: Computer software and applications conference (COMPSAC), 2013 IEEE 37th Annual, IEEE, pp 539–548

90. Zhang S, Vittal V (2014) Wide-area control resiliency using redundant communication paths. IEEE Trans Power Syst 29(5):2189–2199
91. Satyanarayanan M, Schuster R, Ebling M, Fettweis G, Flinck H, Joshi K, Sabnani K (2015) An open ecosystem for mobile-cloud convergence. IEEE Commun Mag 53(3):63–70
92. Schoitsch E (ed) (2010) Computer safety, reliability, and security. In: 29th international conference, SAFECOMP 2010, Vienna, Austria, Sept 14–17 2010, Proceedings (Vol. 6351). Springer
93. Kang W, Kapitanova K, Son SH (2012) RDDS: a real-time data distribution service for cyber-physical systems. IEEE Transac Ind Info 8(2):393–405
94. Martin A, Fetzer C, Brito A (2011) Active replication at (almost) no cost. In: Reliable distributed systems (SRDS), 2011 30th IEEE symposium on, IEEE, pp 21–30
95. Shen M, Kshemkalyani AD, Hsu TY (2015). Causal consistency for geo-replicated cloud storage under partial replication. In: Parallel and distributed processing symposium workshop (IPDPSW), 2015 IEEE international, IEEE, pp 509–518
96. Mittal D, Agarwal N (2015) A review paper on fault tolerance in cloud computing. In: Computing for sustainable global development (INDIACom), 2015 2nd International conference on, IEEE, pp 31–34
97. Yusuf II, Thomas IE, Spichkova M, Androulakis S, Meyer GR, Drumm DW, Schmidt HW (2015) Chiminey: reliable computing and data management platform in the cloud. In: Proceedings of the 37th international conference on software engineering-Vol. 2, IEEE Press, pp 677–680
98. Madsen H, Burtschy B, Albeanu G, Popentiu-Vladicescu FL (2013) Reliability in the utility computing era: towards reliable fog computing. In: Systems, signals and image processing (IWSSIP), 2013 20th international conference on, IEEE, pp 43–46
99. Elmroth E, Leitner P, Schulte S, Venugopal S (2017) Connecting fog and cloud computing. IEEE Cloud Comput 4(2):22–25
100. Wei Y (2014) Auto-configurable, reliable, and fault-tolerant cloud storage with dynamic parameterization. In: Services (SERVICES), 2014 IEEE world congress on, IEEE, pp 295–300
101. Gan H, Chen L (2014) An efficient data integrity verification and fault-tolerant scheme. In: Communication systems and network technologies (CSNT), 2014 fourth international conference on, IEEE, pp 1157–1160
102. Skobelev P, Trentesaux D (2017) Disruptions are the norm: cyber-physical multi-agent systems for autonomous real-time resource management. In: Service orientation in holonic and multi-agent manufacturing. Springer, Cham, pp 287–294
103. Jennings B, Stadler R (2015) Resource management in clouds: survey and research challenges. J Netw Syst Manag 23(3):567–619
104. Taylor B, Abe Y, Dey A, Satyanarayanan M, Siewiorek D, Smailagic A (2015) Virtual machines for remote computing: measuring the user experience
105. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4)
106. Ha K, Abe Y, Chen Z, Hu W, Amos B, Pillai P, Satyanarayanan M (2015) Adaptive vm handoff across cloudlets. Technical report, Technical Report CMU-C S-15–113, CMU School of Computer Science

# Part II
# Security and Privacy

# Chapter 6
# Survey on Access Control Models Feasible in Cyber-Physical Systems


**Mikel Uriarte, Jasone Astorga, Eduardo Jacob, Maider Huarte, and Oscar López**

**Abstract** Security is a key aspect in the development of innovative and valuable services based on Cyber-Physical Systems (CPSs). In the last years, the research area related to CPS security has received a significant attention, dealing with the design of different architectures, security protocols, and policy models. However, beyond monitoring data publishing behavior, CPSs are expected to offer some manageability-related services, and the proper fine-grained and flexible access control model remains challenging due to both criticality and feasibility. In fact, traditional security countermeasures cannot be applied directly to any sensor in CPS scenarios, because they are too resource-consuming and not optimized for resource-deprived devices. Different access control models facing both feasibility and enforcement tightness are arising as a way to solve the mentioned issues related to resource limitations, and this study provides a deep survey on them.

## 6.1 Introduction

This chapter conveys an overview of current security solutions, concretely access control solutions, on Cyber-Physical Systems (CPSs) implemented in constrained devices, and accessible as things in an IP network (IoT). In fact, two behaviors are distinguished within the CPSs integrated in IoT context. On the one hand, the usual behavior of a CPS involves a push operation in which the CPS publishes measurements and events to a few large message brokers. Therefore, CPSs behave as information producers. In the user side, applications consuming measured data

M. Uriarte (✉) · O. López
Nextel S.A., P.T. Bizkaia, Zamudio, Bizkaia, Spain
e-mail: muriarte@nextel.es; olopez@nextel.es

J. Astorga · E. Jacob · M. Huarte
Department of Communications Engineering, University of the Basque Country UPV/EHU, Bilbo, Bizkaia, Spain
e-mail: jasone.astorga@ehu.eus; eduardo.jacob@ehu.eus; maider.huarte@ehus.eus



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_6

query the message brokers in the middleware rather than the CPSs deployed in field directly. Due to this middleware mediation, the number of communicating peers that are architected in some few layers is reduced and well known beforehand, so the security requirements of the push mode operations of the CPSs can be easily fixed by preconfiguring static security associations. Alternatively, other more sophisticated IoT scenarios are also envisioned in which CPSs also act as tiny information servers enabling smarter and more manageable applications. In such use-cases, requesting subjects directly query the CPSs through a secure end-to-end (E2E) communication. Consequently, when any subject tries to interact with a CPS, such E2E interaction between the endpoints needs to be secured, as well as the establishment of the necessary keys for securing the interaction.

These IP networks where CPSs are deployed are exposed to a set of threats and vulnerabilities that need to be tackled. Most remarkable ones could be cloning of devices, malicious substitution, eavesdropping, man-in-the-middle (MITM), firmware replacement, security parameters extraction, routing attack, privacy threat, and Denial-of-Service (DoS) among others. Some of them such as cloning of devices or malicious substitution are related to manufacturing process and physical exposition, so they are left out of scope of this chapter. Some others such as routing attack and DoS are related to the network level and they are also left aside. In this chapter, particular attention is paid to eavesdropping, and MITM, and the related firmware replacement (if it would be over the air), security parameters extraction, and privacy threat. These five threats share the need to protect information as well as the keying data and the security parameters, by means of their integrity and the authenticity and confidentiality of the communication channel. Additionally, they lead to the need of a tight authorization policy enforcement.

The security solutions for such security requirements are defined as a set of features, which are confidentiality, authentication, integrity, authorization, non-repudiation, and availability. Additionally, some other features are duplicate detection and detection of stale packets. These security features are usually implemented, on the one hand, by a combination of cryptographic mechanisms that present a high computational cost, such as block ciphers, hash functions, or signature algorithms, which require a proper handling of cryptographic keys. And, on the other hand, noncryptographic mechanisms relying on previous ones, which implement authentication, authorization, and other security policy enforcement aspects. So security policy has to be adequately codified and an enforcement engine is required, which might be also ready to accept policy changes during the life cycle of the device.

It is generally accepted that authentication mechanisms can achieve non-repudiation and accountability security objectives. Additionally, authorization mechanisms contribute to achieve confidentiality and integrity security objectives. In fact, allowing only selected entities limits the burden on system resources, thus helping to achieve availability. Moreover, frequently, authorization mechanisms rely on authenticated attributes, and both mechanisms work together. Finally, authorization addresses better the least privilege principle [1] and flexibility as the granularity of the policy increases.

In the context of the constrained devices required in large-scale deployments, the security must not only focus on the required security services, but also on how

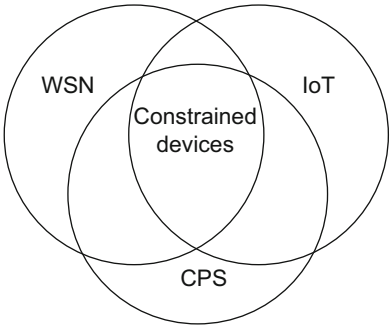



**Fig. 6.1** CPSs integrating sensing capabilities in IoT

these services are realized in the overall system and how the security functionalities are executed. In fact, this chapter covers an overview of the application security solutions in CPSs acting as things that will be recognizable, addressable, accessible, locatable, readable, and controllable over the Internet. And more specifically, it focuses on the access control models and related architectures. How these access control models protect the confidentiality and the integrity of the data exchanged with constrained devices, as well as the authentication and authorization enforcement of any endpoint accessing data in the constrained device are analyzed, under two main criteria: feasibility and least privilege principle adherence.

So hereinafter, firstly, the features and limitations related to the considered constrained devices scenario are conveyed. In such a constrained scenario, shown in Fig. 6.1, where Wireless Sensor Networks (WSN), IoT, and CPS converge in the range of constrained devices, it is relevant to consider how these limitations impact on the feasibility of any access control model. Secondly, an overview of existing policy languages is conveyed pointing out the contribution of the main features to the expressiveness and consequently to the tightness of the enforcement and adherence to the least privilege principle. In fact, the latter is considered as the main contributor to the effectiveness, whereas the former set the constraints to be faced from the efficiency point of view. Finally, currently, there are some access control models tailored for the mentioned scenario and they are analyzed under the aforementioned criteria of the enforcement tightness and feasibility.

Concretely, eight different access control models adapted to IoT are conveyed in this chapter. First, the so broadly adopted XACML has been adapted to IoT as a centralized Attribute-Based Access Control (ABAC) implementation. Alternatively, Usage-Based Access Control (UCON) adapted to IoT is conveyed, which proposes a distributed access control enforcement model not only before the first access but also during it. And the third most differentiated approach is the capability-based access control with two proposals, fourth and fifth, which convey progressive distribution of the authorization decision and enforcement based on capabilities distributed based on a Public Key Cryptography (PKC) schema. The sixth approach proposes a token-based access control that can be seen similar to previous ones but fully compliant with constrained environment standards such as Constrained Application Protocol (CoAP) and Datagram Transport Layer Security (DTLS). The seventh proposal named Ladon, which is absolutely optimized and feasible

in severely constrained scenarios, proposes a Role-Based Access Control (RBAC) authorization enforcement based on a Symmetric Key Cryptography (SKC) schema. Finally, the eighth approach named Hidra, which is feasible in severely constrained devices, proposes a local context-based access control model enabling a tight policy enforcement, through a specific purpose policy language, as well as the dynamic policy provisioning and accounting mechanisms for further tracking and auditing.

The analysis of such eight innovative lightened approaches might conclude to what degree these models do support enhanced fine-grained and tight security policy enforcement in severely constrained devices, since the currently implemented static and coarsely grained policies to be enforced locally in the CPS are not applicable for service-oriented open scenarios where operation and management access is by nature dynamic and ad hoc.

## 6.2 Context-Related Features and Requirements

Internet of Things integrating CPSs is a more demanding environment in terms of scalability and manageability as compared to traditional Internet services [2, 3]. In fact, substantial changes are identified in

- Interaction patterns: short-lived, often causal and spontaneous interactions are different from that of traditional systems.
- Context relevance: requests, data, or authorization might depend on the local context.

Therefore, access control solutions for CPSs integrated in IoT, under least privilege principle adherence criteria, need to face not only the tightness challenge, which is related to the granularity and the expressiveness of the security policy, but also the local context awareness, the scalability, the support of advanced features such as delegation and auditability, the easiness to manage, and the flexibility to adapt to different contexts and applications. When things are deployed on a massive scale, they need to be cheap and, therefore, constrained devices. This fact adds the feasibility challenge to the already stated set of challenges. Moreover, the limits imposed by the devices on computation, memory, network bandwidth, and power require to optimize the energy and network usage in all design requirements in order to be just feasible.

### *6.2.1 Constrained Device Classification*

Sensors integrated in CPSs can be implemented in constrained devices with strict resource constraints such as limited computing capacity, little memory, insufficient network bandwidth, and often limited battery power. Depending on the dimension of such resources, diverse sizes of constrained devices can be distinguished, ranging from camera devices to the smallest networked sensor interacting with other *things*

nearby. Concretely, the range of constrained devices is defined by the IETF. Class 0 (C0) is the lowest level, where devices have less than 10kB and 100kB of data and code memory, respectively. From this lowest level, Class 1 (C1) devices are about 10kB of data and 100kB of code, and class 2 (C2) devices are up to 50kB data and 250kB code memory. These devices are specifically implemented to fit to the requirements of different use-cases and applications. Besides, Moore's law [4] is foreseen to impact more on the price than on the resource capabilities [5, 6]. With respect to available power, mains-powered devices are notably distinguished from ones powered by batteries or by using energy harvesting.

- C0 devices generally cannot be managed or secured in the traditional sense. They can offer some specific tiny services through the network that require high optimization in order to be feasible, and the same happens with supported security functions. Samples of C0 devices are networked sensors and actuators with specific purpose and powered with batteries in massive deployments such as urban monitoring and light switching.
- C1 devices are capable enough to use lightweight protocols such as CoAP over UDP. Therefore, they can act as fully developed peers into an IP network supporting also some more general security functions. Samples of C1 devices are networked sensors and actuators like fire/smoke detectors integrated in industrial control and large buildings, able to support some functions needed for its intended operation and management.
- C2 devices are considered less constrained devices and they support most of the same protocol stacks as used in mobile devices such as smartphones. Samples of C2 devices are networked sensors and actuators integrated in smart energy and building automation environments, able to support a range of services including some management ones.
- Devices with capabilities beyond C2, nearer from unconstrained devices, are less demanding from a standards development point of view as they can largely use existing protocols unchanged, denoted in this document as traditional security protocols. Their principal constraint could be related to the location and the availability of mains-power or the use of batteries, tight to the energy consumption optimization.

In all cases, depending on the use-case and the operational scenario, all these devices still need to be assessed for the type of application they will be running and the protocol functions they would need, and moreover from the manageability and security point of view.

### *6.2.2 Constrained Networks*

The network where constrained devices work is usually also a constrained network. This implies low bandwidth, high packet loss, penalties for fragmentation due to large packets, limits on reachability over time, and lack of advanced services such

as IP multicast. Such networks conveying a variety of wireless links such as the low data rate IEEE 802.15.4 [7] are also denoted by low-power and lossy networks (LLN) [8]. On top of such link layer technologies, the network and transport stack are composed by specifically adapted protocols such as CoAP [9, 10] and 6Low-PAN [11]. However, these protocols have been designed considering the lightweight principles but the security principles have not been properly adopted. In fact, CoAP is the definition of a lightweight version of the HTTP protocol, which runs over UDP and enables efficient application level for things. 6Low-PAN is the specification of methods and protocols for efficient transmission and adaptation of IPv6 packets over IEEE 802.15.4 networks.

Hence, the implementation of more ingenious and valuable applications needs to tackle the insufficient security [12–14], which according to Gartner is dissuading potential investors from large-scale deployments of IoT solutions [15]. In particular, research on security up to now has focused on network security involving key management, message authentication, intrusion detection, etc. [16, 17]. However, until recently low attention has been paid to fine-grained access control models [18].

In any case, constrained devices C0-C2 share the following limitations derived from resource scarcity:

- Complex authorization policies cannot be managed
- Large number of secure connections cannot be managed
- Deprived of user interface
- Deprived of constant network connectivity
- Time cannot be precisely measured
- High power consumption of the wireless communications
- Severely constrained storage space for security policies such as ACLs in massive deployments
- Required to save on cryptographic computations due to a high power consumption

### *6.2.3 Life Cycle and Access Control Requirements*

Some of the requirements for the access control models are better understood when considering that the life cycle of a constrained device consists of several phases. The device is created in the manufacturing phase and that is sometimes also the moment for the initial key provisioning. This key is usually the secret key shared with the central access control server (ACS). Devices are then sold to customers who introduce them to their networks during the commissioning phase. In this phase, the owner of the device might set up the pending initial key or just customize the ACS address and the related security policy.

In the following operation phase, constrained devices fulfill their purpose in life, sometimes alternated with a maintenance phase. In more and more scenarios, devices are required to offer a sort of tiny services. These services enable

both the end-user experience customization and the management of the device, which contribute to a lower operational cost, higher flexibility, and longer validity period. Concretely, they are expected to be accessible E2E avoiding the need for application-level proxies. Additionally, mechanisms for changing the security policy in any phase of the life cycle are needed, i.e., flexible and manageable security is also required.

Some devices change their owner during their lifetime and need to be decommissioned and recommissioned in the handover phase. This implies that initial key and ACS binding need to be replaced, and old ones need to be revoked. At the end of the device's life cycle, the device is decommissioned in the decommissioning phase.

### *6.2.4 Use-Case-Driven Access Control Model*

Concretely, depending on the case, static configuration setup during manufacturing or deployment might enforce proper authentication and authorization, based on fixed trust parties and access control lists. This is particularly applicable to fixed-purpose deployments or well-known set of peers, where the sensor acts as the producer and pushes some measures to a well-known message broker. However, envisioned open and flexible scenarios cannot anticipate the legitimate set of authorized entities and their privileges. That is, there is a need for change during the lifetime of the CPSs. Moreover, access control lists are not scalable in large-scale deployments, and tighter access control policies might be needed in several use-cases.

Furthermore, a local enforcement policy could be based only on authentication. Among other possibilities, it could add an additional authoritative check through access control lists, have hard-coded a particular role-based authorization policy, have an attribute-based policy configured, receive such policy dynamically for each request, or enforce the authorization not only at the request but also during the access driven by a usage-based policy, etc. Each of them provides a different tightness according to the least privilege principle. Each of them might be susceptible to changes, and the policy life cycle must be considered. Therefore, each of them enables a different degree of flexibility and means an impact on the performance, network overload, energy consumption, and feasibility.

### *6.2.5 Security Policy*

When designing and applying access control models, the tightness of the enforcement is related to the granularity of the policy, but there is always a trade-off between expressiveness of the policy and practical feasibility. Moreover, an access control model should cover not only a feasible security policy definition and enforcement, but also a way to enable the policy changes so needed in open and dynamic environments.

#### 6.2.5.1 Policy Language

On the one hand, the model must be able to formally express all required policies as precisely as possible. Choosing an access control model with a high expressiveness enables the application of the principle of least privilege. This principle states that at any point in time, subjects and users should be given the least set of privileges necessary to complete their job. An expressive model allows the specifications of policies that closely match the high-level security requirements and, therefore, are able to give subjects as few permissions as possible.

On the other hand, the model must be practically feasible. This means that it must be viable to implement the model in the concrete organizational context of the involved systems, moreover considering the scarcity of resources on heavily constrained devices. Furthermore, the practical feasibility of an access control model has several additional dimensions such as the manageability of the policy, the efficiency of enforcement, the level of formal guarantees, the ability to deal with changes in the system, the ease of delivering policy specifications, etc. to be considered as well.

The two aspects of expressiveness and practical feasibility are often opposite forces that need to be reconciled when making a design choice for a particular model. Some conventions might be adopted defining a short set of possible values in a policy, their semantic and their syntax, composing the policy domain model (PDM). The specific common understanding by E2E peers can be agreed upon as PDM specifications, which include also the trust relationships among the security architecture actors. In implementations where the PDM is specified in separate files, their modification, provisioning, and activation are much more agile.

#### 6.2.5.2 Policy Changes

Policy changes are typically caused by the modification of the involved applications or their security requirements [19]. Therefore, policy changes are adaptive changes although they can also be corrective. Policy changes can occur at the level of the implementation-level policy rules or at the level of the policy domain model.

Policies are often internally structured in sub-policies and/or rules. In general, policy rules consist of two elements named *condition* and *effect*. The *condition* determines the applicability of the rule, and the *effect* determines the policy decision that is to be enforced. The condition of a policy rule determines the set of authorization requests the rule is intended to govern. A rule condition is defined as a Boolean function over the attributes of the subject, action and resource relevant to a request.

Therefore, a change to the condition of a rule has the effect that the new rule will apply to a different set of requests. From the point of view of policy evaluation, the possible changes in the condition of a rule are four: (1) An internal change in the condition of the rule while using the same attributes and values, such as the modification of an inequality to an equality. This changes the outcome of the rule

but has no impact on the authorization decision-making process. (2) Introduce a dependency on a new kind of attribute or remove the dependency on a particular kind of attribute. (3) Introduce or remove a dependency on a specific attribute value (such as a particular role or department). (4) Any combination of the preceding cases.

In general, in a centralized deployment, it is easier to implement policy changes in a unified policy repository. Instead, in a decentralized deployment, any change requires the updating of the distributed policy instances that involves network interoperability and overload, and this could lead to temporally inconsistent states or alternative pseudo-static coarse-grained configurations.

### *6.2.6 Security Architecture Overview*

All access control models rely on a security architecture where different entities might collaborate in the final E2E secure session establishment. Both centralized and distributed approaches have been broadly analyzed, and there is a general acceptance of the advantages and drawbacks of each one. In any case, it is interesting to point out the main actors and their role in the minimal core access control architecture, in order to better analyze the security protocols that enable the establishment of the E2E security association to meet the security objectives.

#### 6.2.6.1 Inherent Advantages and Drawbacks Related to the Security Architecture

In current IoT implementations, the most common security architectures are fully centralized, where the authorization decision is made and all the security-related message exchanges are handled by a central entity. This trusted central entity has no constraints by means of resources. So, a centralized architecture allows for a central management of devices, security policies, and keying materials as well as for the backup of cryptographic keys. However, it presents some drawbacks since the central party represents a mandatory point of access, and the key agreement between two devices requires online connectivity to the central node. Additionally, authorization decision does not consider the local context of the CPS. The central entity reads the requests and can compromise the privacy of the requester. Finally, trust among entities needs also to be managed, and any security breach might compromise a vast amount of security information.

In decentralized architectures, on the other hand, all the access control logic is embedded in the CPS, and the authorization decision is made locally enabling local context consideration. They also allow setting security relationships on the fly between endpoints, which might not require a single online management entity and are operative in a much more stand-alone manner. However, due to the limited storage capacity of the constrained devices, they cannot manage large security

policies nor handle large sets of credentials, so the fully decentralized architecture is not feasible and it requires the support of a trusted third party to complement security-related functions either in the bootstrapping or operational phases.

The advantages and drawbacks of the architecture condition the way the access control requirements in IoT scenarios are covered. In fact, centralized access control approaches might support the full feature set of standard security protocols, but constrained CPS in IoT require more lightweight mechanisms to avoid computation and network overload. Furthermore, in the envisioned IoT scenarios the distributed access control approaches support higher scalability, interoperability, and local context awareness.

#### 6.2.6.2 Access Control Core Architecture Elements

In the access control scenario, there are three elemental actors: a *subject* (1) that attempts to access a service considered a *resource* (2) in a CPS. This subject needs the collaboration of a trusted third party named *access control server* (ACS) (3), in the establishment of an E2E security association.

When a subject attempts to access a resource in a CPS, an E2E access control is required, but due to CPS constraints, not all the required mechanisms are feasible; therefore, the three-tier architecture is adopted. Figure 6.2 shows a simplified schema of the actors involved, aligned with the IoT reference stack [20].

The *subject* endpoint can be either a constrained endpoint, such as a less-trusted mobile endpoint, or a less constrained endpoint located in the related application data-center.

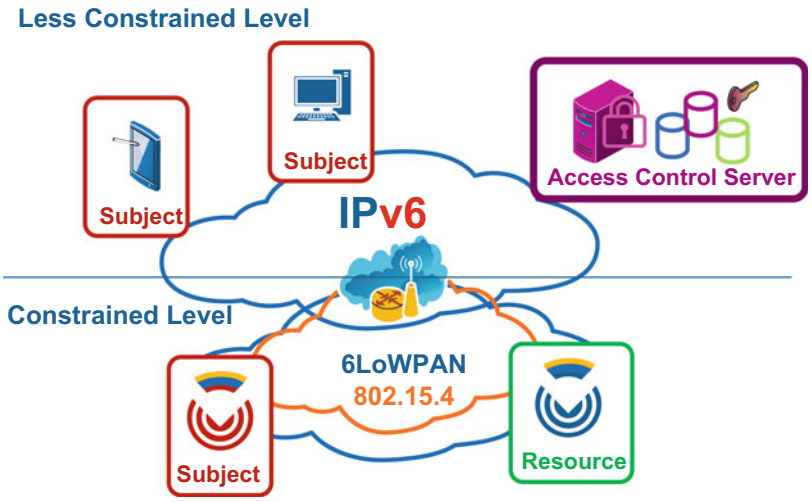



**Fig. 6.2** Access control scenario in constrained CPS networks

In this scenario, a *resource* endpoint is implemented in a CPS and located wherever the sensor is deployed. Resources typically have corresponding interfaces through which subjects interact with them over the IP network.

The trusted third party, that is, the ACS, might support most of the features needed for a complete access control chain such as registration, identification, authentication, authorization, accounting, tracking, auditing and reporting. All these functionalities are considered concentrated in a third party entity but they could work in a cluster or simply be replicated because they belong to different federated domains. Specifically, most resource-intensive features can rely on the ACS, which may perform security functions of the full authorization chain such as identity registering and management, strong authentication, primary authorization, policy administration and deployment, accounting, etc. This architecture enables the integration of standard access control mechanisms that otherwise could not be supported by the severely constrained resources of CPSs. This way, most unauthorized access attempts are denied at a preliminary step. This feature contributes to saving energy by means of minimizing the number of unsuccessful message exchanges with the CPSs. Additionally, unified policy management is also enabled by this approach. Finally, in the case of positive primary authorization, the CPS locally enforces the security policy that can be implemented in different ways.

Whatever is the message flow among the three parties in this architecture, all of them require and check mutual authentication based on any cryptographic schema.

### *6.2.7 Cryptographic Schema and Key Establishment*

The establishment of the necessary keys can be based on optimized PKC schemas, or based on intrinsically lighter SKC schemas assisted by the third party mentioned above. In fact, keys are required both for protection of resource access and for protection of transport of authentication and authorization information. There are classes of devices that can easily perform symmetric cryptography without incurring in excessive performance penalty but consume considerably more time/battery for asymmetric operations so they are severely limited in this sense. Also PKC requires the three-party schema but not so active in runtime if initial private and public keys are established and provisioned out of band. Alternatively, preestablished master keying material may need to be employed for establishing the keys used to protect the information flows among three parties. Constrained devices have only limited storage space and thus cannot store large numbers of preestablished keys or session keys. Asymmetric cryptography has benefits in terms of deployment. This is especially important and affects the scalability when, in addition, constrained networks are expected to consist of thousands of nodes.

Most of security protocols pay attention to the reduction of the computation and communication overhead of cryptographic operations required for key exchanges and signatures with Elliptic Curve Cryptography (ECC) [21]. In fact, most protocols enable the negotiation of cryptographic primitives aiming at higher agility.

Unfortunately, these improvements are only a first step in reducing the impact on performance, and the feasibility in heavily constrained devices remains unsolved. In fact, the use of a standardized symmetric key algorithm, such as AES, is most broadly adopted. Additionally, except for the most constrained devices C0, the use of a standardized cryptographic hash function such as SHA-256 is the most adopted.

End-to-end security is of great importance in individual communications within an IoT domain. Usually and E2E security association is established and the use of intermediate proxies or gateways is avoided. That is, when payloads are cryptographically processed, packets might be encrypted and message authentication codes might be conveyed, so the protected parts cannot be accessed or rewritten by any gateway unless E2E integrity protection is violated. Alternatives such as sharing symmetric keys with intermediate gateways result either simply unacceptable by endpoints or poor performance in the relaxed cases.

## 6.3 Access Control Foundations

The increasing smartness of CPSs enables more valuable IoT applications but security in general, and access control in particular, needs to be adapted to such constrained scenarios. In fact, tight enforcement of fine-grained access control policies is a critical success factor that otherwise is not feasible through traditional solutions. That is, security mechanisms implemented in powerful workstations are not technically viable in CPSs with severe resource constraints. Although the limitation of the capabilities on the CPSs, traditional access control models are analyzed as inspirational references.

### *6.3.1 Policy-Driven Security Management*

Instead of security rules coupled within the applications' logic, policy-driven security management has become the de facto approach for security management in large-scale systems [22]. In fact, CPSs integrated in IoT are rising in scale and incorporating innovative heterogeneous technologies. This fact leads to a complexity that cannot already be overcome with traditional security management strategies, which basically rely on effort-consuming and error-prone manual work.

To resolve these issues, policy-driven security management is proposed to be leveraged, to simplify the administration of the large-scale systems. In a policy-driven management system, security policies specify the conditions according to which a resource can be accessed. That is, a security policy is an intermediate format to map security requirements to specific and feasible operations on resources.

The policy-driven approach has three main advantages over traditional security management methods. Firstly, policies are defined by policy operators through a well-known policy language, and they are stored in a common repository. Such

policies can be later retrieved autonomously. Secondly, the formal foundation of most policy languages introduces automated analysis and verification of policies, with the purpose of ensuring consistency. Finally, because of the abstraction from lower technical details, policies in the policy-driven approach can be inspected and changed dynamically at runtime, without changing the underlying implementation.

#### 6.3.1.1 Policy-Driven Architecture

The Policy Core Information Model (PCIM) [23] describes the components in a policy-driven management system: (1) the policy repository (PR), (2) the policy decision point (PDP), (3) the policy enforcement point (PEP), and an optional fourth component acting as the policy administration point (PAP), which facilitates the formulation, analysis, and verification of security policies.

Both industry and academia are researching policy-driven management systems. In fact, there are several commercial tools based on the PCIM framework in the industry area. Additionally, many policy languages to model access control entities have been proposed by researchers in the academic area.

#### 6.3.1.2 Security Policy Foundation

An abstract policy is implemented through the concrete rules that construct it. Rules in turn, in most policy languages, are based on the following paradigms: Event-Condition-Action (ECA) paradigm and Condition-Action (CA) paradigm. The ECA paradigm [24] differs from the CA paradigm in that it needs an explicit *event* element to trigger the execution of an *action* under certain *conditions*: "ON (Event) IF (Condition) THEN (Action)." The event in most cases is related to a request from a subject but it covers also the periodical trigger so useful in usage-based access control. The condition is usually a function of several checks that can range from simple attribute value matching to expressive functions on attributes as inputs. The action refers to a simple action or a sequence of them.

### *6.3.2 Access Control Models*

There are some foundational models to enforce that subjects can only access the resources they are authorized in a CPS. Concretely, Mandatory Access Control (MAC) [25] and Discretionary Access Control (DAC) [26] are two of the universal conceptual access control models in use. In MAC, administrators create a set of access levels and subjects are entitled with an access level, so that they can access any resource on a CPS labeled equal or below such entitled access level. Alternatively, DAC defines a list of authorized subjects for each resource, so access granting is based on the identity of the subject instead of an assigned access

level. DAC is more flexible but it requires knowing each subject who needs the resource so that they can be given access. This approach scales notably bad in large-scale deployments. Consequently, the RBAC model [27] faces the easiness of management of big communities of subjects, defining much shorter set roles, centrally administered, and granting each subject the correct role. However, it also presents some drawbacks related to the management in open scenarios, where role explosion is the most significant one.

#### 6.3.2.1 Attribute-Based Access Control (ABAC) Beyond Role-Based Access Control (RBAC)

The ABAC model [28, 29], which can be considered as a generalization of RBAC, has become the de facto standard access control model in service-oriented systems. In ABAC, policies are specified in terms of attributes of subjects, resources and the environment. Attributes are properties that describe arbitrary characteristics of an entity, such as the age of a user, the author of a document or the threat level of the environment. Through the use of custom attributes, ABAC supports rich context information as well the categorization of subjects and resources.

As a theoretical model, ABAC is well suited for service-oriented systems since it has inherent support for (1) expressive policies (rich context information can be modeled as attributes), (2) changes (many changes can be supported by a modification of an attribute value rather than a policy update), and (3) administrative scalability (attributes categorize subjects and resources).

#### 6.3.2.2 Usage-Based Access Control: UCON

As an alternative, there is an access control model covering not only authorizations, obligations, and conditions, but also continuity (ongoing controls) and mutability. UCON [30, 31] is an extension of access control that covers both who may access and how the data can be used. Usage control is suitable in distributed systems with different entities that take the roles of data providers and data consumers. When a data provider gives a data item to a data consumer, certain conditions apply. *Provisions* are those conditions that refer to the past. Otherwise, *obligations* are other conditions that govern the future usage of the data. Examples of obligations could be (1) do not further distribute document *D*, (2) play a song at most 3 times, and/or (3) warn the author whenever document *D* is modified. Moreover, mutability issues that deal with updates on related subject or object attributes as a consequence of access are also proposed.

The related *UCONABC* (*Authorization, oBligations, Conditions*) model aims at the refinement of the traditional access control scope, supporting MAC, DAC, and RBAC as well as digital rights management, paving the path to next-generation access control systems.

However, UCON does not properly overcome the challenges of IoT security requirements [33]: It still lacks an exact representation and a detailed specification of the authorization process in IoT, and it is seen as a conceptual model that needs much more research to be implemented.

## 6.4 Access Control Policy Languages

The aspect that has most impact on the least privilege adherence is the granularity of the access control policy. Coarse-grained policies can easily derive in misuse or abuse. Otherwise fine-grained control is achieved through expressive policy languages facing the aspects aforementioned in the access control foundations. Herein a short overview on some of the most broadly adopted policy languages is presented.

### *6.4.1 XACML*

Nowadays, eXtensible Access Control Markup Language (XACML) [32] is widely accepted both in industry and academia as a de facto standard. XACML is a declarative, XML-based policy language, for expressing ABAC policies, mainly for access control management in distributed systems. The XACML standard, standardized by the OASIS consortium, specifies the syntax and semantics of the policy language and defines a request–response format for querying a policy system. XACML defines also a modular architecture where the authorization decision and enforcement are decoupled among other functions such as policy definition and data gathering for authorization decision.

The atomic entity in a policy is the rule as shown in Fig. 6.3. A rule consists of a target, an effect, and a condition. The target specifies the applicability of the rule, expressed in terms of the action and attributes of the subject, resource and environment related to the authorization request. The effect states the policy decision of the rule and can have the values *Permit* and *Deny*. The condition specifies constraints (also in terms of attributes) to further limit the applicability of the rule.

A policy is a grouping of related rules and consists of a target, a rule-combining algorithm, obligations, and the set of contained rules. Similar to the target of a rule, the target of a policy specifies the applicability of the policy. It can be specified explicitly or it can be computed automatically based on the targets of the contained rules. The rule-combining algorithm specifies how to combine the evaluation of the individual rules into a single decision for the policy as a whole (for instance, the deny-overrides algorithm states that if any rule evaluates to Deny, the result of the combination must also be Deny). Obligations are tasks that should be performed by the system or the requesting subject in addition to enforcing the decision.

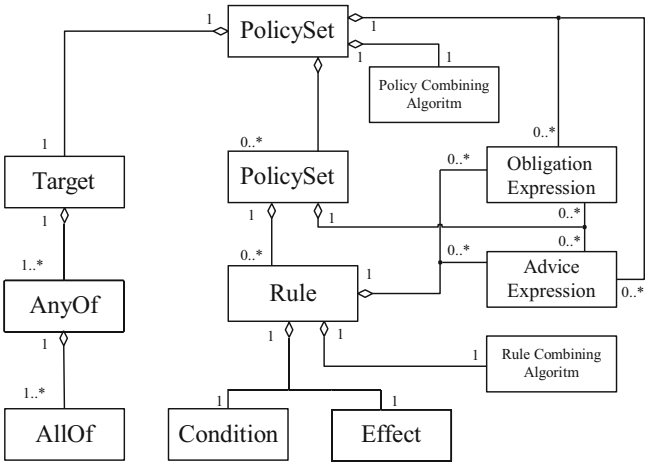



**Fig. 6.3** XACML policy language model 3.0 [32]

Policies are grouped into policy sets. A policy set consists of a target, a policy-combining algorithm, a set of obligations, and the set of contained policies. Similar to policies, the target specifies the applicability, the policy-combining algorithm specifies how the decisions of the contained policies must be combined into a single decision, and the obligations specify obliged actions.

A policy set can contain other policy sets, thus allowing the construction of tree-structured policies of arbitrary depth. Although it is one of the most broadly adopted standards in access control, since XACML is based on XML, policy specifications are very verbose. For instance, the specification of one single policy rule can easily require 50 lines of text, and it implies that is infeasible in most constrained devices. However, it is a remarkable reference for any lightweight but expressive policy language definition.

XACML specification defines the structure of some of the messages necessary to implement the model, but it focuses on the language elements used by the PDP and it does not specify any protocol or transport mechanism.

One standard suitable for providing the assertion and protocol mechanisms needed by XACML is the OASIS Security Assertion Markup Language (SAML), Version 2.0 [34]. With the rise in popularity of APIs and its consumerization, it becomes important for XACML to be easily understood in order to increase the likelihood it will be adopted. In particular, XML is too verbose compared with a lighter representation using JavaScript Object Notation (JSON) for the XACML request and response.

The policy language defined by XACML standard is fully expressive and enables the least privilege principle adherence. However, it specifies neither the required mutual authentication schema nor any key exchange protocol, and it relies on other

standards such as SAML [34], OAUTH [35], etc. It is not feasible in constrained devices but it is a mandatory reference for any lightweight but expressive policy language definition.

### 6.4.2 *Ponder Policy Language*

Ponder [36] is a declarative and object-oriented policy language. Security policies are specified using RBAC, and general management policies for distributed systems can also be specified. In Ponder, related policies are grouped into roles as shown in the structure depicted in Fig. 6.4. The relationships define the interactions between roles. An advantage of Ponder is the reuse and flexibility of policies enabled by such structure.

Ponder is considered to have a broad scope in a variety of policies since it defines five types of policies: (1) authorization policies, (2) filter policies, (3) refrain policies, (4) delegation policies, and (5) obligation policies. The former four types are used to define conditions to be checked in order to grant access to

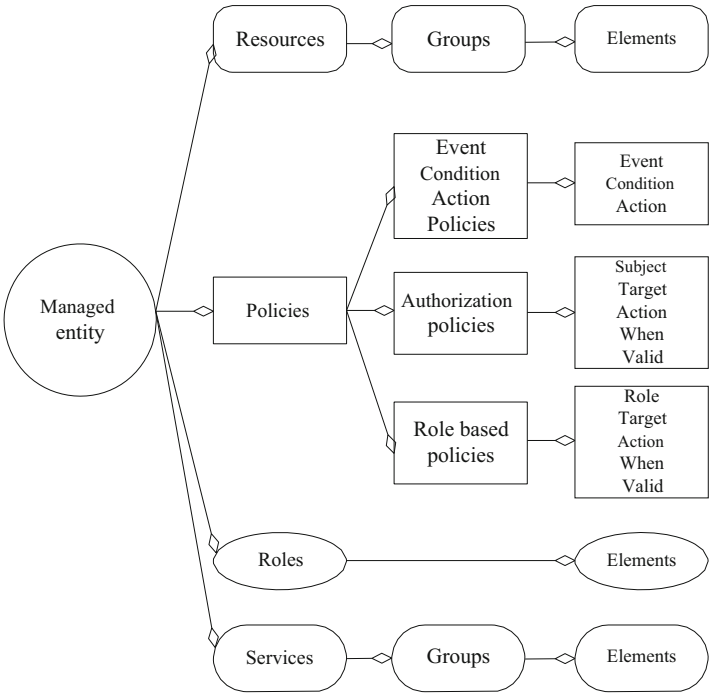



**Fig. 6.4** Ponder policy framework

resources, whereas obligation policies are used to define reactive actions under the event-condition-action paradigm. That is, when specified events and conditions are met, the specified action must be performed on target objects.

### 6.4.3 *Rei Policy Language*

Rei [37] is a declarative policy specification language based on deontic logic. It is concerned with obligations, permission, etc. as shown in the N3 [38] representation in Fig. 6.5, and it is oriented to security and privacy enforcement in dynamic and open computing scenarios. Rei defines the policies as specific constraints over authorized and obligated actions on resources. Rei implementations provide a policy mechanism (engine) to make dynamically an authorization decision on a request from subject, after reasoning over policy instances, correspondent domain knowledge, and some meta-policy instances that are used to ensure that the decision is consistent and conflict-free. In addition, Rei is not strictly RBAC, since it can define individual, grouped, and role-based policies at the same time. Moreover, it supports so-called speech acts which include delegation, revocation, request, and cancellation. These acts enable a simple and decentralized access control of pervasive applications.

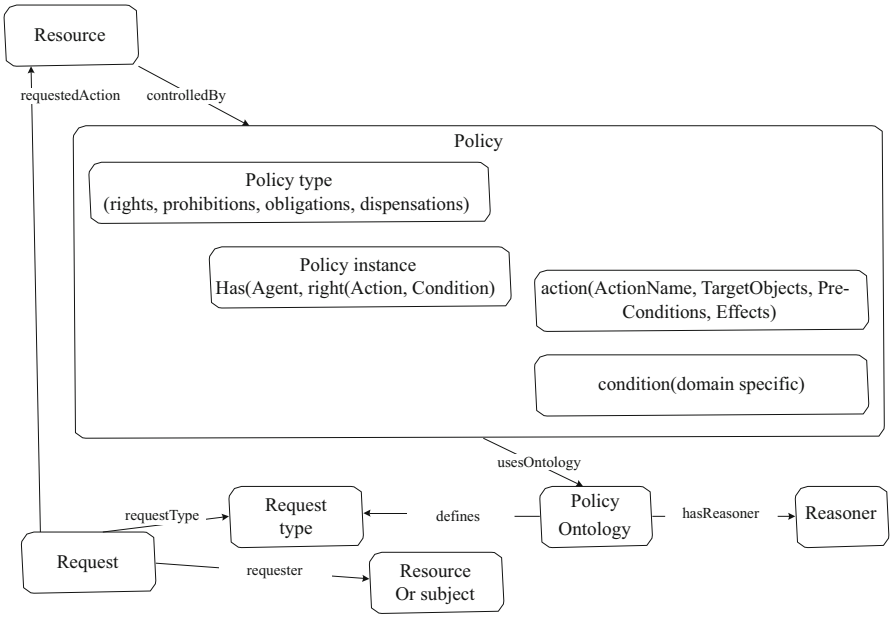



**Fig. 6.5** Rei policy schema in N3 notation

### 6.4.4 Authorization Specification Language (ASL)

Authorization Specification Language (ASL) [39] is an access control language based on first-order logic. ASL conveys essential RBAC components, i.e., users, roles, and objects. ASL defines a rule in an authorization policy instance as a mapping from the quadruple *user*, *role set*, *object*, and *action* to the access decision *authorized*, or *denied*. From this view, the decision-making mechanism of ASL lacks sufficient flexibility and re-usability, because the authorization decision is encoded in the rule itself, and it is not dynamically made by a PDP.

ASL policy instances are written in a Datalog program that is a formal language based on first-order logic. The policy instances are composed of a set of (1) authoritative rules, (2) derivation rules, (3) resolution rules, (4) access control rules, and (5) integrity rules. When an access request arises, these rules are evaluated according to their semantics. Additionally, ASL supports the specification of rules to tackle with authorization derivation and conflict resolution issues.

### 6.4.5 Obligation Specification Language (OSL)

Obligation Specification Language (OSL) [30] is a language for expressing requirements of usage control. It enables constraints on the duration of usage and the kinds of permission-like statements that are often used in digital rights management. A formal semantics for OSL is also defined. This language builds a framework that provides tools for specifying, reasoning about, and enforcing usage control requirements. Moreover, translations between OSL and some rights expression language (REL) have been defined, which are often used to configure Digital Right Management (DRM) mechanisms.

### 6.4.6 Privacy-Focused Policy Languages

Privacy protection within security management is becoming increasingly important since resources are highly interconnected today [22]. Thus, privacy policy languages are required to be designed in a way that can be handled by unqualified users. Furthermore, it might be even more useful if these common users are involved at design time. That is, the policy language needs to be not only editable by humans, but also negotiable to some extent. W3C's platform for privacy preferences project (P3P) and the preference exchange language (APPEL) [40] is used for privacy negotiations between a website and its users. In fact, it collects user's preference captured from GUI, and can make automated or semi-automated decisions about machine-readable privacy policies. Alternatively, Enterprise Privacy Authorization

Language (EPAL) [41] is an XML-based language. It is similar to XACML but focuses only on privacy policies and the differential aspect is the purpose-based authorization decision, which is fully aligned with legal and regulatory requirements.

### 6.4.7 *Capability-Based Access Control CapBAC*

A capability-based access control system (CapBAC) [42] is used as the access control system based on capabilities defined as tokens, tickets, or keys that give the possessor permission to access an entity or object in a computer system. The capability concept was introduced in 1996 to be applied in the operating systems dealing with several processes accessing memory spaces. The reviewed approach for access control provides authorization enforcement as well as some additional features. (1) Delegation support: a subject might give permissions to another subject, as well as the right to further give the received permissions in a limited depth. (2) Capability revocation: capabilities and enclosed permissions can be revoked by authorized subjects. (3) Information granularity: a capability might enclose details of specific permissions on particular data under some conditions.

Figure 6.6 shows a capability specification where policy rights expressiveness can be notably high.


**Fig. 6.6** Capability token XML schema: Access Rights Capability Type definition conveying details of the granted Access Rights (AR) type

**Table 6.1** Summarized overview of foundational policy languages

| | Conditions | | | | | |
|---|---|---|---|---|---|---|
| | Role matching | Attribute matching | Expressive functions on attribute sets | Obligations | Privacy | Delegation |
| XACML | x | x | x | x | | x |
| Rei | x | x | x | x | x | x |
| Ponder | x | x | x | x | | x |
| ASL | x | | | | | x |
| OSL | | x | x | x | | |
| APPEL | | x | | | x | |
| EPAL | | x | | | x | |
| CapBAC | | x | | | | x |

### *6.4.8 Discussion on Foundational Approaches*

In the envisioned scenarios, flexibility is enabled by policy-driven authorization approaches, and according to Sloman [43], security policies define the relationship between subjects and targets. But there is always a trade-off between the expressiveness of the policy and feasibility. A full list of policy languages and frameworks has been collected in [44], but none of them are optimized for constrained devices, and consequently they are not feasible. Concretely, broadly adopted implementations such as (XACML) [32], Rei [37], and Ponder [36], although policy driven and very expressive, behave resource exhausting in constrained devices [22].

Table 6.1 shows a summary of features supported by each of the considered policy languages. All of them support *if-then* or *condition-action* paradigm with different attribute treatment. Instead, ECA, supporting the composition of events as triggers, is only supported by Ponder in this overview. In all cases, *conditions* can be formulated as simple role matching, attribute matching, or more powerful and dynamic expressions on attribute sets. The simplicity of the first one is balanced versus the granularity and tightness of the last one. *Obligations* are used to launch additional tasks if the condition is met, and it is the foundation for reactions and usage control. *Privacy* refers to the support of features to protect the privacy of subjects. In fact, it is linked to the *delegation*, since subjects might require to define and negotiate privacy preferences.

As a representative by its level of adoption, XACML adopts generic authorization architecture and specifies a policy language to express and exchange authorization policies represented in XML. It also provides request–response semantics for messages enclosing authorization decisions to facilitate the enforcement. Therefore, it is a good option for ABAC, and its architecture decouples the policy edition, storage, decision making, and enforcement, which makes it an attractive solution for the envisioned scenarios. However, XACML is too heavy for severely constrained devices. In fact, a CPS can hardly process a XACML policy file of more than 50 lines of text conveying a single rule specification.

Moreover, although XACML is a suitable policy language, it cannot adequately represent semantics or entities relationships as Rei and Ponder do. In fact, Rei, which is oriented to semantic web applications, is based on Web Ontology Language Lite (OWL-Lite) [45] and supported by Resource Description Framework (RDF) [46] enabling further reasoning over policies and related domain knowledge. In a similar way, Ponder, being a declarative object-oriented language that can be used to specify both security and management policies, is also based on OWL and Semantic Web Rule Language (SWRL) [47]. On the contrary, the policy ontologies and rules are still domain dependent in some ways and the reasoning engines, on the one hand, add complexity and uncertainty to the final granting decision making and, on the other hand, raise significantly the processing and storage requirements, making these semantically advanced approaches definitively infeasible in any constrained device.

By means of the usage control enabling control over both who may access and how the data may or may not be accessed afterward, XACML, Rei, Ponder, and OSL enable such a policy specification through the obligations. However, XACML lacks also specific privacy specification features as Rei, APPEL, and EPAL do. Moreover, by means of delegation, XACML is not as simple and agile as CapBAC, which grants access to any subject in the possession of a valid capability or token.

Additionally, XACML like other fully expressive policy languages might rely on other traditional authentication and authorization standards and protocols such as SAML [34], OAuth [35], OpenID [48], Fido [49, 50], etc. These solutions enable cross-domain multi-factor authentication-based authorization, but they involve message exchange protocols which are not optimized for severely constrained devices. The message lengths, the transport protocols, and the adopted cryptographic schemas are not feasible in CPSs with constrained energy consumption, memory footprints, and CPUs.

In any case, conveyed models and standards with their strengths and drawbacks pave the ground for any access control solution ad hoc to IoT integrating CPSs implemented in constrained devices.

## 6.5 IoT Tailored Access Control Approaches

As traditional access control models are not feasible, new access control models specifically optimized for IoT environments have been proposed recently. Among them, seven different approaches are conveyed hereinafter, and feasible from C2 to C0 constrained devices. First, the so extended XACML has been adapted to IoT as a centralized ABAC implementation. Alternatively, UCON adapted to IoT is conveyed, which proposes a distributed access control enforcement model not only before the first access but also during it. And the third most differentiated approach is the capability-based access control with two proposals, fourth and fifth, which convey progressive distribution of the authorization decision and enforcement based on capabilities distributed based in a PKC schema. The sixth approach

proposes a token-based access control that can be seen similar to previous ones but fully compliant with constrained environment standards such as CoAP and DTLS. Finally, the seventh proposal that is absolutely optimized and feasible in severely constrained scenarios proposes an RBAC authorization enforcement based on a SKC schema.

### *6.5.1 Authorization Framework for the IoT Based on XACML*

This proposal [51] conveys an authorization framework based on assertions as a result of an authorization process based on XACML. This assertion is encoded in JSON and is sent to the CPS. The CPS additionally checks locally the conditions conveyed in the authorization assertion.

#### 6.5.1.1 Basic Operation

This approach supports fine-grained and flexible access control to CPSs. Evaluating XACML policies is too heavyweight for CPSs, so most of the authorization decision process is externalized, and the CPS has to perform the enforcement. The framework relies on current Internet and access control standards, and the execution times on a CPS are reasonable.

In order to convey the authorization decisions from the external decision point to the device, assertions are used, which are digitally signed data objects containing asserted information, and in particular SAML authorization decision assertions are used as a template. A similar alternative is the use of OAuth access tokens, but since XACML on SAML is broadly adopted, it has been the preferred option for the authors.

Therefore, three entities are integrated in the framework: a CPS hosting resources, a subject that attempts to access a resource in the CPS, and an ACS acting as trusted third party. In fact, ACS makes the authorization decision evaluating the correspondent policy and releases related authorization assertions to the subject. The subject sends these assertions to the CPS along with the request. The ACS has a repository storing previously configured access policies to each protected CPS. The resulting architecture is illustrated in Fig. 6.7.

A key establishment procedure is also defined. This proposal claims that the authorization framework requires neither a particular authentication protocol nor any key agreement procedure. Nevertheless, the key establishment is considered as an option. Two approaches are considered.

On the one hand, DTLS based on raw public keys or pre-shared keys provides encryption, integrity, and replay protection of CoAP messages. However, the DTLS handshake might impose a significant impact on performance and setup time, and this option does not scale well for severely constrained C0 and C1 CPSs with poor storage capabilities.

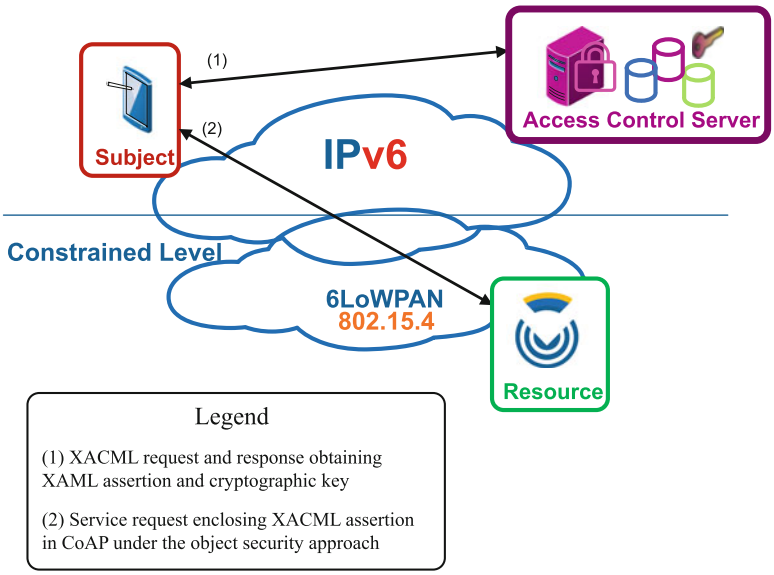



**Fig. 6.7** The authorization architecture

On the other hand, an object security approach using asymmetric keys for object protection is proposed. A nonce is included in the assertion and with the well-known public keys a Diffie–Hellman calculation is done to derive the secret key shared between ACS and CPS. In a similar way, including the public key of the subject in the assertion and a nonce in the payload, the secret key shared between CPS and the subject can be calculated. This can be simplified if pre-shared keys are available among CPS and ACS, so instead of the public key of the user a pseudonym can be included, and one-way functions are computed.

Hence, given any key establishment between CPS and ACS it is assumed that shared symmetric keys *kACSCPS* and *kSCPS* with ACS and subject, respectively, are available in the CPS after reception of the assertion. These shared keys are the basis for securing the data objects passed between ACS and CPS (assertions), and the subject and CPS (payloads), respectively.

#### 6.5.1.2 Tightness and Feasibility Discussion

The subject owning the proper authorization assertion sends a CoAP request to the CPS. Since CoAP supports options between header and payload, a specifically defined *assertion option* is enclosed in the secured request. Moreover, the CoAP payloads might be replaced with object-secured equivalents based on the CPS Key.

The CPS at the reception of such a CoAP request (1) checks the validity of the authorization assertion, (2) checks the permissions enclosed in the assertion with

the actual access request, and (3) checks the local optional conditions enclosed as evaluable parameters in the assertion. There is no detail on the expressiveness of such conditions specification.

In the case of a positive verification, the request is granted and the correspondent response is processed and released. The authorization assertions convey also a short, predefined validity lifetime to avoid replay attacks, and the CPS keeps a record with a list of recently used authorization assertion identifiers.

Since the full syntax of XACML Responses and SAML Assertions includes a large number of features, in one particular implementation [51] it has been defined a subset of both standards, in order to simplify the processing on the CPS. Furthermore, the XML representation of this subset is too verbose for efficient transmission over limited channels, so it has been defined a compact JSON-based notation for the SAML and XACML subset. This approach reduces the size of the assertion roughly by a factor of ten.

The CPS part has been implemented and validated in a platform composed by an Arduino Mega 2560 board3, with 16 MHz processor, 256 kB of Flash Memory, 8 kB of SRAM, and 4 kB of EEPROM. These capabilities are representative of a constrained CPS. The implementation relies on the object security approach.

For wrapping the assertion and payload, the IETF JSON Web Encryption (JWE), an emerging secure object standard, is used. This wrapping expands the payload size drastically.

Furthermore, it requires a centralized authorization server (PDP) to make such granting decisions. SAML and XACML assertions might be processed by devices in IoT, but even they have been redefined in reduced version and represented in JSON, and they are not processable in severely constrained devices, playing neither as servers nor as subjects. So this approach is not feasible in severely constrained devices scenarios.

Summarizing as shown in Table 6.2, this XACML approach adapted to IoT enables very high policy language expressiveness by means of the authorization decision-making delegation to an unconstrained trusted third party. Therefore, the enabled tightness is very high, as well as the scalability and the flexibility on the policy. However, it lacks of local context awareness at enforcement time, and, additionally, the message exchange, the network overload, and the assertion processing requirements are too heavy for severely constrained CPSs making it infeasible.

**Table 6.2** Summary of XACML' analysis highlighting the high policy language expressiveness, the scalability and the flexibility on the policy, but down-lighting the infeasibility in very constrained CPSs such as C0 and C1 classes

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| XACML' | High | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |

### 6.5.2 *Usage-Based Access Control Adapted to IoT (UCON)*

UCON [31] and the attribute-based policy schema [52] are completely different alternatives that extend traditional unconstrained access control systems. This approach proposes the continuous protection of the resources during access, and it considers also consumption activities rather than granting access only before the first access. The proposal [33], which adapts UCON to IoT, is based on Fuzzy theory and some central entities which manage usage control decisions and trust values of devices and services.

#### 6.5.2.1 Basic Operation

The architecture defines three main entities where besides a subject and a resource an ACS integrates a trust management center, a registration center, and an access control module.

The abstraction of the UCON model in IoT is shown in Fig. 6.8. The subject of UCON in IoT runs the control of the application service in the CPS. The attributes of the subject in the ACS include information about the trust value of the subject, the honest usage times of the application services, and dishonest usage time of the

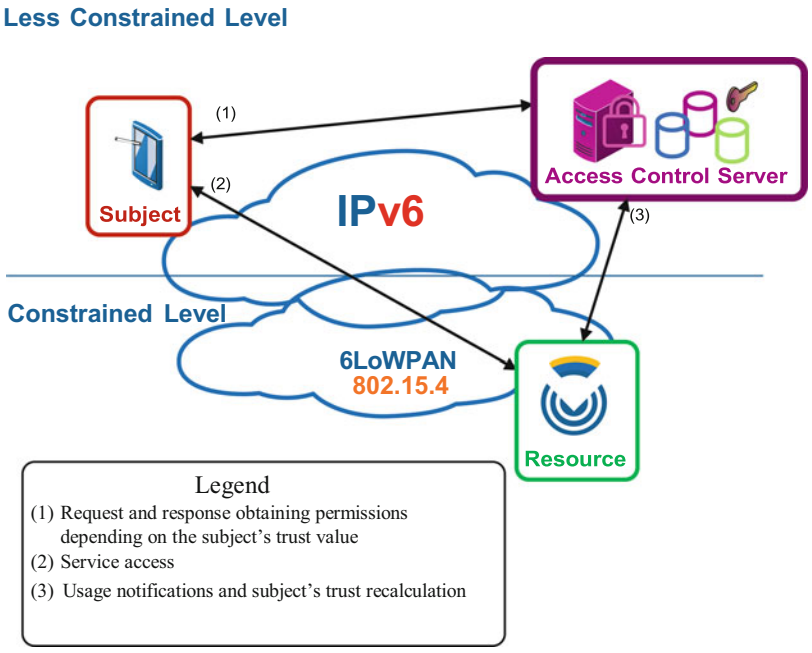



**Fig. 6.8** UCON for IoT model architecture

application services, and some other properties and so on. The condition (C) of UCON in IoT is decided by the policies according to the trust value of the subject and other decision context factors. The oBligation (B) of UCON in IoT is according to the needs of the application services in the CPS to perform any advertising action and so on. The obligation might be performed before or during the usage control in IoT. The Authorization (A) of UCON in IoT is set by the needs of usage control and behaves as the functional entity where requests should be evaluated, and then returns whether the subject has rights to use the application service in the CPS.

The process of the access control runs as shown in Fig. 6.8: (1) the subject queries the resources in the CPS and requests for the usage control in the ACS. Trust value of the subject and trust threshold of the resources on the CPS are compared, and in the positive case, the request is promised and the process proceeds with the second step, otherwise the process is finalized. (2) The subject accesses the resource on the CPS. (3) The CPS sends the subject's feedback to the ACS, which evaluates whether the access control is honest according to the feedback and updates the subject's information and the CPS's information, such as the trust value of the subject for next requests.

#### 6.5.2.2 Tightness and Feasibility Discussion

This approach includes obligations such as usage control but does not include a proposal addressing its feasibility in any CPS class. Although some experiments are presented, the practical feasibility of the approach is not demonstrated in any constrained device. Additionally, the constrained device being accessed by a subject has to notify the usage to the central entities on a regular basis or per use. This means a network overload besides the energy consumption due communications.

In summary, the UCON approach adapted to IoT enables high policy language expressiveness so the enabled tightness is high, as well as the scalability, the flexibility on the policy, and the local awareness. However, the message exchange, the network overload, and the assertion processing requirements are too heavy for severely constrained devices making it infeasible (Table 6.3).

**Table 6.3** Summary of UCON analysis highlighting the high policy language expressiveness, the scalability, the flexibility on the policy, and the local context awareness, but down-lighting the infeasibility in very constrained CPSs such as C0 and C1 classes

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| UCON | High | High | C2-CN | Yes | None | n/a | Yes |

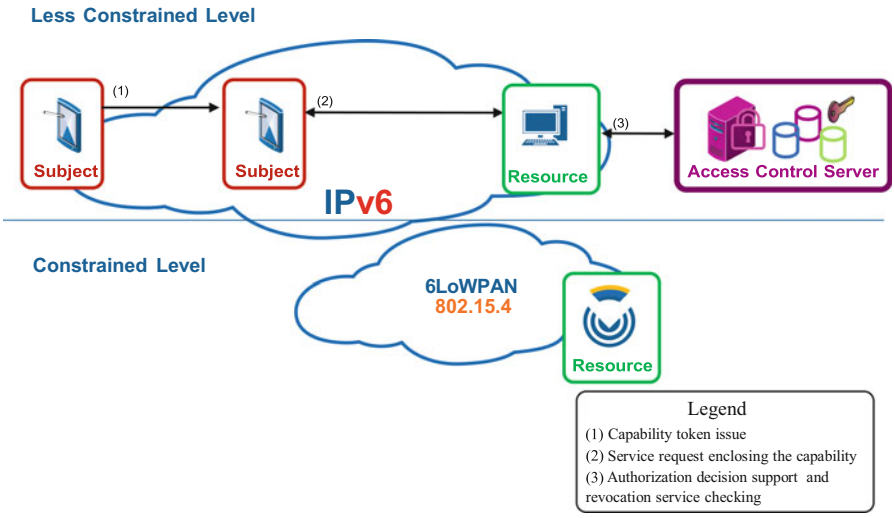



**Fig. 6.9** Capability-based access control schematic concept

### *6.5.3 CapBAC in IoT*

The capability-based security approach to manage access control (CapBAC) in IoT [53] is based on the capability, which is a communicable, unforgeable token of authority as shown in Fig. 6.9. Moreover, by virtue of its possession a subject is granted to access a resource under certain conditions.

#### 6.5.3.1 Basic Operation

Compared with the traditional ACL system, where the server is in charge of checking whether a subject is authorized to perform the requested operation on the requested resource, in this CapBAC model it is the subject who has to obtain and present a valid authorization capability and the server is relieved of further authorization decision making. This is not a new concept but in this proposal it is adapted to the IoT.

This proposal claims a tighter security based on a better adherence to the least privilege principle and a high granularity of the authorization tokens. It also claims the delegation of the management functions out of the CPS itself and the flexibility derived of instant token generation. Additionally, this proposal claims to enable delegation support, capability revocation, and information granularity.

In the simplest architecture, a CPS receiving a request with a token from the subject first checks the token. Formal validation to check the correctness, lifetime expiration, etc. and the logical validation to check the congruence of the request are performed prior to checking externally the authorization decision. This is done against an entity acting as PDP, who recursively checks also the revocation lists.

**Table 6.4** Summary of CapBAC' analysis highlighting the high policy language expressiveness, the scalability and the flexibility on the policy, but down-lighting the infeasibility in any constrained CPSs

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| CapBAC' | High | Low | CN | Yes | HTTP/REST | PKC | Yes |

#### 6.5.3.2 Tightness and Feasibility Discussion

This proposal has been implemented successfully in Java as a set of libraries, tools, and services. Capability tokens are represented by digitally signed XML files. This implementation can be used in appliances and relatively small devices such as smartphones, tablets, which can process XML files and digital signatures based on X.509 certificates.

This approach is flexible since a set of parameters such as validity period, assigned rights, delegation depth, and resource granularity can be tuned, although proper skills are required in this process.

Summarizing as shown in Table 6.4, this CapBAC approach adapted to IoT enables high policy language expressiveness so the enabled tightness is high, as well as the scalability and the flexibility on the policy. However, it lacks of local context awareness at enforcement time, and, additionally, the message exchange, the network overload, and the assertion processing requirements are too heavy for any constrained CPSs making it infeasible.

### *6.5.4 Distributed CapBAC in IoT*

Distributed capability-based access control for Internet of Things [54, 55] is a cryptographic solution that allows E2E access control without the intervention of any intermediate entity, and supports the management of certificates, authentication, and authorization processes.

It claims to make use of technologies specifically designed for IoT environment, unlike [53] conveyed in Sect. 6.5.4, and additionally enables authorization decisions based on local conditions offering context-aware access control.

It inherits the benefits of the capability-based approach: (1) distributed management, (2) delegation, (3) traceability, (4) scalable authentication chains, and (5) standard certificates based on ECC. Concretely, a capability token that is signed with the Elliptic Curve Digital Signature Algorithm (ECDSA) has been designed for CoAP resources, so E2E authentication, integrity, and non-repudiation are guaranteed.

#### 6.5.4.1 Basic Operation

The key concept of this approach is the capability, which originally was defined as a token, ticket, or key that gives the possessor permission to access an entity or object in a computer system. Additionally, such capability is tamper-proof and unequivocally identified. Optionally, a set of rights granted to the possessor can be defined. The simplest option considers that the CPS receiving a request with a capability knows also the correspondent right level. This assumption implies both the simplicity of the authorization mechanism in the CPS but also the lack of flexibility.

Finally, it also supports inherently the delegation from a subject to others and the revocation of the capabilities and associated permissions.

In the implementation of this proposal, JSON is the format to represent the capability token because of its suitability in constrained environments, such as those suggested by IoT scenarios. The capability conveys several fields where the access rights are defined as a set of action-resource conditions, and conditions are defined as type-value-unit trios.

This basic operation consists of several steps, and it starts when a subject gets a capability token from a capability issuer, (Fig. 6.10). This token is signed with ECDSA by the issuer. Such signature is attached to the capability token. The delivery can be preceded by authentication and authorization by the issuer but it is out of scope on the proposal. Then the subject attempts to access the services in a CPS attaching the token into the *payload* of a CoAP request. Additional *Content-Format* header and *Request-Uri* are used to specify the format of the payload and the resource to be accessed, respectively. Finally, the subject also signs the request itself using ECDSA algorithm, whose value is attached to the request by adding a

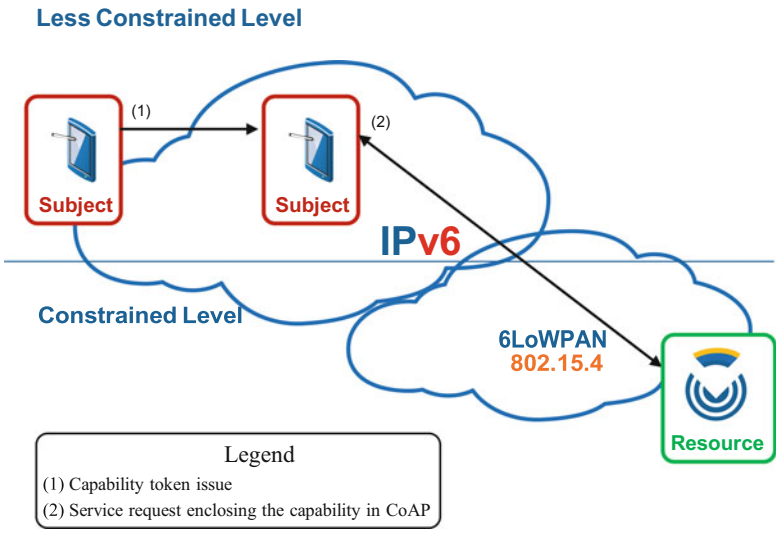



**Fig. 6.10** Distributed capability-based access control architecture and flow

new header called *Signature*. When the CPS receives the request, first it checks the validity of the token. Then it checks sequentially the rights and conditions enclosed in the token. In the positive case, the signature of the token is checked using the public key of the issuer, and finally the authentication of the subject is performed using the public key enclosed in the token and the signature of the CoAP request. These two operations are done last to optimize energy consumption due to their cost. Finally, once authorization decision is made, consequent CoAP response is sent to the subject.

This proposal conveys a set of conditions as security policy enclosed in the token. These conditions need to be met locally in the CPS and might refer to any data available in the CPS. This feature is a promising step forward in the tightness and the dynamic of the access control enforcement. However, the specification of the conditions is limited to the matching function of constant values of some data considered as local context. It does not support either the definition of expressions instead of constants or advanced features such as reactive actions to be performed as derived obligations.

From the security point of view, it is not desirable of the need for clock synchronization required by the way to define the lifetime of the token. An additional drawback is the lack of protection against reply attacks. In the feature set of this approach token revocation is mentioned, but the steps in the basic operation do not enable to carry it out, unless a notification is done to the CPS implying both higher energy consumption and memory footprints. How the token is issued is not described. Moreover, the token is issued signed by the issuer so integrity is guaranteed, but confidentiality is not since it does not travel encrypted, unless CoAP requests include DTLS. Additionally, even it is not mentioned in the proposal, this would require a previous handshake, and a significant impact on the energy consumption.

#### 6.5.4.2 Tightness and Feasibility Discussion

This approach [54] has been implemented in C2 CPSs, and 480.96 ms is the average time for the operation of request, authorization, and response, although no token samples have been described. The used ECC and ECDSA cryptographic functions have a cost by means of resources that C0 and C1 cannot afford and, therefore, feasibility in these CPS ranges is not achievable. It is worthy of note that token format is JSON and that the minimum length is over 160 bytes, what points out the need of packet fragmentation considering the short payload available in 802.15.4/6Low-PAN/CoAP packets (127 bytes). This fact could impact increasingly in the performance as local context conditions are specified.

Regarding tightness, local context in the CPS is not considered. When a subject tries to access a service in a CPS it attaches a capability token to the access request sent first to the PDP. The PDP is responsible for making the granting decision if the subject presenting such capability is authorized or not to access the CPS, but local conditions are not considered.

**Table 6.5** Summary of DCapBAC analysis highlighting the high policy language expressiveness, the scalability and the flexibility on the policy, but down-lighting the infeasibility in severely constrained CPSs

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| DCapBAC | High | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |

Additionally, the centralized approach results in lower performance compared with the distributed one.

Summarizing as shown in Table 6.5, this DCapBAC approach adapted to IoT enables high policy language expressiveness so the enabled tightness is high, as well as the scalability and the flexibility on the policy. However, it lacks of local context awareness at enforcement time, and, additionally, the message exchange, the network overload, and the assertion processing requirements are too heavy for severely constrained CPSs making it infeasible.

### *6.5.5 Delegated CoAP Authentication and Authorization Framework (DCAF)*

Delegated CoAP authentication and authorization framework (DCAF) [56] defines a protocol for delegating subject authentication and authorization in a constrained environment for establishing a Datagram Transport Layer Security (DTLS) channel between resource-constrained nodes as shown in Fig. 6.11.

#### 6.5.5.1 Basic Operation

Inthis approach, authorization information and shared symmetric keys between E2E endpoints are transferred through DTLS. In this proposal, the authentication of E2E endpoints and the provision of authorization information are delegated to a trusted third party with no computation or memory constraints.

DCAF relies on access tokens to enforce access control, so a requester subject previously has to obtain a permission enclosed in an unforgeable token from the CPS's Authorization Manager. Such access tokens contain authentication and, optionally, the authorization information needed to access the CPS. It also encloses the pre-shared key (PSK) for the communication between the subject and the CPS. Once a subject has got a valid access ticket it uses obtained PSK to establish a secure channel with the CPS through a key exchange algorithm (handshake).

When the secure channel is established, authorized resource requests can be sent by the subject to the CPS. The CPS will check every CoAP request confronting it against the authorization information received in the ticket. Concretely, which action

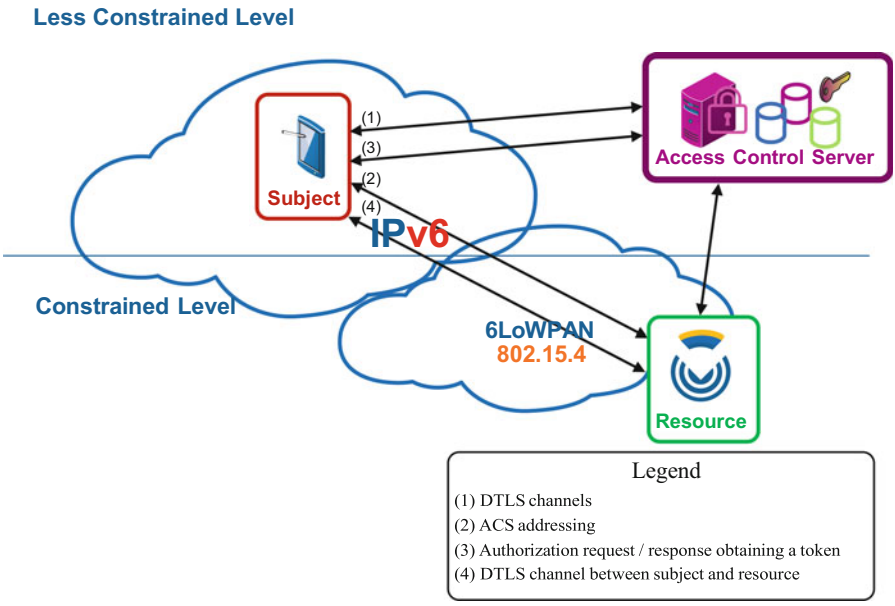



**Fig. 6.11** Delegated CoAP authentication and authorization framework (DCAF) architecture and flow

is the subject allowed to perform on a resource of CPS. Aforesaid authorization information is defined as a data structure to describe subject's permissions for CPS's resources. In such data object, the resources are the local part of the URI (Uri-Path and Uri-query options of CoAP). The permissions are simply the CoAP methods of *GET*, *PUT*, *POST*, or *DELETE*. In such authorization information schema, basic condition specification is not supported and, therefore, local context checking is not supported.

#### 6.5.5.2 Tightness and Feasibility Discussion

This approach is based on using a token to distribute pre-shared keys. Then, if authorized, a handshake is performed to establish a DTLS channel. DTLS has been defined as the basic building block for protecting CoAP, but few implementations for small, constrained devices are available. TinyDTLS [57] has been developed offering the first open-source implementation of the protocol for small devices. However, a performance evaluation and a feasibility assessment analysis are required. In fact, as well as TLS, DTLS was designed for traditional computer networks and, thus, some of its features may not be optimal for resource-constrained networks. For instance, the loss of a message requires the retransmission of all messages inflight. A smaller MTU leads to higher fragmentation of messages, which implies large buffer requirements and more re-ordering and reassembly processing to compose the

**Table 6.6** Summary of DCAF analysis stating the medium policy language expressiveness, the scalability, and the flexibility on the policy, but down-lighting the infeasibility in severely constrained CPSs

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| DCAF | Medium | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |

complete DTLS message. Moreover, the fragmentation of the handshake messages allows easy denial of service attacks. Finally, the DTLS handshake is completed only if the verification of the *Finished* message is successful, which checks all previous handshake messages and, therefore, requires a large buffer to queue them.

A DTLS handshake involves significant computation, communication, and memory overheads in the context of constrained devices. The RAM requirements of DTLS handshakes with public key cryptography are prohibitive for certain constrained devices such as C0 and C1. Finally, certificate-based DTLS handshakes require significant volumes of communication, RAM (message buffers), and computation. Therefore, this approach is only feasible from C2 CPSs on, and the memory limitations of CPSs restrict the number of DTLS sessions.

When assessing the performance of DTLS, besides cryptographic mechanisms, enclosed security policies and their impact on the system performance and network overhead should also be considered in the analysis.

Authorization policies are specified as a basic set of actions granted to be performed by a subject accessing a resource in a CPS. Additionally, CBOR serialization [58] instead of the JSON representation for policies enables to compact the payloads in the CoAP protocol. In particular, the length of the capabilities and the enclosed policies based on conditions on attributes are significantly compressed.

Regarding security features analysis, the expressiveness of the policy is coarse-grained and local context is not considered at all. Given that CBOR is a general-purpose serialization solution, the resulting compression of the policy is still not sufficient for the C0 and C1 CPSs considered in the envisioned scenarios; it is feasible only in C2 CPSs.

Summarizing as shown in Table 6.6, this DCAF approach enables medium policy language expressiveness so the enabled tightness is also medium, as well as the scalability and the flexibility on the policy. However, it lacks local context awareness at enforcement time, and, additionally, the message exchange, the network overload, and the assertion processing requirements are too heavy for severely constrained CPSs making it infeasible.

### *6.5.6 OSCAR*

OSCAR [59] is an architecture for E2E security in the IoT. It is based on the concept of object security that relates security with the application payload.

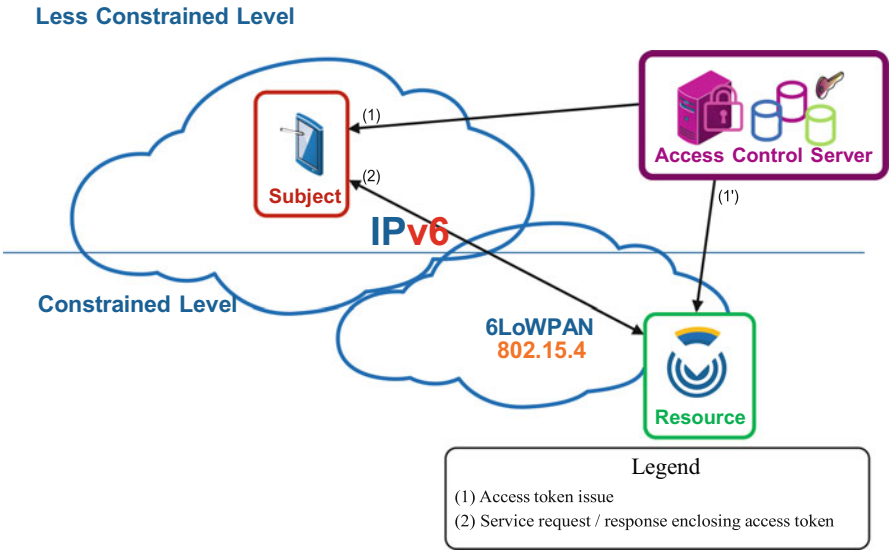



**Fig. 6.12** OSCAR: Object security architecture and flow

#### 6.5.6.1 Basic Operation

The architecture includes an Authorization Server that provides subjects with access secrets that enable them to request resources from constrained CoAP nodes as shown in Fig. 6.12. The nodes reply with the requested resources that are signed and encrypted. The scheme intrinsically supports multicast, asynchronous traffic, and caching.

The security requirements of IoT are tackled at the network level and it adopts the Representational State Transfer (REST) architecture model, so it behaves stateless between the server and client. This feature is achieved through the concept of object security that involves data security instead of communication endpoints.

In the OSCAR architecture, some computationally expensive operations are off-loaded from constrained CoAP CPSs to more powerful workstations. Specifically, constrained CoAP CPSs publish their certificates to *Authorization Servers* (AS), so subjects first obtain properly signed and encrypted *Access Secrets* from the AS to be authorized to access resources from constrained CoAP CPSs.

The scheme combines the object security principle with the capability-based access control to provide communication confidentiality and protect CPSs from replay attacks. Yet, a vast amount of work is based on the DTLS protocol and it uses secure channels for authenticated certificate and *Access Secret* distribution. So it brings together the concepts of connection-oriented security (DTLS for authorization information) with those of content centric networking following REST approach.

Concretely, both subjects and CPSs have valid certificates. An access secret is a token generated by the Authorization Server and delivered to both the subject and

CPS. The token encloses a symmetric key to encrypt the resource representation in a CPS, and it is also used by the subject in the access request. The CPS returns the resource signed with its private key and encrypted with the shared symmetric key. These two actions can be performed during sleep time and be ready for service time. It also enables caching and it is clearly oriented to content delivery services. Otherwise, it is not oriented to requests beyond GET content method.

#### 6.5.6.2 Tightness and Feasibility Discussion

OSCAR does not support local context-based access control enforcement in the CPS. It has been evaluated in two hardware platforms [59]: (1) WiSMote platform based on 16-bit MSP430 (series 5) microcontroller unit (MCU) with 16 kB of RAM and an 802.15.4-compatible CC2520 radio transceiver; and (2) ST GreenNet tag, an energy-harvested prototype platform from STMicroelectronics based on an ultralow-power 32-bit ARM Cortex-M3 MCU (STM32L) with 32 kB of RAM and an 802.15.4 radio transceiver. The results show that OSCAR outperforms a security scheme based on DTLS when the number of nodes increases. OSCAR also results in low energy consumption and latency for C2 CPSs, but it does not perform so well in C1 CPSs.

The implementations point out requirements of additional 2156 bytes of ROM and 500 bytes of RAM due to the library to parse the security objects, and due to the storage of the ECC certificate of the CPS, the corresponding private key, and the root trust certificate as the minimal necessary cryptographic material. Therefore, it is not feasible in C0 and C1 CPSs. Moreover, the excessive overhead of the DTLS handshake is analyzed in [60].

Summarizing as shown in Table 6.7, this DCapBAC approach adapted to IoT enables high policy language expressiveness so the enabled tightness is high, as well as the scalability and the flexibility on the policy. However, it lacks local context awareness at enforcement time, and, additionally, the message exchange, the network overload, and the assertion processing requirements are too heavy for severely constrained CPSs making it infeasible.

### *6.5.7 Ladon*

Ladon [61, 62] enables mutual authentication and authorization in severely constrained CPSs C0 and C1 through the establishment of E2E security associations

**Table 6.7** Summary of OSCAR analysis highlighting the high policy language expressiveness, the scalability and the flexibility on the policy, but down-lighting the infeasibility in severely constrained CPSs

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| OSCAR | High | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |

based on pairwise keys. Ladon, which has been formally validated, ensures confidentiality, integrity, and non-repudiation, and it is resistant to message losses as well as replay attacks. It is based on Kerberos, but it avoids the requirement for the clock synchronization and it is very efficient in terms of consumed energy.

#### 6.5.7.1 Basic Operation

The operation of Ladon is organized into three different phases: the authentication phase, the authorization phase, and the service access phase as shown in Fig. 6.13. In the authentication phase, the Ladon authentication server (AS) verifies the identity claimed by the requesting subject. In a positive case, the subject obtains a ticket granting ticket (TGT), which is a proof of the identity to request further service tickets to the ticket granting server (TGS) during the validity period of the TGT.

In the authorization phase, the TGS checks if a legitimately authenticated subject has permissions to access resources in a CPS. In a positive case, both the CPS and the requesting subject are provided with the information needed to establish an E2E secure association.

Finally, during the service access phase, the subject presents the service ticket enclosed in the request to the CPS, who checks it, and in the positive case, the CPS responds with the requested information.

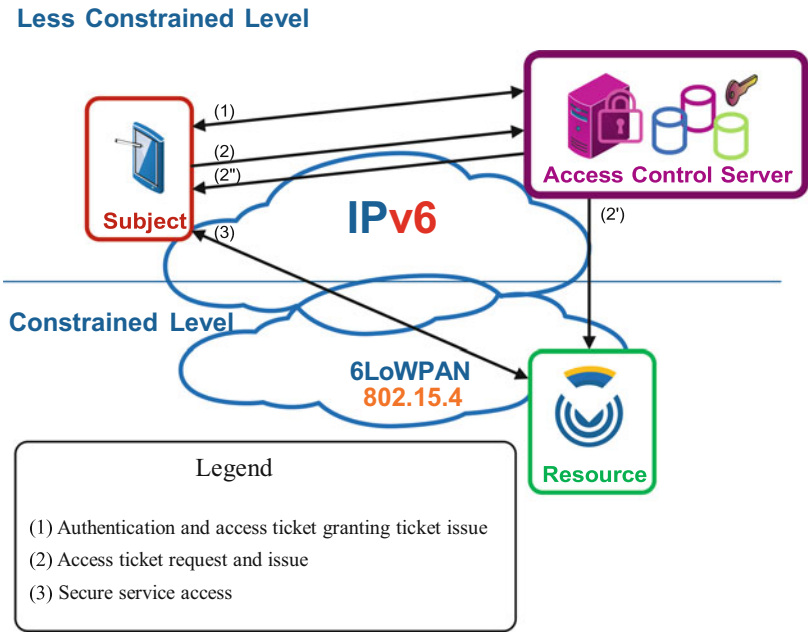



**Fig. 6.13** Ladon architecture and main steps

The proposed authorization model is based on a combined design of RBAC and ABAC, integrating attributes with pure RBAC. It follows an attribute-centric approach where a role is not a collection of permissions, but the name of an attribute called role. Each resource to be accessed in the CPS is preconfigured with the accepted role value that is enclosed in the service ticket by the ACS dynamically. In fact, the ACS might enclose different role values depending not only on the identity of the subject but considering also some context conditions evaluated by the ACS.

But this dynamism affects only the central entity making the authorization decision. After authenticating an incoming service ticket request, the TGS makes the authorization decision, and in a positive case, it generates a service ticket with the corresponding role embedded as an authenticated attribute. Therefore, Ladon service tickets assert both, the veracity of the identity claimed by the subject and his right to access the requested service.

#### 6.5.7.2 Tightness and Feasibility Discussion

Ladon has been validated in C0 CPSs. However, Ladon does not support policy provisioning and accounting functionalities. In fact, Ladon defines a preconfigured static policy specified as the allowed values for subject's role in the CPSs, and it supports the secure transmission of such authenticated role values in the subject's requests. This behaves neither flexible nor scalable.

Summarizing as shown in Table 6.8, this Ladon approach is feasible in severely constrained devices even with a low tightness in enforcement as well as scalable. However, it lacks flexibility on the policy and local context awareness at enforcement time.

### *6.5.8 Hidra*

Hidra [63] is a formally validated solution for establishing E2E security associations through pairwise keys, guaranteeing mutual authentication, expressive policy injection, tight enforcement, and accounting for further tracking and auditing purposes. This proposal claims to be the first to enable tight access control suitable for severely constrained C0 CPSs. This novel model tackles the limitations of current access control approaches for constrained devices by defining an expressive but lightweight

**Table 6.8** Summary of Ladon analysis highlighting the feasibility in severely constrained CPSs and the scalability, but down-lighting the low enforcement tightness, the low policy language expressiveness, and the lack of flexibility

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| LADON | Low | None | C0-CN | Yes | UDP | SKC | No |

policy language. Moreover, it also enables tight policy enforcement in CPSs based on local context conditions and corresponding obligations. Furthermore, policy evaluation and enforcement is performed not only during the security association process but also afterward, while the security association is in use. In order to optimize both computation and storage requirements in CPSs, such a dynamic policy cycle specifies an efficient message exchange protocol.

#### 6.5.8.1 Basic Operation

Hidra enables the E2E security association establishment between the requesting subject and the sensor acting as a tiny information server. Hidra is based on SKC and each endpoint is assumed to own a secret key shared with the ACS. The basic operation relies on the use of tickets [63, 64], a token that is distributed by the ACS, which contains a proof of the identity of the requesting subject. Such tokens are encrypted with the proper secret keys so that only the entities which they are intended for are able to decrypt them.

Hidra, depicted in Fig. 6.14, is based on a three-party architecture, and provides authentication, authorization in two steps, dynamic policy provisioning, and accounting through four phases. First a subject addresses the ACS, and after a successful authentication (1) the subject that wants to access a service in the CDS obtains a ticket granting ticket (TGT).

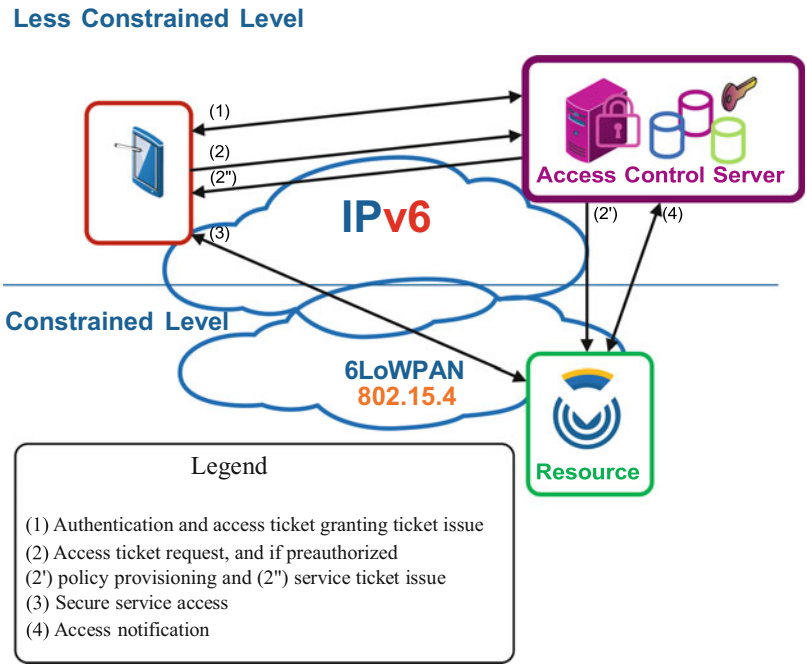



**Fig. 6.14** Hidra architecture and main steps

(1) This TGT is enclosed by the subject in the request to obtain resource tickets required to access any resource on the CPS. This solution supports the ABAC enforcement in two steps. In the first one, as condition to release any resource ticket, fine-grained primary authorization is performed in the ACS (2), based on the attributes of the subject, requested resource, and expected actions. In the case of a positive authorization decision, the ACS sends a message to the CPS conveying an expressive authorization policy instance (2′), and also sends a message to the subject including a resource ticket (2″). This dynamic policy deployment avoids the permanent storage of the policies in the CPS and reduces significantly network overhead compared with approaches enclosing the policy in the resource ticket.

In the second authoritative step, the subject accesses the CPS enclosing the obtained resource ticket in the request, and the local context-based access control is performed in the CPS (3). First, the correspondent rule is evaluated to make the granting decision, and then the corresponding reactive obligations are enforced. In a positive authorization case, a shared session key is established to be used on further E2E resource accesses.

Hidra supports also a pair of messages to enable precise accounting (4). By means of these messages, the CPS notifies details like who performed what action, where, and when. These notifications are collected, normalized, and processed by the ACS. Additionally, the ACS can react and send a properly generated policy message, enabling the dynamic delegation, request, cancellation, and revocation of permissions.

Then, whereas the lifetime is not over or security association is not abruptly finalized, policy (2′)-based tight authorization is enforced in the CPS autonomously in each and every further request attempt (3).

Therefore, unified, coherent, and adaptive policy management is performed by the ACS. Additionally, the adopted architecture enables to rely on the most expensive features on the ACS, which entails the usage of standard security and access control technologies in the unconstrained interactions. This feature enables the denial of the most unauthorized access attempts before reaching the CPS, avoiding unsuccessful message exchanges and, thus, saving energy in the CPS, which is a crucial aspect.

#### 6.5.8.2 Tightness and Feasibility Discussion

Hidra is fully compliant with both enforcement tightness and feasibility requirements. The specific purpose policy language can be seen as a balanced subset of XACML and enables fine-granularity through expressive conditions, obligations, and usage control. Additionally, Hidra enables a two-step authorization as well as custom policy provisioning, enabling the local context-based authorization enforcement and raising the flexibility through dynamic policy-driven access control.

The feasibility and scalability of this approach have been formally and experimentally validated in C0 constrained devices [63, 64] at a laboratory. By means of transport protocol stack, this specifically designed security protocol runs over UDP

ignoring trending CoAP/DTLS standard adoption, but by means of adoption, there is no reference to any use-case scenario pointing to high-technology readiness level values.

Summarizing as shown in Table 6.9, this Hidra approach enables high policy language expressiveness so the enabled tightness is also high, as well as the scalability and the flexibility on the policy. It also enables local context awareness at enforcement time, and it is remarkable for its feasibility in severely constrained CPSs.

### 6.5.9 *Discussion About IoT Taylored Access Control Solutions*

Up to here, access control approaches specifically tailored to IoT applications have been analyzed, and Table 6.10 shows a summary of their main features. Basically, the currently implemented and adopted static and coarsely grained policies to be enforced locally in the CPS are not applicable for service-oriented open scenarios, where operation and management access is by nature dynamic and ad hoc. However, some approaches specifically adapted to IoT are available ready to be validated in real implementations. From the feasibility perspective, one easily adopted option

**Table 6.9** Summary of Hidra analysis highlighting the balance between the tight enforcement and the feasibility in severely constrained CPSs

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| HIDRA | High | High | C0-CN | Yes | UDP | SKC | Yes |

Additionally, it enables local context awareness as well as scalability and flexibility on the policy

**Table 6.10** Summary of the access control models tailored for constrained devices

| | Tightness | | Feasibility | | | | Flexibility |
|---|---|---|---|---|---|---|---|
| | Policy language expressiveness | Local awareness | Platform (C0-CN) | Scalability | Transport layer | Key schema | Policy changes |
| XACML' | Very high | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |
| UCON | High | High | C2-CN | Yes | None | n/a | Yes |
| CapBAC' | High | Low | CN | Yes | HTTP/REST | PKC | Yes |
| DCapBAC | High | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |
| DCAF | Medium | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |
| OSCAR | High | Low | C2-CN | Yes | CoAP/DTLS | PKC | Yes |
| LADON | Low | None | C0-CN | Yes | UDP | SKC | No |
| HIDRA | High | High | C0-CN | Yes | UDP | SKC | Yes |

Analysis is based, firstly, on the effectiveness by means of the expressiveness of the policy and the local context awareness; secondly, on the efficiency conveying the feasibility in different C0-CN constrained device categories, the scalability, the standards adoption in transport layer and cryptographic schema; and finally, flexibility, where policy changes are highlighted.

is a centralized architecture, where an unconstrained central server grants access to requester subjects based on traditional standard mechanisms and protocols such as XACML. This solution, named XACML' in Table 6.10, supports the most expressive policy languages but the granting decision is unaware of the local context in the CPSs. An additional drawback is that, according to [65], it also implies high energy consumption and high network overhead because messages containing requests and authorization responses are transmitted through the network.

From the continuous access control enforcement, the Usage Control model [31] and the attribute-based policy schema [52] enable a completely different alternative solution, which extends traditional unconstrained access control systems. In this approach, the decision to grant access is made before the first access occurs rather than during it, and it enables the continuous protection of the resources during access considering also consumption activities. This solution supports obligations for usage control but does not include a proposal addressing its feasibility in any CPS class.

Analyzing the distributed architecture approaches, a recent alternative to ABAC or RBAC is capability-based access control, which is adequate to IoT in the proposal [53], named CapBAC' in Table 6.10. CapBAC for IoT supports the agile authorization decisions based on the possession and presentation of an unforgeable token denoted by capability [66]. Such a token conveys a subject identifier, the permissions, a validity period, and the authorization chain. The token is represented as an XML schema, and the validated scenarios are implemented in Java language. Consequently, the resource implementation is feasible in small devices but not in constrained CPSs. Furthermore, neither entity authentication nor digital signatures are supported. Nevertheless, CapBAC for IoT is an interesting model.

The concept of capability-based access control has inspired some other approaches specifically defined for constrained CPSs. A fully distributed approach enables the CPSs to make authorization decisions autonomously [54, 67], and it has been successfully validated in smart building scenarios [68]. This proposal, named DCapBAC in Table 6.10, is based on an optimized version of the ECDSA to enable PKC in CPSs. It guarantees E2E authentication, integrity, and non-repudiation without requiring the intervention of any intermediate entity. Concretely, PKC is suitable for large-scale deployments since it enables E2E autonomous session key establishment between peers. The tokens (also referred to as capabilities or tickets) might optionally include additional contextual conditions expressed as type, value, units (TVU) tuples, which are locally checked in the CPS after the capability has been validated.

Still, this solution discloses some drawbacks. First, this option fails to tackle the commissioning of the tokens, namely, the life cycle that involves multiple generations, the delivery conditions and constraints, or the usage and revocation. Furthermore, this approach implies that the subject must retrieve a valid token beforehand from a trusted third party. That is, it requires that, prior to the access attempt, the subject obtains a valid capability. This requirement is neither flexible (because it implies granting coarse-grained permissions) nor dynamic (because it depends on the lifetime of the token). Therefore, the required local conditions are static because there is no token refreshing option and, therefore, it supports changes

neither in local execution context nor in the status of resources in CPSs. Moreover, specific to the policy language, since it does not support expressions, condition checking is limited to matching functions of static values, and obligations are not supported. Finally, successful validation has been conducted in C2 CPSs with 32-bit CPUs, but due to the computational cost and the energy consumption related to the PKC and the length of the capabilities, this approach is not feasible in C0 and C1 CPSs [55, 69].

The delegated CoAP authentication and authorization framework (DCAF) [56] is a similar alternative. This approach proposes the use of a token to distribute preshared keys. That is, in the case of a positive authorization, a handshake is done to establish a secure DTLS channel. Authorization policies convey local conditions that are evaluated as attribute value matching. In addition, instead of the JSON representation for policies, CBOR serialization [58] is adopted, to compact the payloads in the CoAP protocol and, specifically, to compress the length of the tokens and the enclosed policies. However, since CBOR is a general-purpose serialization mechanism, the compression factor is not sufficient for the C0 and C1 CPSs yet. Consequently, it is feasible only in C2 CPSs.

An object security architecture for IoT, named OSCAR, is proposed in [59]. This scalable solution that enables E2E secure access control enforcement relies on PKC, and it supports multicast, asynchronous traffic, and caching. Unfortunately, OSCAR involves an excessive overhead required by DTLS handshakes. In order to overcome such overhead, the delegation of the session key establishment process to an intermediate entity is undesirable from a security point of view. Additionally, since the specification of the access control mechanisms is poorly described, some requirements related to local context awareness or obligations to implement reactive actions are not covered.

Focusing on the access control models which are feasible and adopted in severely constrained devices, Ladon [61] enables authentication, authorization, and secure E2E session establishment based on SKC. However, it supports neither policy provisioning nor accounting functionalities. In fact, Ladon defines a preconfigured static policy specified as the allowed values for subject's role in the CPSs, and it supports the secure transmission of such authenticated role values in the subject's requests. Therefore, Ladon is not a flexible solution and scales poorly.

Finally, there is an innovative access control model named Hidra, which enables high policy language expressiveness so the enabled enforcement tightness is also high, as well as the scalability, the flexibility on the policy, and the traceability. It also enables local context awareness and usage control at enforcement time, and it is remarkable for its feasibility in severely constrained CPSs.

Up to here several approaches have been analyzed. Table 6.11 conveys the tightness, the feasibility, and the flexibility of the aforementioned access control models. To summarize, (1) traditional access control solutions are not feasible in all constrained devices (C0-C2 CPSs) due to their big impact on the performance although they provide the highest effectiveness by means of tightness and flexibility. (2) Recent access control solutions designed for constrained devices can be implemented only in C2 CPSs and lack policy expressiveness in the local authorization

**Table 6.11** Access control models overview categorized by the most suitable constrained device category

| | Tightness | | Feasibility | Flexibility |
|---|---|---|---|---|
| | Expressiveness | Local awareness | Performance impact | Policy changes |
| Traditional AC models | High | High | High | High |
| AC models adapted to C2-CN | Medium | Low | Medium | Medium |
| AC models adapted to C0-CN | Low | Low | Low | Low |

Performance impact limits the feasibility in severely constrained devices, whereas effectiveness is understood as expressiveness of the policy, local awareness, and flexibility decreased with the capabilities of the devices

enforcement. Most of them share the token-driven approach (XACML assertion, capability, token, etc.) in a mixed enforcement architecture enabling an acceptable degree of changes in the policy. (3) Access control solutions currently feasible in C0 and C1 CPSs have been based on authentication and very coarse-grained and static policies, scale badly, and lack a feasible policy-based access control solution. However, an innovative approach named Hidra enables tight and feasible local context-aware enforcement, being flexible and scalable, and includes also accounting and usage control functionalities for any CPS level (C0-C2).

Therefore, there are several solutions that can be adapted to different scenarios and provide expressive, policy-based fine-grained authorization services in the envisioned scenarios of constrained but manageable sensor networks, which need to be validated in real use-case scenarios to raise their technology readiness levels.

## 6.6 Conclusions and Future Work

This chapter conveys an overview of current security solutions, specifically access control solutions, on CPSs implemented in constrained devices, and accessible as things in an IoT.

Concretely, the main contribution of this chapter is an exhaustive analysis of eight different access control models suitable for IoT, which being tailored for constrained devices are assessed under two key criteria: tightness and feasibility. The analysis of such eight innovative adjusted approaches evaluates in what degree these models do support fine-grained and tight security policy enforcement in severely constrained devices, since the currently implemented static and coarsely grained policies to be enforced locally in the CPS are not applicable for service-oriented open scenarios where operation and management access is by nature dynamic and ad hoc.

In fact, the aforementioned CPSs are the smarter the better, and they contribute to a higher visibility in field activity, so they enable enhanced decision making and more adaptive behavior of systems and ecosystems that integrate pervasive and ubiquitous ICT technologies. Large-scale IoT applications involve a massive deployment of sensors and actuators, which are cheap, so they are implemented in a range of constrained device sensors and CPSs classified from C0 to CN according

to [69]. Moreover, depending on the use-case and location, they may require power autonomy and, therefore, demand low power consumption mechanisms. Additionally, beyond the behavior of the CPSs as measurement publishers, more intelligent IoT scenarios are also envisioned, in which CPSs integrate tiny information servers that support smarter and more manageable applications. In such use-cases, users directly query the tiny servers, through a secure E2E communication. However, in such IoT applications, security (and more specifically, access control) remains an insufficiently solved need.

The security solutions to meet the aforesaid security requirements are formulated as a set of features as confidentiality, authentication, integrity, authorization, non-repudiation, and availability. These security features rely, on the one hand, on a combination of cryptographic mechanisms that present a high computational cost, such as block ciphers, hash functions, or signature algorithms, which require a proper handling of cryptographic keys. And, on the other hand, these features are implemented through non-cryptographic mechanisms for authentication, authorization, and other security policy enforcement functions. For the latter purpose, the security policy has to be adequately codified and an enforcement engine is required, which might be also ready to accept policy changes during life cycle of the device.

In the context of the constrained devices required in large-scale deployments, the security must focus not only on the required security services, but also on how these services are realized in the overall system and how the security functionalities are executed. How these access control models protect the confidentiality and the integrity of the data exchanged with constrained devices, as well as the authentication and authorization enforcement of any endpoint accessing data in the constrained device are analyzed, under two main criteria: feasibility and least privilege principle adherence.

For that end, firstly, the features and constraints related to the considered constrained devices are described. It is relevant to consider how these constraints impact on the feasibility of any access control model. The constrained device classification depending of the resource capabilities such as limited computing capacity, little memory, insufficient network bandwidth, and often limited battery power helps to understand the limitations toward the traditional access control solutions. Additional difficulties of the constrained networks augment the limitations derived from resource scarcity.

On the other hand, when designing and applying access control models, the tightness of the enforcement is related to the granularity of the policy, but there is always a trade-off between expressiveness of the policy and practical feasibility. Moreover, an access control model should cover not only a feasible security policy definition and enforcement, but also a way to enable the policy changes so needed in open and dynamic environments. For that purpose, the chosen security architecture has inherent benefits and drawbacks whether it is either centralized or distributed. Additionally, it is analyzed how they tackle with the local context awareness. Finally, since cryptographic mechanisms support the implementation of the aimed security features, less exigent symmetric key schemas' weakness and strengths are compared to that of public key schemas, which present better scalability.

Secondly, access control foundations are presented where the policy-driven paradigm enables more flexible and scalable solutions. Additionally, an overview of the existing access control models is conveyed, where the most extended is the ABAC model but the alternative UCON proposes the policy enforcement not only prior to the access but also during the access. Although these two models are not feasible directly, they can be considered as the foundations for any access control model optimized for constrained CPSs. Thirdly, an overview of existing policy languages is conveyed pointing out the contribution of the main features to the expressiveness and, consequently, to the tightness of the enforcement and adherence to the least privilege principle.

Finally, as traditional access control models are not feasible, new access control models specifically optimized for IoT environments have been presented. Among them, eight different approaches are conveyed as appropriate to be deployed in several scenarios with different constrained device classes. First, the so extended XACML has been adapted to IoT as a centralized ABAC implementation. Alternatively, UCON adapted to IoT is described, which proposes a distributed access control enforcement model not only before the first access but also during it. And the third most differentiated approach is the capability-based access control with two proposals, third and fourth, which convey progressive distribution of the authorization decision and enforcement based on capabilities distributed based on a PKC schema. The fifth approach proposes a token-based access control that can be seen similar to previous ones but fully compliant with constrained environment standards as CoAP and DTLS. The sixth is OSCAR, which conveys an object security architecture relying on PKC and DTLS. The seventh proposal named Ladon that is absolutely optimized and feasible in severely constrained scenarios proposes a RBAC authorization enforcement based on a SKC schema. Finally, the eighth approach named Hidra that is feasible in severely constrained devices proposes a local context-based access control model enabling a tight policy enforcement, through a specific purpose policy language, as well as the dynamic policy provisioning and accounting mechanisms for further tracking and auditing.

To conclude, there is an innovative suitable solution that can provide expressive, policy-based, fine-grained authorization services in the envisioned scenarios of severely constrained but manageable CPS networks. Moreover, currently existing IoT tailored approaches share the concept of mixture of the authorization decision. That is, most of them involve a trusted third party which enforces a preliminary authorization and then issues a kind of token (XACML assertion, capability, token, etc.) enclosing granting data. Among other enclosed data, most of them propose to enclose required keying data and also a locally enforceable policy limiting their expressiveness to a short set of attributes. They differ basically in the representation of the token, the policy, and the relationships with the issuer. Derived from these differences, some of them are more suitable or not in the range of constrained device classification. Therefore, further research is required in the validation scenarios for the expressiveness and feasibility of the local context-aware policy definition, provisioning, and enforcement.

## References


1. Gegick M, Barnum S Least privilege. https://www.us-cert.gov/bsi/articles/knowledge/principles/least-privilege
2. Sfar AR, Natalizio E, Challal Y, Chtourou Z (2017) A roadmap for security challenges in the internet of things. Digital Communications and Networks
3. Vermesan O, Friess P, Guillemin P, Gusmeroli S, Sundmaeker H, Bassi A, Jubert IS, Mazura M, Harrison M, Eisenhauer M et al (2011) Internet of things strategic research roadmap. Internet Things-Global Technol Soc Trends 1(2011):9–52
4. Schaller RR (1997) Moore's law: past, present, and future. IEEE Spectr 34(6):52–59. https://doi.org/10.1109/6.591665 URL https://doi.org/10.1109/6.591665
5. Gargini P Itrs past, present and future. https://spcc2016.com/wp-content/uploads/2016/04/02-01-Gargini-ITRS-2.0-2.pdf
6. Waldrop MM The chips are down for moore's law. http://www.nature.com/news/the-chips-are-down-for-moore-s-law-1.19338
7. Montenegro G, Kushalnagar N, Hui J, Culler D (2007) Transmission of IPv6 packets over IEEE 802.15.4 networks. Technical Report. RFC 4944, Internet Engineering Task Force
8. Vasseur J (2014) Terms used in routing for low-power and lossy networks. RFC 7102, RFC Editor. URL http://www.rfc-editor.org/rfc/rfc7102.txt. http://www.rfc-editor.org/rfc/rfc7102.txt
9. Kovatsch M (2013) Coap for the web of things: From tiny resource-constrained devices to the web browser. In: Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, UbiComp '13 Adjunct, pp 1495–1504. ACM, New York. doi: https://doi.org/10.1145/2494091.2497583. URL http://doi.acm.org/10.1145/2494091.2497583
10. Shelby Z, Hartke K, Bormann C (2014) The constrained application protocol (coap). RFC 7252, RFC Editor. http://www.rfc-editor.org/rfc/rfc7252.txt
11. Kim E, Kaspar D, Gomez C, Bormann C (2002) Problem statement and requirements for ipv6 over low-power wireless personal area network (6lowpan) routing. RFC 6606, RFC Editor. URL https://www.rfc-editor.org/rfc/rfc6606.txt
12. Roman R, Zhou J, Lopez J (2013) On the features and challenges of security and privacy in distributed internet of things. Comput Netw 57(10):2266–2279 https://doi.org/10.1016/j.comnet.2012.12.018. Towards a Science of Cyber SecuritySecurity and Identity Architecture for the Future Internet
13. Sicari S, Rizzardi A, Grieco L, Coen-Porisini A (2015) Security, privacy and trust in internet of things: the road ahead. Comput Netw 76:146–164. http://dx.doi.org/10. 1016/j.comnet.2014.11.008
14. Yan Z, Zhang P, Vasilakos AV (2014) A survey on trust management for internet of things. J Netw Comput Appl 42:120–134. https://doi.org/10.1016/j.jnca.2014.01.014
15. Rivera J (2015) Survey analysis: the internet of things is a revolution waiting to happen. http://www.gartner.com/newsroom/id/2977018. http://www.gartner.com/document/2965320
16. Chan H, Perrig A, Song D (2003) Random key predistribution schemes for sensor networks. In: Security and privacy, 2003. Proceedings. 2003 symposium on, pp 197–213. IEEE
17. Zhu S, Xu S, Setia S, Jajodia S (2003) Lhap: a lightweight hop-by-hop authentication protocol for ad-hoc networks. In: Distributed computing systems workshops, 2003. Proceedings. 23rd international conference on, pp 749–755. doi: https://doi.org/10.1109/ICDCSW.2003.1203642
18. Wang H, Li Q (2006) Distributed user access control in sensor networks. In: Distributed computing in sensor systems. Springer, Berlin, pp 305–320
19. Goovaerts T (2011) Distributed authorization middleware for service-oriented architectures (gedistribueerde autorisatiemiddleware voor service-georiënteerde architecturen)
20. Tschofenig H, Arkko J, Thaler D, McPherson D (2015) Architectural considerations in smart object networking. RFC 7452, RFC Editor
21. Hankerson D, Menezes AJ, Vanstone S (2003) Guide to elliptic curve cryptography. Springer-Verlag New York, Inc., Secaucus

22. Han W, Lei C (2012) A survey on policy languages in network and security management. Comput Netw 56(1), 477–489. https://doi.org/10.1016/j.comnet.2011.09.014. URL http://www.sciencedirect.com/science/article/pii/S1389128611003562
23. Moore B (2003) Policy core information model pcim extensions. RFC 3460, RFC Editor
24. Dayal U, Hanson E, Widom J (1994) Active database systems. Technical Report 1994–20, Stanford Infolab. URL http://ilpubs.stanford.edu:8090/54/
25. Loscocco PA, Smalley SD, Muckelbauer PA, Taylor RC, Turner SJ, Farrell JF (1998) The inevitability of failure: the flawed assumption of security in modern computing environments. In: In Proceedings of the 21st national information systems security conference, pp 303–314
26. Harrison MA, Ruzzo WL, Ullman JD (1976) Protection in operating systems. Commun ACM 19(8):461–471
27. Sandhu R, Ferraiolo D, Kuhn R (2000) The nist model for role-based access control: Towards a unified standard. In: Proceedings of the fifth ACM workshop on role-based access control, RBAC '00, pp 47–63. ACM, New York. doi: https://doi.org/10.1145/344287.344301. URL http://doi.acm.org/10.1145/344287.344301
28. Hu VC, Ferraiolo D, Kuhn R, Friedman AR, Lang AJ, Cogdell MM, Schnitzer A, Sandlin K, Miller R, Scarfone K, Hu VC, Ferraiolo D, Kuhn R, Friedman AR, Lang AJ, Cogdell MM, Schnitzer A, Sandlin K, Miller R, Scarfone K, Cybersecurity S (2013) Guide to attribute based access control (abac) definition and considerations (draft)
29. Yuan E, Tong J (2005) Attributed based access control (ABAC) for Web services. In: IEEE International Conference on Web Services (ICWS). Institute of Electrical & Electronics Engineers (IEEE). doi: https://doi.org/10.1109/icws.2005.25. URL https://doi.org/10.1109/ICWS.2005.25
30. Hilty M, Pretschner A, Basin D, Schaefer C, Walter T (2007) A policy language for distributed usage control. In: European symposium on research in computer security. Springer, New York, pp 531–546
31. Park J, Sandhu R (2004) The uconabc usage control model. ACM Trans Inf Syst Secur 7(1):128–174. https://doi.org/10.1145/984334.984339 URL http://doi.acm.org/10.1145/984334.984339
32. Parducci B (2013) Extensible access control markup language (xacml) version 3.0, standard. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html. URL http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
33. Guoping Z, Wentao G (2011) The research of access control based on ucon in the internet of things. J Softw 6(4):724–731
34. Cantor S, Kemp J, Philpott R, Maler E (2005) OASIS: security assertions markup language, SAML 2.0. http://saml.xml.org/saml-specifications
35. Hardt D (2012) The oauth 2.0 authorization framework. RFC 6749, RFC Editor (2012). URL http://www.rfc-editor.org/rfc/rfc6749.txt. http://www.rfc-editor.org/rfc/rfc6749.txt
36. Damianou N, Dulay N, Lupu E, Sloman M (2001) Policies for distributed systems and networks: international workshop, POLICY 2001 Bristol, UK, January 29–31, 2001 Proceedings, chapter. The ponder policy specification language, pp 18–38. Lecture notes in computer science. Springer Berlin Heidelberg, Berlin, Heidelberg. doi: https://doi.org/10.1007/3-540-44569-2.2. URL https://doi.org/10.1007/3-540-44569-2_2
37. Kagal L, Finin T, Joshi A (2003) A policy language for a pervasive computing environment. In: Policies for distributed systems and networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on, pp 63–74. doi: https://doi.org/10.1109/POLICY.2003.1206958
38. Berners-Lee T, Connolly D (2008) Notation3 (n3): a readable rdf syntax. W3c team submission, W3C. URL http://www.w3.org/TeamSubmission/n3/
39. Jajodia S, Samarati P, Subrahmanian VS (1997) A logical language for expressing authorizations. In: IEEE symposium on security and privacy, pp 31–42. IEEE computer society. URL http://dblp.uni-trier.de/db/conf/sp/sp1997.html#JajodiaSS97
40. Cranor L, Langheinrich M, Marchiori M (2002) A p3p preference exchange language 1.0 (appel 1.0). World Wide Web Consortium, Working Draft WD-P3P-preferences-20020415

41. Ashley P, Hada S, Karjoth G, Powers C, Schunter M (2003) Enterprise privacy authorization language (EPAL). Technical report. IBM Research, Rschlikon
42. Levy HM (1984) Capability-based computer systems. Butterworth-Heinemann, Newton
43. Sloman M (1994) Policy driven management for distributed systems. J Netw Syst Manag 2(4):333–360. https://doi.org/10.1007/BF02283186 URL https://doi.org/10.1007/BF02283186
44. at W3C, P.L.I.G Review of policy languages and frameworks. https://www.w3.org/Policy/pling/wiki/PolicyLangReview
45. Bechhofer S, van Harmelen F, Hendler J, Horrocks I, McGuinness DL, Patel-Schneider PF, Stein LA (2004) OWL web ontology language reference. Technical report, W3C, https://www.w3.org/TR/owl-ref/
46. Candan KS, Liu H, Suvarna R (2001) Resource description framework: metadata and its applications. SIGKDD Explor Newsl 3(1):6–19. https://doi.org/10.1145/507533.507536 URL https://doi.acm.org/10.1145/507533.507536
47. Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosofand B, Dean M (2004) SWRL: a semantic web rule language combining OWL and RuleML. W3C member submission. URL http://www.w3.org/Submission/SWRL/. Last access on Dec 2008 at: http://www.w3.org/Submission/SWRL/
48. specs@openid.net (2007) Openid authentication 2.0 – final. URL http://openid.net/specs/openid-authentication-2_0.html
49. Lindemann DR, Baghdasaryan D, Tiffany E, Alliance F (2014) Fido uaf protocol specification v1.0. https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-protocol-v1.0-ps-20141208.html
50. Srinivas S, Balfanz D, Tiffany E, Czeskis A, Alliance F (2015) Universal 2nd factor u2f overview. https://fidoalliance.org/specs/fido-undefined-undefined-ps-20150514/fido-u2f-overview-v1.0-undefined-ps-20150514.html
51. Seitz L, Selander G, Gehrmann C (2013) Authorization framework for the internet-of-things. In: 2013 IEEE 14th international symposium on "a world of wireless, mobile and multimedia networks" (WoWMoM), pp 1–6. doi: https://doi.org/10.1109/WoWMoM.2013.6583465
52. Su Z, Biennier F (2013) On attribute-based usage control policy ratification for cooperative computing context. CoRR abs/1305.1727. URL http://arxiv.org/abs/1305.1727
53. Gusmeroli S, Piccione S, Rotondi D (2013) A capability-based security approach to manage access control in the internet of things. Math Comput Model 58(56):1189–1205 https://doi.org/10.1016/j.mcm.2013.02.006. The Measurement of Undesirable Outputs: Models Development and Empirical Analyses and Advances in mobile, ubiquitous and cognitive computing
54. Hernández-Ramos JL, Jara AJ, Marın L, Skarmeta AF (2013) Distributed capability-based access control for the internet of things. J Internet Serv Info Secur (JISIS) 3(3/4):1–16
55. Ramos JLH, Jara AJ, Marin L, Gomez AFS (2016) Dcapbac: embedding authorization logic into smart things through ecc optimizations. Int J Comput Math 93:345–366
56. Gerdes S, Bergmann O, Bormann DC (2005) Delegated CoAP Authentication and Authorization Framework (DCAF). Internet-Draft draft-gerdes-ace-dcaf-authorize-04, Internet Engineering Task Force. Work in Progress
57. Bergmann O Eclipse tinydtls. https://projects.eclipse.org/projects/iot.tinydtls
58. Bormann C, Hoffman P (2013) Concise binary object representation (cbor). RFC 7049, RFC Editor
59. Vucinic M, Tourancheau B, Rousseau F, Duda A, Damon L, Guizzetti R (2015) OSCAR: object security architecture for the internet of things. Ad Hoc Netw 32, 3–16. doi: https://doi.org/10.1016/j.adhoc.2014.12.005. URL http://www.sciencedirect.com/science/article/pii/S1570870514003126. Internet of Things security and privacy: design methods and optimization

60. Kothmayr T, Schmitt C, Hu W, Brünig M, Carle G (2013) Dtls based security and two-way authentication for the internet of things. Ad Hoc Netw 11(8):2710–2723. https://doi.org/10.1016/j.adhoc.2013.05.003 URL https://doi.org/10.1016/j.adhoc.2013.05.003
61. Astorga J, Jacob E, Huarte M, Higuero M (2012) Ladon: end-to-end authorisation support for resource-deprived environments. IET Inf Secur 6(2):93–101
62. Astorga J, Toledo N, Jacob E, Higuero M (2013) Taxonomy of security protocols for wireless sensor communications. In: Security for multihop wireless networks. CRC Press, Taylor & Francis Group, Boca Raton
63. Uriarte M, Astorga J, Jacob E, Huarte M, Carnerero M (2017) Expressive policy based access control for resource-constrained devices. IEEE Access PP(99):1. https://doi.org/10.1109/ACCESS.2017.2730958
64. Uriarte M, Astorga J, Jacob E, Huarte M, Carnerero M (2017) Feasbility assessment of a finegrained access control model on resource constrained sensors. In: Proceedings of the XIII Jornadas de Ingenier'ıa Telema'tica (JITEL 2017)
65. Shnayder V, Hempstead M, Rong Chen B, Allen GW, Welsh M (2004) Simulating the power consumption of large-scale sensor network applications. In: Proceedings of the 2nd international conference on embedded networked sensor systems (SenSys '04). ACM, New York, pp 188–200
66. Dennis JB, Van Horn EC (1996) Programming semantics for multiprogrammed computations. Commun ACM 9(3):143–155. https://doi.org/10.1145/365230.365252 URL https://doi.acm.org/10.1145/365230.365252
67. Skarmeta AF, Hernandez-Ramos JL, Moreno MV (2014) A decentralized approach for security and privacy challenges in the Internet of Things. In: Internet of things (WF-IoT), 2014 IEEE World Forum on, pp 67–72 (2014). doi: https://doi.org/10.1109/WF-IoT.2014.6803122
68. Hernndez-Ramos JL, Moreno MV, Bernab JB, Carrillo DG, Skarmeta AF (2015) SAFIR: secure access framework for IoT-enabled services on smart buildings. J Comput Syst Sci 81(8), 1452–1463. doi: https://doi.org/10.1016/j.jcss.2014.12.021. URL http://www.sciencedirect.com/science/article/pii/S0022000014001858
69. Bormann C, Ersue M, Keranen A (2014) Terminology for constrained-node networks. RFC 7228, RFC Editor. http://www.rfc-editor.org/rfc/rfc7228.txt

# Chapter 7
# Security Challenges and Concerns of Internet of Things (IoT)


**Aniruddha Bhattacharjya, Xiaofeng Zhong, Jing Wang, and Xing Li**

**Abstract** The Internet of Things (IoT) signifies the interconnection of exceedingly heterogeneous networked entities, for instance, sensors, actuators, smart phones, etc. In accord with concrete functions, the network structure of the IoT is divided into three hierarchies: the bottom hierarchy is the sensing equipment for information acquisition; the middle hierarchy is the network for data transmission, whereas the top hierarchy is intended for applications and middleware. The uniqueness of the IoT proclaims new challenges to security requirements, dissimilar from previous technology trends. Moreover, to guarantee resilience, fail-over and recovery mechanisms must be provided to uphold operations under failure or attacks, and to return to normal operations (failure/attack mitigation). To uphold the end-to-end method, the gateway requirements to endure invisible to the communicating endpoints. The Constrained Application Protocol (CoAP) is an ideal protocol, for being used with constrained devices and low-power networking. To give more security, to the major UDP (User Datagram Protocol) well-known applications, for instance, Voice over IP/Session Initiation Protocol (VoIP/SIP), Datagram Transport Layer Security (DTLS) can run on top of UDP instead of TCP (Transmission Control Protocol). In our research, we have found that hybrid RSA (Rivest–Shamir–Adleman) algorithm can be a good one with efficiency, more security, and more privacy protected way and can work for end-to-end encryption requirements for future Internet of Everything (IoE). In general, future researches in the security issues of the IoT would mostly quintessence on the following characteristics, the open security system, individual privacy protection mode, terminal security function, related laws for the security of the IoT, etc. It is unquestionable that the security of the IoT prerequisites a series of policies, laws, and regulations, perfect security management system for mutual collocation.

A. Bhattacharjya (✉) · X. Zhong · J. Wang · X. Li (✉)
Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China
e-mail: li-an15@mails.tsinghua.edu.cn; zhongxf@tsinghua.edu.cn; wangj@tsinghua.edu.cn; xing@cernet.edu.cn



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_7

## 7.1 Introduction

The Internet of Things (IoT) signifies [1–20] the interconnection of exceedingly heterogeneous networked entities, for instance, sensors, actuators, smart phones, etc. In accord with concrete functions, the network structure of the IoT is divided into three hierarchies: the bottom hierarchy is the sensing equipment for information acquisition; the middle hierarchy is the network for data transmission, whereas the top hierarchy is intended for applications and middleware. The IPv6 and web services as major building blocks for IoT applications have formed a homogeneous protocol ecosystem, letting simple integration of IoT devices in a Low-power and Lossy Network (LLN) with Internet hosts. The uniqueness of the IoT proclaims new challenges to security requirements dissimilar from previous technology trends. Moreover, to guarantee resilience, fail-over and recovery, mechanisms must be provided to uphold operations under failure or attacks, and to return to normal operations (failure/attack mitigation). We can choose Datagram Transport Layer Security (DTLS) as our security protocol that depends on this protocol stack. Alike the security needs in traditional networks, such as the Internet, we can think about three security goals for IoT scenario [1–20]:

*Authenticity*: Receivers of a message can recognize their communication companions and can identify if the sender information has been forged.
*Integrity*: Communication companions can identify modifications to a message for the duration of transmission.
*Confidentiality*: Attackers cannot get information about the matters of a secured message.

DTLS fulfills these goals. The authentication is accomplished during a fully authenticated DTLS handshake and depends on an exchange of X.509 certificates comprising Rivest–Shamir–Adleman (RSA) keys. An unconstrained network (UCN) is classically signified by the Internet, while the loT comprising of a low-power wireless personal area network (LoWPAN) signifies the constrained domain. An loT gateway placed on the edge among the constrained network (CN) and the UCN adapts the communication among these two domains. Its role typically encompasses the adaptation between dissimilar protocol layer implementations. Also called a border router, it carries out protocol translations vis-a-vis end-to-end loT security. The gateway is usually an unconstrained device, which can be used for scaling down the functionalities from the UCN to the CN domain. The gateway can be used for handling security settings in peripheral constrained networks. To uphold the end-to-end method, the gateway requirements to endure invisible to the communicating endpoints. A node on the UCN can be either Hypertext Transfer Protocol (HTTP) enabled or only Constrained Application Protocol (CoAP) enabled. The communication protocols existing or being designed at the IEEE and IETF now empower a standardized protocol stack. The mechanisms founding this stack must thus empower Internet communications encompassing constrained sensing devices, while coping with the necessities of low-energy communication environments and

the aims and the lifetime of IoT applications [21–40]. In order to talk this issue for the IoT, the IETF has started the Constrained RESTful Environments (CoRE) working group, which aims at standardizing the incorporation of constrained devices with the Internet at service level. The CoRE proposal aims to permit the integration of constrained devices with the Internet, at service level. CoRE proposes the use of CoAP in constrained devices, a specialized RESTful Web transfer protocol. CoAP is a specialized web transfer protocol aimed to be used by constrained devices in loT machine-to-machine (M2M) applications. It is responsible for a client/server interaction model between application endpoints and comprises the same key functionalities of HTTP. So, CoAP can be easily interfaced with HTTP, resulting the web integration simplified while also guaranteeing M2M critical necessities, for example, built-in discovery, simplicity, multicast support, and low overhead. Yet, application layer protocols recurrently delegate security techniques to the transport layer, which benefits in attaining end-to-end security. The overhead caused by this security mechanism is very significant to the overall system performance. One such protocol is DTLS, which furthermore has inbuilt binding within CoAP. Security is fundamental for the application areas. We should take care of the basic security services, for example, confidentiality, authentication, and freshness of secret keys between two communicating entities. Information exchanged in the network requisite to be protected end-to-end. To cope with these security necessities, CoAP offers DTLS and when DTLS NoSec mode is selected, the CoAP communication could be secured using IP Security (IPSec) at the network layer in an LLN. Nevertheless, DTLS was not intended for lossy networks and constrained devices, it has appeared as a vital candidate to deliver security in loT. Nevertheless, it cannot be employed as it is, ever since it is well-thought-out to be too heavy for using in constrained environments and networks such as loT. Thus emerged numerous lightweight implementations of DTLS are there now for use in loT. Lightweight DTLS implementation could depend on employing any of the following techniques:

1. Pre-shared key (PSK)
2. Raw public key
3. Certificates

The *CoAP protocol* defines bindings to *DTLS (Datagram Transport Layer Security)* to secure CoAP messages, together with a few mandatory minimal configurations suitable for constrained environments. The acceptance of DTLS implies that security is reinforced at the transport layer, rather than being designed in the context of the application layer protocol. DTLS provides promises in terms of confidentiality, integrity, authentication, and non-repudiation for application layer communications using CoAP.

In the last section of this chapter, we have highlighted some case studies and open research issues.

## 7.2 Internet of Things Architectures, Properties, and Security Requirements

### *7.2.1 Architectures and Basic Properties*

With the contextual features of Internet, the IoT [29–33, 35–45] is an emergent technology uniting EPC standard, wireless communications technology, Radio Frequency Identification (RFID) technology, and so on, to empower human to commendably solve various defies of modern society. IoT brings together and processes detailed information consisting of events and environments, by use of billions of connected things, making our life more comfortable, more productive, safer, and healthier. To explore IoT's hidden prospectives, to address many global complications, for example, energy scarcity, pollution, food, climate change, and water, along with the challenges of transportation, urbanization, and healthcare, the International Telecommunication Union (ITU) is making the IoT standardized for several years in the Telecommunication Standardization Sector (ITU-T). *ITU-T Study Group 20* was formed in recent times, to further endorse coordinated advancement of global IoT technologies, services, and applications. M2M communication technologies deliver an efficient, reliable, and secure communication platform for the almost all of 50 billion IoT devices, that are anticipated to be linked to the succeeding generation (recognized as 5G) mobile networks in 2020 and beyond. IoT is not only a confined infrastructure for interconnecting things only inside a locality (e.g., a building, an enterprise, or a city), but a global infrastructure to connect things by use of interoperable underlying communication networks. "Overview of the Internet of Things" (ITU-T Y.2060), ratified in 2012, has been enriched by a number of recommendations on IoT common framework, capabilities, use cases, and necessities. In ITU-T Y.2060, the thing has been well defined as an object of the physical world (physical thing) or the information world (virtual thing), which can be totally able to be identified and integrated into communication networks. Figure 7.1 depicts the IoT reference model detailed in ITU-T Y.2060. This layered reference model shows us a generic and universal model, with the critical functions and abilities of the IoT architecture. It has a great advantage of decreasing the implementation difficulties and endorses interoperability amid numerous IoT applications and communication technologies. The IoT reference model comprises of four horizontal layers and the common management and security capabilities allied with all layers. The application layer is the topmost layer that comprises numerous IoT applications, e.g., smart grid, intelligent transport systems, e-health, and smart home. The service and application support layer is the second layer, which comprises generic support capabilities along with application-specific support capabilities. The generic support capabilities are common abilities relevant to many applications, while the application-specific capabilities work for a particular application's necessities as their names denote. The network layer comprises the networking and transport capabilities. The networking capabilities execute the connection of things to networks and maintain that connectivity. They

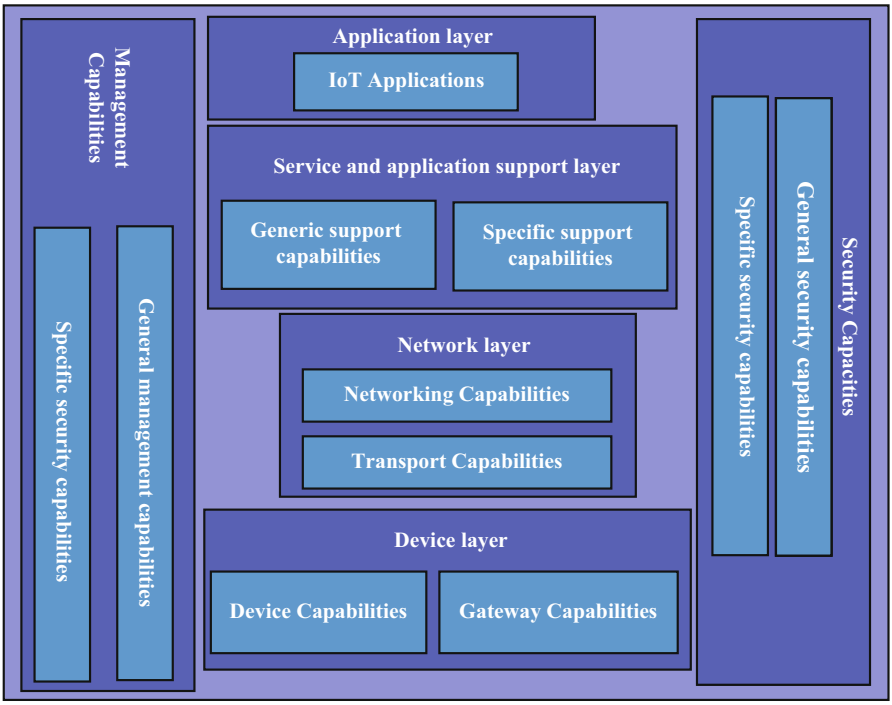



**Fig. 7.1** Internet of Things (IoT) reference model

comprise functions for resource allocation, routing, mobility management, access control, etc.

Likewise, the transport capabilities comprise functions for transporting IoT application data along with control and management instructions. The device layer at bottom comprises a collection of device capabilities and gateway capabilities. The device capabilities empower things to interact with a network straight or via a gateway. They are comprised of ubiquitous sensor networking functions. Likewise, the gateway capabilities comprise privacy protection, security, and protocol translation functions to allow resource-constrained IoT devices empowered with heterogeneous wireless technologies, such as Zigbee, Bluetooth, and WiFi, to be connected securely through a network. Management and security capabilities are also considered as generic and specific capabilities. The generic management capabilities comprise device management functions such as software update, network topology management, status monitoring and control, and traffic, congestion control, and remote activation. The generic security capabilities comprise integrity protection, privacy protection, access control, authentication, confidentiality, and authorization, etc.

The essentials of IoT security [29–33, 35–44] include information sensing with high safety, trustworthy data transfer and information control with high safety. The

security system of the IoT can be classified into three layers, *the Sensor Layer Security, the Network Layer Security, and the Application Layer Security*. *Firstly*, any object in this earth is having connection to the Internet. So, it is well understood that nodes will communicate effortlessly with each other. *Secondly*, sensing at any time anywhere in all place like an all-round sensing, resulting in identification of any object connected in IoT automated, no manual intervention is needed. *The third* is intelligent processing. Intelligence control, self-feedback, and automation, etc., characterize the intelligent processing. This security framework is depicted in Fig. 7.2. In general, we always have to take care about three characteristics in IoT. The *first one* is entirely perception. To make clearer, to gain access to the information of object anywhere at any time by use of various means, like RFID, sensors and two-dimensional code. The *second one* is reliable delivery. To make clearer, it is to send information of the object correctly at real time through incorporating

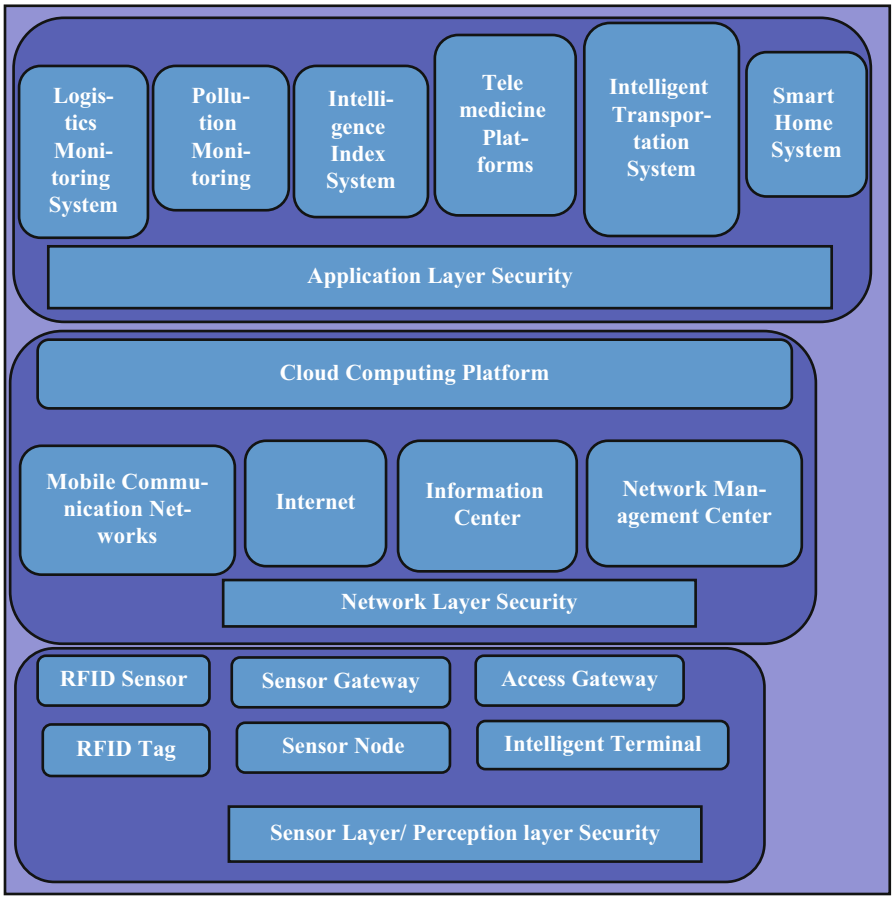



**Fig. 7.2** Security framework of IoT

telecommunication network and Internet. The *third one* is intelligent processing, analyzing, and processing massive data and executing intelligent governing power on objects by the usage of cloud computing, fuzzy recognition, and other intelligent computing techniques. The most frontend layer is Sensor Layer or Perception Layer, which is mainly responsible for information collection and so it is well understood that it has one of the most significant roles in the security of IoT.

So, now let us have some highlights on Security issues in sensor layer. If we consider the case of traditional network, sensor nodes in IoT positioned in an unattended environment, there are some new characteristics in sensor network.

1. Wireless link signal strength is very feeble

Sensor nodes spread data to each other primarily by wireless network and most of them can work well in longtime environments and with low-power environment. The disturbing waves usually affect the wireless communication's signal. So, it is obligatory to not to transfer information by wireless network.

2. Node is visible

In the wireless data communication, hidden terminal and exposed terminal problems are most prominent problems, as wireless channel is an open and shared channel. For better understanding, let us consider an example, when we use RFID technology in sensor layer, the object which embedded an RFID chip will be censored not only by its owner but also by others. So this way, we can understand that the sensor node is the best place for all kind of attackers.

3. The network topology is dynamic

Locations of IoT node frequently change from one place to another. In comparison with traditional Transmission Control Protocol (TCP)/IP network, all network monitoring technologies or cyber defense technologies have to deal with more complex network data, more exactingly real-time demand in the scenarios when 50 billion IoT devices will be connected.

4. Computing capacity, storage capacity, and energy are limited

Typically, IoT node is a product of low-power consumption. Most vulnerable issues are that their computing capacity, storage capacity, and energy are limited. So, it is well understood that our present security technologies of traditional network cannot shift to IoT effortlessly.

So, now let us have some highlights on security technology in sensor layer.

1. Encryption mechanism

Point-to-point encryption and end-to-end encryption are two uppermost forms of cryptographic applications in traditional network. From the IoT framework, generally it can be seen that, the node of sensor layer, is low speed CPU, for instance, single chip system. So, for good security, we need to use large storage and high power for Encryption and Decryption but here we cannot use large storage and high power. So, Encryption technique in IoT should be very much lightweight.

2. Access control

Access control mechanism in IoT is very special and differs than normal networks. In our TCP/IP network, a "person" used to give approval to access the system but in IoT, it is "machine." So, it prerequisites to assign and transfer sharing data in a self-determined method between node and node.

3. Authentication mechanism of nodes

At receiver end, the authentication mechanism is used to make sure of the real identity of sender and make sure whether the data is altered for the duration of the transmission. It is very obligatory, for IoT architecture, to make sure that the true node is working, Encryption mechanism can make the data confidential by encoding the data, and it can stop intruder from stealing and altering crucial information by use of data encryption.

In other opinion, sometime we say that the Sensor Layer as the Perception Layer. So, the functions are totally same, it is like another name only. So, the perception layer is primarily responsible to capture and gather, distinguish, and identify objects' information in physical world. The layer consists of laser scanner, cameras, GPS, sensors, RFID tags, literacy device, and so on.

The second layer is the network layer as shown in Fig. 7.2. This layer is used to transmit and process information acquired by the perception layer or sensor layer. Also, this layer is responsible for delivering reliable communication support to the application layer.

The top level is the application layer as shown in Fig. 7.2. This layer is used to process intelligently massive amount of data, data accumulated from numerous sources with various types and interactive display. The layer uses cloud computing, data mining, middleware business management, and so on, for the control and management of objects' information. We have to look for very coordinated association of the information technology and the industry-specific technology for the upcoming and development of the application layer.

Now, at the time of building the security architecture of the IoT, which is used to resolve the security difficulties, now we are facing it from the bottom layer to the top one of the IoT system. Some of the most concerning security problems among the security difficulties are information acquisition security, information transmission security, information processing security, physical security, and so on. So, when we are designing the security architecture, we have to take care about vulnerabilities in every layer. Figure 7.3 depicts one kind of ideal security architecture of the IoT. As explained in Fig. 7.3, the whole system consists of four layers, which work layer-wise. It works for the physical security of terminal equipments positioned in perception layer and local data storage, the protection of wireless transmission of sensor networks, the security of computer networks and mobile communication transmission, and the data service security on application layer.

Ever since the terminal equipment, like RFID tags, being used for identifying entities and all sorts of low-cost sensors, being used to observe objects' status modification or alteration positioned at the perception layer, is mostly restricted by

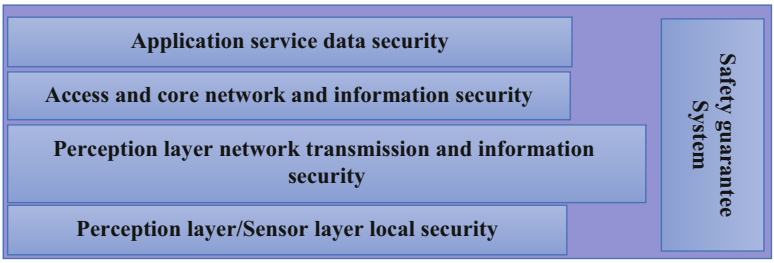



**Fig. 7.3** Security architecture of the IoT

the constrained computing resources and mass positioning but with unverified status in positioning environment, those terminal ends are highly in danger to several kinds of attacks.

The network layer security as per its function is categorized into two types, based on the access layer and the core layer transmission. The core network transmission security problem has a complete security protection ability due to its traditional benefits of network information safety. It also has the traditional network security dangers and defenselessness. Moreover, quantitative scale of nodes positioned in the IoT is gigantic, which an attacker can explode very easily to initiate a denial of service (DoS) attack and block network finally. On the contrary, the access layer offers access to heterogeneity and it yields foremost security vulnerability owing to dissimilar media switching technologies and the location management technology. It has wireless or wired multiple access methods. Furthermore, the openness of wireless interface in wireless mobile communication transmission offers malicious individuals with tapping wireless channel, along with that gives chance for capturing even deleting, inserting, retransmitting, and modifying messages communicated through radio interface with the intention of fake user identification or for identification of deceived server. There are severe potential reasons for privacy leakage of information due to different requirements for the same data, the number of systems, multiform data, numerous applications' integration, and various sources in Application Layer of the IoT. The application layer also has another security issue like shielding users' privacy from unsolicited access to personal information, while those users have right of entry to the application service platform for carrying out identity authentication.

The existing security ways and measures for each distinct layer in the IoT are independent of each other, so it is well understood that it is not adequate to offer security assurance for the whole IoT application. One example can be that certification is the identification among different levels in the traditional authentication technique, for that reason, the authentication positioned at the network layer is independent of that to be done at the application layer. So, it is vivid that there is no relationship among the two types of authentications. But, in the IoT environs, business applications and network communications are connected very closely as they work altogether. So, well-understood matter is that it is hard to make them

work independently. So, an example for better understanding is that the security prerequisite of privacy protection is not only dependent on a certain level of the IoT, but in practice it also includes each one of it. Moreover, from a design aspect, the IoT network security architecture does not imply to communicate between devices or any articles. So, in a nutshell, we should replan the security architecture of the IoT. This replanned security architecture should improve the security shielding procedures in the application process and in the development of the IoT.

Designing the security architecture of the Trusting IoT is to facilitate information security protection for data transmission, sensor data security, and tag privacy. Also, another goal is carrying out an intensive systematic research on the transmission and information security of the core network founded on the IoT or networking business security of the IoT. The security architecture of creditable IoT will bring together trusted computing into the architecture to build a chain of trust from the perceived source, and associate the network and service platform, sensor node, with the trust relationship. So, it is well understood that it adds safety techniques in each layer, which is different from the existing network security system. As an outcome, the new architecture could offer the solid theory basis and application guide for the application of the IoT. Also, this architecture is credible and controllable material network architecture, which promotes the networking applications and development.

The essential tactic to make a real-time trusted IoT is to consider three layers as shown in Fig. 7.2. Also, after taking perception/sensor, network, and application layers, for making the Trusting IoT, the resulting most important outcome is the much more improvement in the cyber security. Some examples of this architecture can be as follows.

At first, one of the well-known ciphers, the ECC algorithm can be embedded into tags. The reason is that it will execute the privacy protection to shield the data from modification, usage, duplication, or illegal access. Also, we can use the CPK, which is a known identity-based authentication. It will help us to resolve the mass and fast authentication at the sensor/perception layer.

Second, for the network layer, for providing identification authentication, we can embed the CPK-specific communication chip into the wired or wireless-oriented communication equipment. So, this eliminates the needs of a trusted third party certification. Also along with this, transport code authentication can be used to launch data integrity and confidentiality for communicating data encrypted. Here, the cryptographic power is not less as the key size is not less than 256 bits in the process of data transmission encryption. So, as a nut shell, the above methods can be anticipated to implement an identity-based data transmission encryption. Also, it does not need a third party among entities labeled on identifier.

At last, the trusted access control can be used for the application layer. This trusted access control is used to avoid the illegitimate incursion and safeguard users' unique legitimacy when they log into and necessitate services. Also, this trusted access control can be used also to track main performance, for instance, the operation of business, conforming events for guaranteeing the act of operating non-repudiation and identification of actual operator. And then, to accomplish a

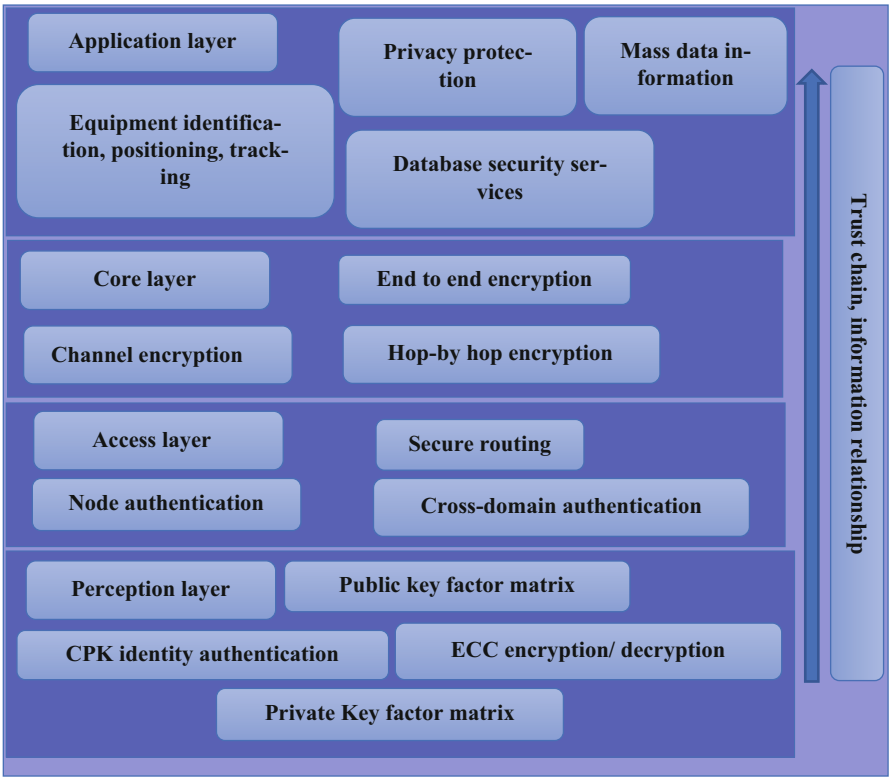



**Fig. 7.4** Creditable security architecture of IoT

trusted and safe runtime in open and unsafe network environs, the authentication on code, trusted thread, and process can be accomplished. Furthermore, trusted database can be used to execute data access mutual authentication, for more defense for the network layer. As a result, the creation of whole defense is built to make sure space safety and manageability of the IoT's field. Figure 7.4 has shown the Trusting IoT's security architecture.

### *7.2.2 Main Security Requirements and Their Sub-Components*

If we try to review security requirements from the domain of the IoT, then we have to consider also the correlated areas of IT and their necessities in the context of the properties of the IoT. For that, we can classify the requirements into five groups: *Network Security, Identity Management, Privacy, Trust, and Resilience*. These five key security necessities together with their sub-components are depicted in Fig. 7.5.

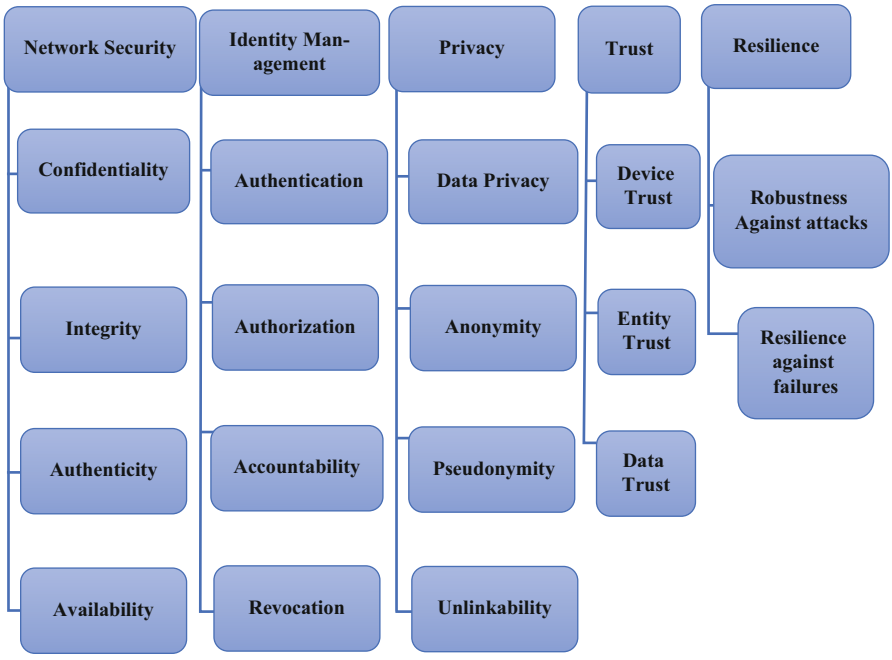



**Fig. 7.5** Main security requirements and their sub-components

**Table 7.1** IoT properties and security requirements: the "*" symbols represent the level of influence in a scale from one (low) to three (high)

| | Network security | Identity management | Privacy | Trust | Resilience |
|---|---|---|---|---|---|
| Uncontrolled environment | * | * | * | ** | * |
| Heterogeneity | * | ** | * | ** | * |
| Scalability | * | * | ** | * | *** |
| Constrained resources | ** | * | ** | * | * |

In addition, Table 7.1 illustrates the association among the numerous IoT properties and the security necessities. It is well understood, for network security, that the constrained resources have the strongest connection. It is for the reason that, mainly due to the constrained resources, there are some restrictions to implement traditional security mechanisms, e.g., cryptography in IoT. Heterogeneity of the IoT mostly has influence on the identity management. Privacy is commonly linked with scalability and the constrained resources, as limitations are posed to the technology candidates that can be utilized. Additionally, the uncontrolled environs and the heterogeneity of the IoT have a big effect on trust. Also, resilience is straightly connected to the need of the IoT for scalability.

Let us now discuss the five requirements in detail as shown in Fig. 7.5.

1. *Network Security*: We can split this requirement into confidentiality, authenticity, integrity, and availability. When we are considering the IoT security architecture, we need the architecture that necessitates the architectures, which deal with the heterogeneity of things. So, it is well understood that interconnecting things may necessitate confidentiality. For example, it should be able to stop eavesdropping the sensitive information via Internet transmission. We already have this for our Internet transmission to fulfill this requirement, for instance, IPSec and Transport Layer Security (TLS). Nevertheless, overhead may exceed the resource constraints of things and therefore dedicated secure network stacks for the IoT exist in this era. We have taken care about authenticity, as it offers evidence that a connection is established with a legitimate entity. Integrity makes sure to detect if any data is lost or modified during transmission. The integrity can be obligatory in the absence of authenticity to detect and recover failures also. But, IoT scenarios need some different, like it may necessitate transactional integrity, like, critical infrastructures, so we can take the architectures as well. Availability makes sure that the connectivity of a thing or service continues, in the scenario of link failures. For that reason, IoT architectures should guarantee that link handover is possible.
2. *Identity Management*: Identity management is really a big challenge in the IoT, as we can have 50 billion devices by 2020 and then another challenge is the complex relationship between users, devices, services, and owners. Henceforth, we have to pay more and more attention to accountability or non-repudiation, authentication, and authorization including revocation. Also, if the abilities of direct authentication to the devices exceed, then user provisioning option should be there, meaning that a user with her/his service credentials can be able to provisioning many devices. Henceforth, ways and means to claim ownership and have control over devices are obligatory. Inside the IoT scenarios, interactions may stretch through numerous domains but our existing authorization solutions, e.g., Kerberos, presume a single domain that enfolds services, owners, users, and devices. Consequently, resolutions for federated authorization that works for non-trusted devices permit the delegation of access through many domains and offer swift revocation. An example can be for broken or rogue devices, it is obligatory. One of the big challenges in IoT is Accountability, for the reason of the magnitude of reuse of data, services, and devices also for many purposes. It makes sure that every action is obviously bound to an authentic entity. Therefore, accountability's obligation is to pact with massive amounts of actions, delegation of access to entities that span continuous derivation of data along with organizational domains.
3. *Privacy*: As the involvement of citizens is increasing day by day in IoT and ubiquitous data collection, e.g., in smart home scenarios, is also increasing day by day, so as an outcome, Privacy is now one of the most dominant challenges in the IoT. Data privacy actually ensures the confidential data transmission. Like a stored data record, it requisites not to uncover undesired properties, for instance, the individual's identity. So, it is very well understood that this requisite is a big challenge in the IoT, due to the reason that many sensing devices have to bring

together personal information. Actually, huge amount of such data turn out to be Personally Identifiable Information (PII), when combined together and this data recognize a person. There are some models which can "anonymize" these kind of data records. But, we have seen that they are insufficient. Addition to that, models to defend this data privacy under data exchange among domains are rather uncharted and complex for implementing it. The property of a single person not being recognizable as the source of data or an action is called as Anonymity. Anonymity is anticipated in the IoT on every occasion, when a persons' identity is not obligatory to fulfill the data minimization laws (Directive 95/46/EG). Along with that, it is anticipated to dismiss preconceptions that arise with data collection in the IoT. It is very hard to attaining anonymity, due to the reason that wearable and mobile devices may disclose PII, for example, IP addresses and location unwittingly. In the present time, we have technologies like anonymous credentials and onion routing, but it may not balance appropriately with the IoT. To trade-off anonymity with accountability, the best tactics should be Pseudonymity. In pseudonymity, actions of a person are allied with a pseudonym, which is nothing but a random identifier, instead of an identity. Pseudonyms can be used in multi-purpose. An example can be connecting several activities of the same person or offering elegant degradation of anonymity for abuse cases. Also, pseudonyms may give resolution for the issues like privacy and accountability concerns in the IoT. Only, standardized resolutions that accompany several domains are obligatory. As definite actions of the same person must not be connected together, so we can say that unlinkability qualifies pseudonymity. Unlinkability defends the profiling in the IoT. Although pseudonyms may resolve unlinkability. One of the examples can be a dissimilar pseudonym being used for each action, cross-implications with anonymity, in specific unidentified meta-data, remain a challenge. In addition, some entity can every time link every pseudonym to a person. So, it is well understood that it can thus also link all activities of that person.

4. *Trust*: One of the crucial prerequisites in the IoT is Trust. The reason is that in reality it is dependable on qualitative data, along with that it is highly distributed also. We can classify the Trust into data trust, entity trust, and device trust. Data trust takes place in the IoT in a dual manner. At first as we know, data come out from several and potentially illegitimate devices. Henceforth, trusted data need to come out from illegitimate sources. It can be done like by applying data aggregation and machine learning techniques. It is well understood that due to the reason that a priori trust in devices cannot at all times be established, so device trust is really a big challenge. It cannot be established due to many reasons, like for high dynamics and cross-domain relations. Henceforth, methods, for instance, trusted computing (for standardized devices) along with computational trust, are obligatory to constitute device trust. Furthermore, every entity may consider trust in a device in a different way. So, as an outcome, IoT architectures have to work with non-singular views of trust. Anticipated behavior of participants, for example, persons or services, is referred as Entity trust in IoT. As we know, device trust can be constituted via trusted computing.

But, planning such methods to introduce into device trust, e.g., via behavioral attestation, is much more challenging and experimental. Another issue is that new data is always obtained from IoT services. One example can be by integrating diverse types of data, we can get new derived data. So, as an outcome, a new trust assessment is obligatory for these newly generated data. Solutions can be in many ways like via computational trust.

5. *Resilience*: One of the most important necessities of IoT is resilience and robustness against attacks and failures. The reason of these attacks and failures are uniting of scale of the IoT in terms of devices. Architectures have to be built up in such a way that it should be able to offer way to adeptly select services according to their robustness (failure/attack avoidance), transmission paths, and things. Moreover, to safeguard fail-over, recovery mechanisms and resilience, the architecture must be able to uphold operations in case of failure or attacks. Also, the architecture should be designed to return to normal operations (failure/attack mitigation).

For end-to-end communication, we have some security solutions at corresponding layers of stack used in that end-to-end communication. So, let us have some highlights on those security solutions.

#### 7.2.2.1 Link Layer: IEEE 802.15.4 Security

The IEEE 802.15.4 protocol is used as link layer in the 6LoWPAN networks. 802.15.4 Link layer security is the current security resolution for the IoT. The node which is being used for communication process needs to be trusted in the link layer. As we know in the link layer, several numbers of nodes along with multiple numbers of hops can be used for communication. A key has to be well defined before the communication starts. This key has a very big role, it is actually always been used for defending all the particular communication going on, in the communication cycle. So, it is well understood that if this key is compromised, then the security of the whole layer is totally gone. Another highlight is that unwanted alteration at individual hop can be discovered by the per-hop security procedure. Data integrity has to be offered for all the hops security measures, with the 6LoWPAN networks. As we know that link layer security has a big disadvantage, it can only provide security in the communication among two adjacent nodes. Still, it is one of the flexible preferences as it can be used with several protocols at any layer, which is above the link layers.

#### 7.2.2.2 IP Security: Network Layer

The IPSec protocol is able to offer security for the network layer. Most relevant thing is that this offers end-to-end security with replay protection, integrity, confidentiality, and authentication. Another advantage is that the IPSec protocol can

be used with several transport layer protocols, for instance, HTTP, User Datagram Protocol (UDP), CoAP, and TCP. IPSec, being a network layer security solution, its security is shared by all the applications, in a running state on a particular device.

#### 7.2.2.3 Security for Transport Layer

IPSec has robustness problem in case of web protocols, and it really lacks robustness. In Transport Layer, TLS or its predecessor Secure Sockets Layer (SSL) is used generally. TLS protocol has solo use over stream-oriented TCP. So, it is well understood that it is not a great technique for wireless communication. The connection-oriented TLS protocol has solo use; it is used in over stream-oriented TCP. But, the problem is that it is not the favored technique of communication for embedded smart objects. Datagram TLS is actually a special protocol, and it is an adaptation of TLS for UDP. DTLS actually can guarantee the end-to-end security of dissimilar applications. It can protect DoS attacks, as it can use the cookies in the web protocol domain. But, again DTLS can be used with the UDP protocols. Thus, it is imperious to make use of the DTLS for offering end-to-end security with IoT.

#### 7.2.2.4 Network Security

As we know that the network is vulnerable to the network attacks, so it is well understood that these attacks can compromise the security. There are many Intrusion Detection Systems (IDS), which are able to detect impostors and malicious happenings in the network. Also, Firewalls are obligatory to block unauthorized access to networks. 6LoWPAN networks of the IoT are susceptible to several attacks from the Internet and from inside the network. So, as a whole it is well understood that it is easier to compromise the wireless domain resource-constrained IoT world than our present regular Internet. So, it is most urgent to develop unique IDS for developing a more complete security for IoT-enabled devices.

#### 7.2.2.5 Data Security in the Internet of Things World

It is well understood that various network security mechanisms make the network communication secure. Our next big issue is how to safeguard the data that IoT devices have stored. We know that the stored data in the IoT devices can be private and sensitive and prerequisites to be secured. IoT world will encompass many tiny nodes which will be resource constrained. So, it is very well understood that the biggest issue is the difficulty to safeguard each of these billions of devices physically or by the use of Trusted Platform Modules (TPMs). Generally, in IoT security, we first take care about setting up of security services at basic level, including authorization, availability authentication, integrity, non-repudiation, and

confidentiality. The present multitude of control protocols for the IoT systems is the Zigbee standard though the security architecture of IoT is in a high progression.

We know that IoT embeds dissimilar kind of sensors into a diversity of goods in reality, so it is very easily perceived that the application of IoT encompasses a lot of private information about users, for example, location, personal information, etc. Now, the reality is that on one side, we presume that the service suppliers to provide the most correct outcomes with our own provided information. On another side, we anticipate that our highly solicited personal privacy can be secured from illegal access. The present-day, privacy shielding procedures consist of space encryption, anonymous space and time, location camouflage, and so on. The role-based access control (RBAC) method in the architecture of IoT defends the security of information to level. But, it is not a complete solution, as it has some insufficiency on identity falsification, information revelation, and other attacks. Another disadvantage of RBAC is that it prerequisites to accumulate huge amounts of information in the database. One good option is Privacy protection based on cryptography. It encompasses homomorphism encryption technique and secure multi-calculation technique and so on. In addition, we prerequisite additional computing resources to add these techniques. Another good result we get after using the K-anonymity technology is that here attackers cannot detect the definite target. But, the disadvantage of this technique is that it does not have any mechanism to protect individual information, as a result, we have to wait for the number of objects attaining K in the group.

Privacy homomorphism was first projected by Rivest in 1978. Many scholars, after that, projected several encryption schemes. But, we have seen that these schemes either only have homomorphism on multiplication (RSA algorithm) or only on addition (IHC algorithm), so these are with limited options of security. But, a very few have homomorphism both on multiplication and addition. But, the biggest problem with these very few algorithms is that we cannot use it in real-time scenario, due to their security flaws. In the past, we have seen that the deterministic privacy homomorphism can be broken in polynomial time. In 2009, a mathematical object based on ideal lattice to understand fully homomorphism algorithm by working together with the encrypted data in this particular way was proposed by Graig Gentry, a researcher from IBM. But, due to the synchronization efficiency improvement, it was not put in real-time use, but it was really a big innovation in fully homomorphism area. Cryptographic community's one of the most enlightened research areas is now homomorphism technology. It permits direct working out on encrypted ciphertext, devoid of decryption and likewise, the result is alike to the ciphertext of the plaintext computation. When we use the privacy homomorphism technology in IoT, a diversity of services are offered to the users, devoid of decrypting users' secret data. So, in a nutshell, a good resolution of personal data security in IoT will be a *personal secrecy protection policy model* that depends on homomorphism encryption. This type of model can advance the efficiency of encryption algorithm. We can enjoy the suitability of the services, although the service providers cannot decrypt the ciphertexts of private information.

Nowadays, IoT extends itself from “anywhere, anyhow, anytime” computing to new extent, we can say the new one as “anything, anyone, any service.” More and more use of IPv6 protocols are there now, for interconnecting present series of computers, along with the smart objects those are in development in the area of Wireless Sensor Networks (WSNs). Most excitingly, merging of IoT-based systems into the kingdom of Internet is going to make a huge change in the direction of future. We can now think that the world with full of IoT-based objects and unified communication is going from end to end through these objects on the IPv6 platforms. So, we know that to make the IoT infrastructure trustable and reliable, the infrastructure needs to be able to offer confidentiality, device and data integrity protection, authentication, privacy protection, transaction auditing, access control, authorization, etc. NFV is a beneficial tool for permitting us to enforce dissimilar levels of security necessities for having a perfect match with the criticality of the services offered in each logically isolated network partition. In the same way, we can use the gateways to impose strict security actions to separate a user-premise network (e.g., a human-body area network (biological sensor networks) used for healthcare) from illegitimate outside domains. Here, it is well understood that resource-constrained user devices are defended by the gateways from illegitimate access. Also, the gateway defends resource-constrained user devices, from being compromised by a mischievous outer entity.

So, in a nutshell, to have a proper secure and privacy protected proliferation of IoT services, we need architectures which are entailed with customized security and privacy levels. These all above literatures give us a wide-ranging overview of many open issues with future directions in the IoT security field. In precise, the secured IoT necessitates compliance with well-defined security and privacy strategies, privacy for users and things, confidentiality, access control, and trustworthiness among devices and users.

## 7.3 Constrained Application Protocol: Application Layer Connection-Less Lightweight Protocol for the Internet of Things

### *7.3.1 Constrained Application Protocol*

The CoAP is a standard web transfer protocol. This CoAP is an ideal protocol, for being used with constrained devices and low-power networking. For M2M applications, it is an ideal choice. Some of the examples can be smart energy and building automation. The CoAP runs over UDP, resulting in non-reliable message transport. Another highlighting point is that it is not session based, along with that the CoAP can tackle loss or delayed delivery of messages. CoAP offers a request/response communication model among application end points. It also has built-in discovery of services and resources support. The CoAP comprises


**Fig. 7.6** Constrained Application Protocol (CoAP) message format

significant conceptions of the Web, such as extensible header options, URIs, and RESTful interaction, etc. CoAP's special ability is that it can effortlessly interface with HTTP for incorporation with the Web, at the same time, meeting specialized necessities, for instance, and simplicity for constrained environments, very low overhead and multicast support. CoAP message structure is shown in Fig. 7.6.

The first byte encompasses the protocol version Ver, a type field T, and TKL. The T is a type field consisting of basic message type information. TKL represents the size in bytes of the Token field. Then, we have the Code field. The Code field encompasses more specific message type information. Then, we have Message ID field. The Message ID field is a unique ID. The work of this unique ID is to track messages and distinguish likely duplications. To match request and response messages, the optional Token field can be used. The value of this Token must be produced at random, and in addition to that, it should be unique for each request. The field varieties are in between 0 and 8 bytes in size. These varieties of field are actually for making CoAP more robust to battle the IP-spoofing attacks. We should use this just in case security is not offered at the transport layer. Moreover, more than a few dissimilar CoAP options have been well defined. Now, it is possible to state a list of them in line with a Type–Length–Content scheme. At last part of the structure of CoAP message, it has the Payload field. As we know, the IETF CoRE working group has projected the CoAP as a new application-level protocol for constrained devices. But, astonishingly, the CoAP has no security measures, but nowadays, research works have projected positioning the DTLS or IPSec protocols to offer a secure CoAP.

### *7.3.2 Constrained Application Protocol–IP Security*

We know that IPSec is a layer three protocol. It is ideal for use with IPv6, but later stage, it is now can be used for IPv4. It can protect application and transport layers' applications but good thing is that it is not an application-dependent protocol.

The reason for this independence is that the IPSec is integrated into the kernel, resulting in transparency to the applications. For the reason of this transparency, TLS and Secure/Multipurpose Internet Mail Extensions (S/MIME) [RFC 3851] can be used by IPSec. The IPSec can offer various security services like: Limited Traffic Flow Confidentiality, Anti-Replay mechanism, Access Control, Confidentiality, Connectionless Integrity, and Data origin Authentication. One way to use IPSec, to secure the CoAP transactions, can be Encapsulating Security Payload Protocol [RFC 2406] (IPSec-ESP). It can be a special case, if the hardware provisions encryption at layer 2 (it is the situation with some IEEE 802.15.4 radio chips). Another way can be the 6LowPAN extension, for using the IPSec with AH [RFC 2402] or ESP.

There are some issues with IPSec. *First* point is that basically the IPSec and DTLS were not considered for the constrained environs. At that time, the constraints were not considered in the IPSec/DTLS designs. *Second* point is that IPSec has been identified with problems for making use of Network Address Translation (NAT) and/or Port Address Translation (PAT). *Third* point is that performance of the network gets worse when communicating small packets, as the encryption procedure of IPSec produces a large overhead. *Fourth* point is that security association (SA) has an issue in IoTs, i.e., the mobility. The Security Parameter Index (SPI), Destination IP Address, and Security Protocol Identifier identify the SA, uniquely. Now, in this case, the issue is that if a node alters its IP address afterward the formation of the SA, then new SA prerequisites to be formed, which will give unnecessary performance degradation. *Fifth* point is that IPSec is inserted in the IP stack, so any alterations will have the need of kernel level. *Sixth* point is that Configuring/Managing/Troubleshooting IPSec and Internet Key Exchange (IKE) are very much composite tasks. It is well understood that an enormous number of constrained devices are taking part in the network. Any wrong configuration of security parameters of IPSec could give security holes or performance problems. *Seventh* point is that every scenarios/nodes cannot be supported by IPSec. Simply to understand, the support of IPSec for multicast communication is problematic. Last but not the least, as per the CoAP's draft, it is promising to use IPSec (ESP) with layer-2 encryption hardware. It provisions the use of AES-CBC (128-bit keys).

A comparison of IPSec and DTLS in various security dimensions is described in Table 7.2.

Also apart from the above issues, the DTLS and IPSec are not the most enhanced resolutions, to offer proper protection to CoAP for many reasons. The reasons are, at *first*, IPSec and DTLS necessitate extra messages, to work for the security parameters and form the security associations (SAs). But, the overhead and drain out of the resources of the constrained devices will be increased much more. This is very problematic for the mobile types of the devices in the IoTs, as new AS prerequisites to form every time the device in mobility. The *Second point* is that if we think about the environs of the communication among two dissimilar networks, the ideal security resolution is dependent on either IPSec or DTLS, which point towards the existence and provision of these protocols, in both the source and destination networks. But, this ideal idea cannot be realistic in many circumstances, particularly

**Table 7.2** A comparison of IPSec and DTLS in various security dimensions

| Security dimension | IPSec | DTLS |
|---|---|---|
| Access control | No | No |
| Authentication | Yes | Partially server only |
| Non-repudiation | Yes/No, as per the authentication method. PKI not supported by constrained devices | Yes/No, as per the authentication method. PKI not supported by constrained devices |
| Confidentiality | Yes | Yes |
| Communication security | Yes | Yes |
| Integrity | Yes | Yes |
| Availability | Mitigation—no full defend | Yes—stateless cookie |
| Privacy | No | No |

when we think about the fact that the IPSec protocol has a compatibility problem with firewalls throughout the networks. *Third* issue is that both IPSec and DTLS count on the IKE and the Extensible Authentication Protocol (EAP), for setting up the secure association and sometimes any other. So, it is well understood that this points towards that all constrained devices' vendors requisite to support these additional protocols (IKE and EAP). *Fourth* point is that the IPSec and DTLS are aimed at securing connections among two static and remote devices. So, the IPSec and DTLS attempt to offer the most possible secure connection among the two ends, devoid of the QoS, the network trustworthiness, or any other restrictions on the end devices' considerations. But, in the environs of the constrained environment, there is a need for more dynamic and sensible actions that think about the constrained type of the end devices at the time of negotiating the security parameters. The *fifth* point is that the IEEE 802.15.4 specification describes that the payload should be 127 bytes as whole. So, if we use the DTLS as security protocol, to defend CoAP exchanges, 13 bytes (out of the 127 bytes of IEEE 802.15.4 frame) has to be assigned for DTLS record. Also, 25 bytes has to be used for link layer addressing information, and 10 bytes for 6LowPAN addressing; along with that 4 bytes for CoAP header. So, as an outcome, only 75 bytes are available, for application layer payload. But, it is not sufficient space for communicating the actual data. Subsequently, one big piece of data (bigger than 75 bytes) will use additional resources from the nodes and the network itself. The reason is that it will be broken into several pieces and will be sent twice. Hence, some header compression mechanisms are good solutions, at the exact cases where needed. The compressing and decompressing necessities are the reason, for more constraints to the nodes and network resources. The *Sixth* point is, in the case of DTLS, that some applications might necessitate security services to be more and more customized in relation to the applications' or scenarios' requirements. Nevertheless, if the security were applied as per the requirements of the application or scenario, it would offer to decrease the usage of resources existing and definitely would increase the network enactment. *Last but not the least*, in the Internet draft of "Datagram *Transport Layer Security in Constrained Environments*," the authors point out seven prospective problems, correlated to

DTLS protocol, if employed in constrained environs. The authors also have pointed out some projected workaround, to resolve these problems. Still, much of work is required, to make the DTLS perfect for making it a good and prospective security resolution for IoTs.

The *Secure CoAP (S-CoAP)* is a secure variant of CoAP. In S-CoAP, the security technique is actually an integrated part of the protocol itself. With S-CoAP, security measures will be integrated into the plain CoAP transactions. So, one of the good features is that it will have its own compromise stage that thinks through the limits of the constrained devices. The S-CoAP prerequisites to offer security for normal connection setup, in addition to that, for the case of mobility also. So, in a nutshell, the advantage is that the security will be an integral part of the CoAP protocol. It is well understood that this security is offered by other standards, so the S-CoAP should be capable to function across numerous sites and networks.

## 7.4 Datagram Transport Layer Security Overview and Supporting Constrained Application Protocol

### *7.4.1 Datagram Transport Layer Security Protocol*

The DTLS protocol is UDP based. The DTLS comprises of four protocols: the Handshake protocol, Alert protocol, the Change Cipher Spec protocol, and the Record protocol. The DTLS protocol offers message fragmentation at the Handshake layer. This enables the DTLS to get rid of message fragmentation in the network layer. These fragmented packets bring many problems, like data loss rate increases and unnecessary delays made by packet retransmission. So, it results in worse LLN conditions. The main burden to a memory-constrained device is to reunite a fragmented message packet, due to the reason that devices have to retain fragmented pieces of the message in the buffer unless until all the pieces reach. To resolve these issues, the DTLS In Constrained Environments (DICE) standard WG was shaped. Nevertheless, definite solutions have not been projected yet. So, it is a well-known thing that to decrease the load on memory of the devices used in making IoT environs, *lightweight DTLS* was projected. *Lightweight DTLS* is able to decrease the DTLS code size, for decreasing the burden on constrained memory of a device. Another way to reduce the load can be by decreasing the transmitted message size by compressing the DTLS header.

The CoRE WG projected TLS_PSK_WITH_AES_128_CCM_8, as a basic cipher suite of DTLS to decrease difficulties like packet fragmentation and loss and delay in an LLN. But, here we have one limitation. Here, the PSK is a necessary thing, due to the reason that if it is not there then the devices cannot make use of this cipher suite. To resolve this issue, Gerdes and Bergmann projected a system, in which a ticket is generated. After executing a DTLS handshake among delegators, each delegator produces a ticket. The CoAP server and the CoAP client execute

the DTLS handshake using the ticket. A PSK is encompassed in the ticket. So this way, key circulation is made likely to form PSK-based DTLS channel among nodes. Here, the security policy has not been determined in advance. To have the network efficiency, we can decrease added header data, due to message fragmentation. So this way, we can decrease the packet loss rate and delay. URI based on a CoAP communication environment having a RESTful structure is a good practical approach. Let us now discuss some of the issues about attacks on the above kind of system.

#### 7.4.1.1 Secure Service Manager Spoofing Attack

If an attacker is the secure service manager (SSM), then the most dangerous thing is that the attacker can acquire all the information about the session, due to the reason of delegating the DTLS handshake. So, there is a chance that the encrypted data among end nodes can be exposed to the attacker. A good solution can be the use of PSK_DN (which is shared among the SSM and a constrained device in the bootstrapping phase). This is a perfect solution for protecting from SSM spoofing attack. The good reason for this protection of the SSM spoofing attack is that data is encrypted by use of PSK_DN and then sent, and the attacker cannot deceive a constrained device and cannot get the right to use the encrypted data.

#### 7.4.1.2 Semi End-to-End Security

We have to ensure end-to-end security. The SSM can acquire all session information, by just delegating the DTLS handshake. As we know, the encrypted session information is sent to a constrained device instantly, but the SSM does not do the accumulation of session information. So, it is well understood that end nodes joining in the DTLS communication will encrypt and decrypt data themselves only. The SSM is only responsible for the data relay after sending the session information to the constrained device. In this kind of system, the executor of the encryption and decryption is the end node, in the DTLS communication. There is one obligatory thing: the SSM must trust the preregistered device, for example, smart phone of user. So, as an outcome, we can get an end-to-end security (semi end-to-end security exactly) definitely.

#### 7.4.1.3 Denial of Service

The devices setting up IoT have low CPU performance and a small amount of memory. So, it is a well understood fact that sending a DTLS handshake request message to these low-memory and low-performance devices can seem to be a DoS attack, even supposing that the request is from a legitimate user. Another case is if an attacker transmits a DTLS handshake message straight to a constrained device with

conditions in the LLN, then as an outcome, the devices become more dangerous. So, we can understand that the SSM benefits to resolve the DoS issue by delegating the DTLS handshake. The SSM stops constrained devices from receiving a lot of messages directly.

#### 7.4.1.4 Single Point of Failure

Numerous methodologies applying delegation can give a single point of failure (SPOF). It is one of the utmost predictable, but serious difficulties in security field. We know that the SSM has a significant role of delegating DTLS handshake in place of numerous CoAP sensors. So, it is well understood that if the SSM is negotiated or fails, then all the sensors under the SSM cannot create a secure session with client or server, which are outer of the LLN.

A well-defined trust manager can somehow protect such an SPOF issue. The trust manager has the option to choose alternative authentic device, as a new SSM. Then, he/she can broadcast associated information to his sensors. Only thing is that the SSM should be a resource rich device in smart home or smart building (e.g., smart healthcare devices, etc.). Another way can be virtually applied SSM in cloud system. It is harder to compromise a virtual SSM in Cloud, as it is operated and supervised by security manager, compared to attack a home device or smart phone, which is operated by its usual user. One highlighting point is that here, a secure registration method between the SSM and IoT devices controlled by the SSM is there. Moreover, another supposition is that the secret key, which is common for both SSM and its devices, cannot be compromised. Future research can be on designing and implementing a concrete secure system, with additional mechanisms including key revocation, secure bootstrapping, trust management, and so on.

#### 7.4.1.5 Fragmentation Attacks

A packet fragmentation mechanism is a good resolution for dissimilar MTU size among Internet and LLN. An IPv6 adaptation layer, 6LowPAN, has a provision with a method to fragment large IPv6 packets into small frame. Normally, sensing data and control data for actuators can be small in size. Though, DTLS handshake message is bigger in size than the maximum frame size of LLN, for instance, IEEE 802.15.4 (i.e., 127 bytes). Particularly, DTLS fragmentation is unavoidable at the fourth flight of DTLS handshake. The reason is that it encompasses comparatively large size of certificate of server and key exchange message. We can send 27 DTLS fragmented datagrams in case of using TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 with Raw Public Key Certificate. Significant transmission overhead is the outcome from these fragmented datagrams for the reason that the header is added to each of the frames. But, some other critical issues are that, due to the deficiency in authentication mechanism at 6LowPAN layer, it gives chance for attackers to try buffer reservation attack,

fragment duplication attack, and fragmentation attacks. An attacker eavesdrops and modifies a fragmented frame in the middle of the wireless multi-hop link, to lunch the fragment duplication attack. At the time of receiving, the Target node cannot identify the altered frame. So, as an outcome, the attacker's just a single forged frame can stop successful reassemble execution of the target node. Additionally, the target node requisites to abandon all frames in the buffer and awaits for retransmission once more, resulting in the DoS attack. We know that the first frame retains a memory space for reassembling the original packet and it is indicated in the header (i.e., datagram size field) at the target node. Also, the buffer reservation attack exploits this fact. The attack can be very simple, like the attack can be done by sending a forged start frame encompassing large number in the datagram size field.

A good option with a good efficiency can be a scheme, which uses the SSM to delegate the DTLS handshake phase. For the constrained network like an LLN, network overhead, and delay and loss problems, due to fragmented handshake message packets, are resolved by delegating the handshake. For the constrained device, the device need not retain the fragmented handshake packets, in the buffer until receiving all of them. In addition, DTLS communication devoid of any source code for a DTLS handshake can be used by a constrained device. Here, the end-to-end security is definite, as data encryption and decryption are done in the end node. Also, its more important feature is that the system can tackle an SSM spoofing attack and DoS attacks on a constrained device. Another highlight is that the SSM and the constrained device are tangibly distinct but can virtually be considered as one system in a trusted relation with a shared key. This shared key is a pre-shared. So, in a nutshell, this kind of scheme can benefit for deploying constrained devices in a secure manner in constrained environments.

### *7.4.2 Supporting Constrained Application Protocol*

The DTLS protocol is nothing but an improved type of the very popular TLS protocol [RFC 5246]. To give more security, to the major UDP well-known applications, for instance, Voice over IP/Session Initiation Protocol (VoIP/SIP), DTLS runs on top of UDP instead of TCP. This is a key difference. The DTLS offers automatic key management, confidentiality, authentication, and data integrity. It also provisions wide range of dissimilar cryptographic algorithms. As per the CoAP's draft, CoAP describes four security modes with the intention of achieving the security services, which is obligatory. They are: NoSec, PreSharedKey, RawPublicKey, and Certificate. In case of NoSec mode, the packets are transferred usually as UDP datagrams over IP. The CoAP scheme indicated this as coap://. In case of all other three security modes, security is attained by DTLS and the scheme is indicated by coaps://.

Now, let us discuss some issues of the *DTLS supporting the CoAP*. At *first*, multicast communication is not offered by DTLS protocol, but it is an essential

part of CoAP protocol and main feature in IoTs. *Second* thing is that the DTLS handshake protocol is not protected at all; anytime it can be attacked by the exhaustion attack of the resources of battery-powered device, may be with the stateless cookie also. So, it is well understood that as an outcome, the nodes could not work properly in the network and make interruption to the whole communication. *Third*, bitmap window can defend the DTLS from replay attack, but still the nodes have to obtain the packets first, then process and occasionally even forward them also. This attack could make the network flooded. So, good resolution can be filtering proxy, for instance, 6LoWPAN Border Router (6LBR). Also, one point in this resolution is that the possibility of running this kind of filtering on a 6LBR cannot protect all situations. Furthermore, handling the replied packets is energy consuming. *Forth* issue is that Handshake phase is strongly defenseless, ever since no end-host has been authentic to the other end-host. *Fifth* issue is that DTLS's security advantages do not match with the CoAP. For example, the loss of a message in-flight necessitates the re-communication of all messages in-flight. But, if all messages in-flight are communicated together in a single UDP packet, its good, but more, resources are obligatory for dealing with large buffers. Additionally, if CoAP client prerequisites Internet access, which essentials the CoAP/HTTP mapping process, then it is well understood that the DTLS handshake process will be a big issue. Mainly, it is not clear if a partial mapping among TLS and DTLS can be accomplished. This topic could also be more complex, since a CoAP client would not be capable to distinguish which device has started the request. Last but not the least, CoAP messages have two transactions (one round-trip); one message starting at the client (request) and the other starting from the server (response). If DTLS is used in these two transaction processes, then we need four round trips, three round trips for DTLS ( 40–50 Bytes) and additional one round trip for CoAP. It should be before CoAP's actual contents are exchanged.

Distributed IoT applications can use the CoAP at the application layer, with the intention of regaining the resources from sensing devices and in case of the autonomous communications, among WSN and Internet devices. CoAP can be used to empower the application layer RESTful communications with these sensing platforms. So, this can be one of the foundations for the forthcoming great future of future IoT applications. So, it is well understood that the security in case of the CoAP has a major importance. The existing CoAP specification accepts DTLS (Datagram Transport Layer Security) at the transport layer security, for the purpose of transparent secure CoAP communications at the application layer. DTLS offers end-to-end security. But, in actuality, DTLS has a conflict with one functionality designed in CoAP which is the usage of proxies, to help communications among the Internet and WSN communication domains. Another prospect for DTLS for CoAP necessitates the use of public key authentication by use of ECC (Elliptic Curve Cryptography), for the purpose of the authentication and key agreement.

The *handshake* is a big issue for the end-to-end security. The reason is that, after completion of the authentication and key negotiation, the end-to-end security implementation issue can be resolved in the sensing device very efficiently with AES/CCM encryption. We know that the transparent interception and mediation of

DTLS also give us advantages, other than permitting the ECC encryption to make provision for high security with CoAP. The end-to-end security's one of the key components can be the DTLS handshake. It permits for mutual authentication and key agreement, within communicating both the parties. But, it takes some more load, due to its high computational costs, so we should try to offload such costly computations. But, when we are thinking this, we prerequisite to support sensing devices for moving freely in between several WSN domains. We have to take care about the matter that, in the environs of a given IoT application, CoAP resources that exist on sensing devices are securely reachable. The reachability with security should be regardless of the present location of the device. In parallel, there should not be any changes for CoAP and DTLS as maintained on such devices.

## 7.5 Case Studies and Open Research Issues

At first, let us highlight the ongoing projects and consider them as our case studies. The European Union is working on *Butler* (European Union FP7 project) [46–48]. This project facilitates the expansion of secure and smart life assistant applications, along with the security and privacy necessities. Also, this work has developed a mobile framework. The smart applications which are targeted are like smart home/smart office, smart mobility/smart transport, smart health, smart shopping, and smart cities. Another European Union project is *EBBITS* (EU FP7 project) [47]. This project works for an IDS, by use of latest IPv6 over 6LoWPAN devices. Ever since, 6LoWPAN protocol is defenseless to wireless and Internet protocol attacks. This projected IDS framework comprises a monitoring system and a detection engine. The *Hydra project* [49] has projected a middleware for Network Embedded Systems. This middleware is founded on a Service-Oriented Architecture (SOA). Hydra considers the distributed security concerns and social trust within the middleware constituent. Hydra is designed for P2P communication and diagnostics, architecture formed on Semantic Model and the Device and Service Discovery. Another project which is to increase the user trust is *uTRUSTit* [50]. *uTRUSTit* stands for Usable Trust in the IoT (EU FP7 project). It is actually a trust feedback toolkit to potentially increase the user trust. It empowers the system manufacturers and system integrators to express the security ideas. It agrees to create effective decisions on the trustworthiness. *iCore* is another EU project. *iCore* [51] has a management framework with very significant security protocols/functionalities. These protocols/functionalities are having relation with the ownership and privacy of data and the access to objects. This management framework has three levels of functionality: virtual objects (VOs), composite virtual objects (CVOs), and functional blocks. The *iCore* solution can be part of various smart environs, like supply chain management, smart office, smart transportation, and ambient-assisted living.

Now, another very well-known DARPA project is *HACMS* [52]. It stands for High Assurance Cyber Military Systems. This project actually has tried to have

patch of the security vulnerabilities of IoT. This project has taken account of drones, medical equipment, and military vehicles. HACMS provides the seeds for future security protocols and achieves sufficient standardization and security.

NSF, *National Science Foundation*, has a *multi-institutional project* [53]. This project is actually working for the security in the cyber-physical systems. This multi-institutional project is working on several solutions, like trying to discover the efficient resolutions, finding novel network architectures and networking conceptions, trying to invent new communication protocols. Also, they are bearing in mind about the trade-offs between mobility and scalability, technical challenges, trusted data, the integrity along with authentication, trust models, and use of network resources on mobile environments. The EU, China, and Korea are working together in a project called *FIRE* [54, 55]. It stands for *Future Internet Research and Experimentation*. The *FIRE* works for discovering resolutions, for the setting out of IoT technologies in numerous application areas, like medical and health service, urban management, social security, people livelihood, and public safety. They are also trying to give proper focus on intellectual property right, privacy, and information security. Another EU and Japanese collaborative project is *EUJapan ICT Cooperation project* [47]. They have already made the common global standards, to make sure about seamless communications and shared ways to accumulate and have right to use the information. They are also trying to confirm the highest security and energy efficiency standards.

In 1999, the Auto-ID Laboratory of Massachusetts Institute of Technology has introduced us *Thought of "the Internet of things"*. Then, in 2005, we had the "*ITU Internet Reports: The Internet of Things*." We need to develop the security structural design of the IoT, for the reason of offering information security defense for tag privacy, sensor data security, and data transmission, etc. We need very deep systematic research on the transmission and information security of the core network, depending on the IoT or networking industry security of the IoT. We have seen that recent works are simply adding safety methods in each layer. But, this is not at all sufficient. We have seen that, depending on the *privacy homomorphism*, the computational insufficiency of traditional algorithms is enhanced to make sure users' personal privacy security. It is one of the milestone ideas. But, the homomorphism technology presently is not matured enough as required. Now, the homomorphism algorithm is capable of offering the complete integer operations. Still nowadays, it is comprehensive to the real region that the security comes out to a big issue. Also, another disadvantage is that very few homomorphism properties are held by the privacy homomorphism. So, we need more developed homomorphism, which can be extensively used in IoT. We have worked on multilayer *Hybrid RSA*-based solution [45, 56–59] for personal messaging for more efficiency and strong security and privacy as shown in Fig. 7.7. Our Hybrid RSA scheme now works for human messaging, and in later stage this Hybrid RSA cipher [45, 56–59] can be used for Internet of Everything (IoE) for end-to-end encryption with high efficiency and high security with authentication and privacy protection.

In generic, the security actions to be taken for IoT denote to the basic facility of security services comprising availability, authentication, authorization, non-

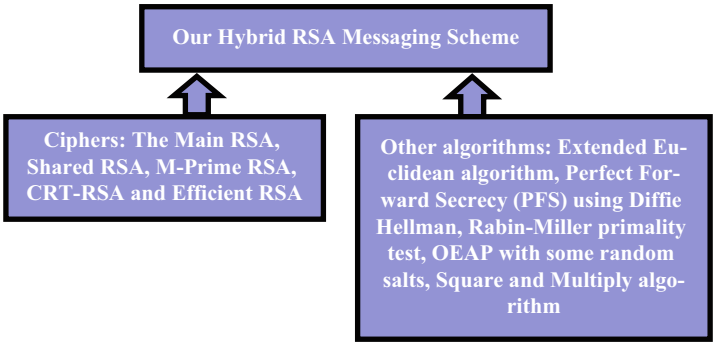



**Fig. 7.7** Our Hybrid Rivest–Shamir–Adleman (RSA) scheme

repudiation, confidentiality, and integrity. The security structural design of IoT is still growing. So, the best way to represent the security need can be by using a reference model as we discussed earlier. So, it is well understood that any single structural design will be problematic for referring to the system. All the researchers, governments, and industries are dedicated for evolving and regulating identity and security mechanisms, for IoT building blocks. We already know that researchers are forming better cryptographic algorithms and modes for IoT devices. The *ISO/IEC 29192 standards* aim for lightweight cryptography for constrained devices. This standard includes block and stream ciphers and asymmetric mechanisms. Sony's *CLEFIA* is an example of block cipher with 128-bit key supports (www.sony.net/Products/cryptography/clefia/about/index.html). The *eSTREAM project* (www.ecrypt.eu.org/stream) has considered the robustness of stream ciphers, for instance, *Salsa20/12* and *Trivium*. These are very much beneficial for embedded systems. Also, we know that some researches on *lightweight dedicated hash functions* are going on. Everybody in this area is trying to make a new *cryptographic hash algorithm* that is able to transform a variable-length message into a short message digest. The digest can be a portion of either generating digital signatures or message authentication codes or can be many other security applications in the information setup (http://csrc.nist.gov/groups/ST/hash/sha-3/index.html). We have already works which are on forming lightweight hash functions, depend on lightweight block ciphers. We know that *AES-CCM* and *AES-GCM* project data integrity and confidentiality. Another way for optimization can be algorithm management in a *cross-layer architecture*. Here, the reason for optimization is numerous security mechanisms that share *one algorithm*. The Internet Engineering Task Force has an intention to execute Internet standards in the IoT. We have seen that many researchers have tweaked the IPSec protocol, for offering the network layer security between Internet hosts and constrained devices. But, still some issues are hard to resolve. We all know that the IPSec prerequisites a shared password, for doing the encryption and decryption for all incoming and outgoing messages. But, big issue is that if these passwords are static,

then it can be compromised after some 1000 messages. For resolving this issue, the IKE (Internet Key Exchange) and IKEv2 protocols were formed. These protocols promise a protected communication among two devices and are capable to generate new shared passwords, by use of circling derivative tactics. We can use DTLS for protecting UDP packets (even over IPSec). By use of an initial handshake, it sets the passwords. Then, the content of the UDP packet is encrypted (usually with TLS PSK over AES) and a header of 13 bytes is added. This process is done together with the initialization Vectors (IV) (over 8 bytes for AES128), integrity values (8 bytes), and the padding prerequisite by the cipher suite. In general, future researches in the security issues of the IoT would mostly quintessence on the following characteristics: the open security system, individual privacy protection mode, terminal security function, related laws for the security of the IoT, etc. It is unquestionable that the security of the IoT is more than a technical problem, which also prerequisites a series of policies, laws, and regulations, perfect security management system for mutual collocation.

**Acknowledgments** This work is supported by National Natural Science Foundation of China (No. 61631013) and Key Laboratory of Universal Wireless Communications (Beijing University of Posts and Telecommunications), Ministry of Education, P.R. China (No. KFKT-2014101).

## References

1. Jara A, Kafle V, Skarmeta A (2013) Secure and scalable mobility management scheme for the internet of things integration in the future internet architecture. Int J Ad Hoc Ubiquitous Comput 13(3-4):228–242
2. Li S, Gong P, Yang Q, Li M, Kong J, Li P (2013) A secure handshake scheme for mobile-hierarchy city intelligent transportation system. In: International conference on ubiquitous and future networks. ICUFN, Da Nang, pp 190–191
3. Kang KC, Pang ZB, Wang CC (2013) Security and privacy mechanism for health internet of things. J China Univ Posts Telecommun 20(Suppl 2):64–68
4. Goncalves F, Macedo J, Nicolau M, Santos A (2013) Security architecture for mobile e-health applications in medication control. In: 2013 21st international conference on software, telecommunications and computer networks. SoftCOM, Primosten, pp 1–8
5. An J, Gui X, Zhang W, Jiang J, Yang J (2013) Research on social relations cognitive model of mobile nodes in internet of things. J. Netw Comput Appl 36(2):799–810
6. Kasinathan P, Costamagna G, Khaleel H, Pastrone C, Spirito M (2013) Demo: an ids framework for internet of things empowered by 6lowpan, Berlin, Germany, pp 1337–1339
7. BETaaS Consortium (2014) BETaaS building the environment for the things as a service D2. 2. 2–Specification of the extended capabilities of the platform, pp 1–61
8. IoT-A Consortium (2014) IoT-A unified requirements. http://www.iot-a.eu/public/requirements/. 31 Jan 2014
9. Gao L, Bai X (2014) A unified perspective on the factors influencing consumer acceptance of internet of things technology. Asia Pac J Mark Logist 26(2):211–231
10. Gazis V (2014) Carlos Garcia Cordero, Emmanouil Vasilomanolakis, Panayotis Kikiras, and Alex Wiesmaier. Security perspectives for collaborative data acquisition in the internet of things. In: International conference on safety and security in internet of things. Springer, New York
11. IoT-A Consortium (2014) IoT-A – Internet of things architecture. http://www.iot-a.eu/. 27 Jan 2014

12. Logvinov O, Kraemer B, Adams C, Heiles J, Stuebing G (2014) Mary Lynne Nielsen, and Brenda Mancuso. Standard for an architectural framework for the internet of things (IoT) IEEE P2413 Webinar Panelists, pp 1–12
13. Zanella A, Bui N, Castellani AP, Vangelista L, Zorzi M (2014) Internet of things for smart cities. IEEE Internet Things J 1:22–32
14. Grieco LA, Alaya MB, Monteil T, Drira KK (2014) Architecting information centric ETSI-M2M systems. In: IEEE PerCom
15. Anderson J, Rainie L (2014) The internet of things will thrive by 2025, Pew research internet project. http://www.pewinternet.org/2014/05/14/internet-of-things/
16. Yan Z, Zhang P, Vasilakos AV (2014) A survey on trust management for internet of things. J Netw Comput Appl 42:120–134
17. Piro G, Boggia G, Grieco LA (2014) A standard compliant security framework for IEEE 802.15.4 networks. In: Proceedings of IEEE world forum on internet of things (WF-IoT), Seoul, South Korea, pp 27–30
18. Lee J-Y, Lin W-C, Huang Y-H (2014) A lightweight authentication protocol for internet of things. In: 2014 international symposium on next-generation electronics, ISNE 2014, Kwei-Shan, pp 1–2
19. Turkanovi M, Brumen B, Hlbl M (2014) A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. Ad Hoc Netw 20:96–112
20. Ye N, Zhu Y, Wang R-CB, Malekian R, Lin Q-M (2014) An efficient authentication and access control scheme for perception layer of internet of things. Appl Math Inf Sci 8(4):1617–1624
21. Cherkaoui A, Bossuet L, Seitz L, Selander G, Borgaonkar R (2014) New paradigms for access control in constrained environments. In: 2014 9th international symposium on reconfigurable and communication-centric systems-on-chip (ReCoSoC), Montpellier, pp 1–4
22. Sicari S, Rizzardi A, Cappiello C, Coen-Porisini A (2014) A NFP model for internet of things applications. In: Proceedings of IEEE WiMob, Larnaca, Cyprus, pp 164–171
23. Wang X, Zhang J, Schooler E, Ion M (2014) Performance evaluation of attribute-based encryption: Toward data privacy in the IoT. In: 2014 IEEE international conference on communications, ICC 2014, Sydney, NSW, pp 725–730
24. Su J, Cao D, Zhao B, Wang X, You I (2014) ePASS: an expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the internet of things. Futur Gener Comput Syst 33:11–18
25. Peng LB, Ru-chuan WB, Xiao-yu S, Long C (2014) Privacy protection based on key-changed mutual authentication protocol in internet of things. Commun Comput Inf Sci 418:345–355
26. Ukil A, Bandyopadhyay S, Pal A (2014) IoT-privacy: to be private or not to be private. In: Proceedings – IEEE INFOCOM, Toronto, ON, pp 123–124
27. Sicari S, Cappiello C, Pellegrini FD, Miorandi D, Coen-Porisini A (2014) A security-and quality-aware system architecture for internet of things. Inf Syst Front 18:1–13
28. Tormo GD, Marmol FG, Perez GM (2014) Dynamic and flexible selection of a reputation mechanism for heterogeneous environments. Futur Gener Comput Syst 49:113–124
29. Gu L, Wang J, Sun BB (2014) Trust management mechanism for internet of things. China Commun 11(2):148–156
30. Liu Y-B, Gong X-H, Feng Y-F (2014) Trust system based on node behavior detection in internet of things. Tongxin Xuebao/J Commun 35(5):8–15
31. Singh J, Bacon J, Eyers D (2014) Policy enforcement within emerging distributed, event-based systems. In: DEBS 2014 – Proceedings of the 8th ACM international conference on distributed event-based systems, pp 246–255
32. Neisse R, Steri G, Baldini G (2014) Enforcement of security policy rules for the internet of things. In: Proceedings of IEEE WiMob, Larnaca, Cyprus, pp 120–127
33. Gòmez-Goiri A, Orduna P, Diego J, de Ipina DL (2014) Otsopack: lightweight semantic framework for interoperable ambient intelligence applications. Comput Hum Behav 30:460–467
34. Colistra G, Pilloni V, Atzori L (2014) The problem of task allocation in the internet of things and the consensus-based approach. Comput Netw 73:98–111

35. Wang Y, Qiao M, Tang H, Pei H (2014) Middleware development method for internet of things. Liaoning Gongcheng Jishu Daxue Xuebao (Ziran Kexue Ban)/J Liaoning Tech Univ (Nat Sci Ed) 33(5):675–678
36. Ferreira H, De Sousa R Jr, De Deus F, Canedo E (2014) Proposal of a secure, deployable and transparent middleware for internet of things. In: Iberian conference on information systems and technologies. CISTI, Barcelona, pp 1–4
37. Niu B, Zhu X, Chi H, Li H (2014) Privacy and authentication protocol for mobile RFID systems. Wireless Pers Commun 77(3):1713–1731
38. Jeong Y-S, Lee J, Lee J-B, Jung J-J, Park J (2014) An efficient and secure m-IPS scheme of mobile devices for human-centric computing. J Appl Math 2014:1–8
39. Geng J, Xiong X (2014) Research on mobile information access based on internet of things. Appl Mech Mater 539:460–463
40. Kubler S, Frmling K, Buda A (2014) A standardized approach to deal with firewall and mobility policies in the IoT. Pervasive Mobile Comput 20:100–114
41. Daubert J, Wiesmaier A, Kikiras P (2015) A view on privacy & trust in IoT. In: IOT/CPS-Security Workshop, IEEE international conference on communications, ICC 2015, London, GB, June 08–12, 2015, page to appear. IEEE
42. Sadeghi AR, Wachsmann C, Waidner M (2015) Security and privacy challenges in industrial internet of things. In: Annual design automation conference. ACM, New York, p 54
43. Sicari S, Rizzardi A, Grieco LA, Coen-Porisini A (2015) Security, privacy and trust in internet of things: the road ahead. Comput Netw 76:146–164
44. Zhang Z-k, Cheng M, Cho Y, Shieh S (2015) Emerging security threats and countermeasures in IoT. In: ACM symposium on information, computer and communications security. ACM, New York, pp 1–6
45. Bhattacharjya A, Zhong X, Wang J (2016) Strong, efficient and reliable personal messaging peer to peer architecture based on Hybrid RSA. In: Proceedings of the international conference on internet of things and cloud computing (ICC 2016) ISBN 978-1-4503-4063-2/16/03. The Møller Centre-Churchill College, Cambridge. https://doi.org/10.1145/2896387.2896431
46. BUTLER Project. http://www.iot-butler.eu
47. EU-Japan Project. http://www.eurojapan-ict.org/
48. European FP7 IoT@Work project. http://iot-at-work.eu
49. HYDRA Project. http://www.hydramiddleware.eu/
50. Usable Trust in the Internet of Things. http://www.utrustit.eu/
51. iCORE Project. http://www.iot-icore.eu
52. HACMS Project. http://www.defenseone.com/technology
53. National Science Foundation Project. http://www.nsf.gov
54. FIRE EU-China Project. http://www.euchina-fire.eu/
55. FIRE EU-Korea Project. http://eukorea-fire.eu/
56. Bhattacharjya A, Zhong X, Wang J (2018) An end to end users two way authenticated double encrypted messaging scheme based on hybrid RSA for the future internet architectures. Int J Inf Comput Secur 10(1):63–79
57. Bhattacharjya A, Zhong X, Wang J, Xing L (2018) On mapping of address and port using translation (MAP-T). Int J Inf Comput Secur. http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijics. https://doi.org/10.1504/IJICS.2018.10008372
58. Bhattacharjya A, Zhong X, Wang J (2018) HYBRID RSA based highly efficient, reliable and strong personal full mesh networked messaging scheme. Int J Inf Comput Secur. http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijics. https://doi.org/10.1504/IJICS.2018.10010256
59. Bhattacharjya A, Zhong X, Wang J, Xing L (2018) Secure IoT structural design for smart cites. In: Smart cities cybersecurity and privacy. Elsevier, New York. ISBN: 9780128150320. https://www.elsevier.com/books/smart-cities-cybersecurity-and-privacy/rawat/978-0-12-815032-0#

**Aniruddha Bhattacharjya** is with Department of Electronic Engineering, Tsinghua University, Beijing, China, as a PhD scholar (under Chinese Government Scholarship (CGS)). His research interests are cryptography, Network security, RFID-based architectures and middleware, security in fixed and wireless Networks, applications of cryptography, and IoT security. He has received the ICDCN 2010, PhD Forum Fellowship. He achieved the best paper award in ACM ICC 2016, in Cambridge University, UK. Since 2012, he has been working as an IEEE mentor and ACM faculty sponsor. He is a member of 34 IEEE societies and various IEEE technical committees. He has published 33 International Journal papers, International Conference papers and International Book chapters as well as one Chinese innovation patent is filed presently.

**Xiaofeng Zhong** received his PhD in Information and Communication Systems from Tsinghua University in 2005. He is an Associate Professor in the Department of Electronic Engineering at Tsinghua University. He performs research in the field of mobile networks, including users' behaviors and traffic model analyses, MAC and network protocol design, and resource management optimization. He has published more than 30 papers and holds seven patents.

**Jing Wang** received his BS and MS degree in Electronic Engineering from Tsinghua University, Beijing, China in 1983 and 1986, respectively. He has worked as a Faculty member in Tsinghua University since 1986. He is currently a Professor at the School of Information Science and Technology. He also serves as the Vice Director of the Tsinghua National Laboratory for Information Science and Technology. His research interests are in the area of wireless communications, including transmission and networking technologies of 4G/5G. He has published more than 150 conference and journal papers.

**Xing Li** is a Professor in the Department of Electronic Engineering, Tsinghua University, Beijing, China and deputy Director of CERNET Center. He obtained his PhD degree from the Department of Electrical and Computer Engineering at the Drexel University, Philadelphia, USA in 1989. He has published numerous papers and is the author of several RFCs. He is also WWW10, PC Chair in Searching Area and Co-Chair of the Coordination Committee of the Intercontinental Research Network (CCIRN).

# Chapter 8
# Cyber-Physical System Security Controls: A Review


**Subhrajit Majumder, Akshay Mathur, and Ahmad Y. Javaid**

**Abstract** The term cyber-physical system (CPS) could be described as a system that integrates the computational and physical capabilities and can work as the connection between the cyber and physical worlds. The capabilities of the system to interact with the physical world through more efficient implementations of these "connections" is a crucial enabler for developing the future technologies. For these, there should be a targeted approach to ensure successful implementation of security measures to address the issues of security such as curtail any illegitimate activity or the breach of data that could lead to damage of a large group of the population, influential business entity or even a government agency. Over the years, researchers have proposed novel techniques and security measures to ensure the security and proper functioning of CPSs. Although with time, many such measures have become obsolete and pose a completely new series of security challenges as the recent attacks have become more deleterious and harder to detect. In this chapter, we identify the threats on CPSs due to the systems' vulnerabilities, discuss recent successful attacks on the systems, problems in security control of the system, investigate the defenses that it can provide and propose a set of challenges that need to be addressed for the improvement of cyber-physical systems security.

## 8.1 Introduction

A CPS is a system made with diverse types of components where the high-end components can monitor and control the other low-end components present in our physical world through cyber (Internet) and physical (wired) connections with the help of computer-based algorithms. In a CPS, the software and physical components work together on their respective spatial scales, displaying numerous

S. Majumder · A. Mathur · A. Y. Javaid (✉)
University of Toledo, Toledo, OH, USA
e-mail: subhrajit.majumder@rockets.utoledo.edu; akshay.mathur@rockets.utoledo.edu; ahmad.javaid@utoledo.edu



S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_8

distinct behavioral modalities while interacting with each other. In recent times, we have seen an exponential development in various CPSs. In our life, the applications of CPSs are constantly enhancing such as industrial control systems, smart grid, medical devices, autonomous automobile system (smart cars), process control systems, automatic pilot avionics, and robotics. However, with more technologies come more vulnerabilities which need to be managed to keep the system secure. Mainly four contents are discussed in this chapter which are (1) the cyber, physical, and cyber-physical components, (2) the possible threats and vulnerabilities of the CPS, (3) the real-life attacks on the CPS, and (4) the existing research on security controls that are required as solutions of these attacks, and what security measures are required to make these solutions even better. We discuss these in detail about four most popular CPS which are

- Industrial Control System
- Smart Grids
- Medical Devices
- Smart Cars

These CPSs are selected primarily because, in our world, the above-mentioned applications of CPS are the most conventional ones and environments around these are very critical. Thus, attacks on these CPSs can bring severe consequences in our daily life. We have briefed about each of the applications which will give an overview of these separately and the synopsis of the communication structure between different components. The components of a CPS are categorized into three types, i.e., cyber, cyber-physical, and physical. Due to heterogeneous properties, we have focused on these distinct categories of aspect individually according to their applications. We have discussed the possible threats due to the vulnerabilities present in the CPSs and the real-life attacks on the systems which have taken place till now. This helps to decide which technologies are required to secure the systems properly as we have shown that different components of the systems were exploited in those real-life attacks, even though there were some security measures present. For references, research which has been made regarding this is demonstrated along with the challenges which will give the idea of which area needs more research.

## 8.2 Background

### *8.2.1 Cyber-Physical Systems*

In simple words, CPSs are the systems that are used to monitor and control our physical world [128]. In other words, CPS is an integration of computation process with the physical world's components [80]. The developments in the information and communication technologies (ICT) have made this integration operate properly. These explanations describe the heavyweight of the interactions between the cyber and physical worlds.

#### 8.2.1.1 Industrial Control Systems (ICS)

ICS is used to better control, monitor, and produce in various industries such as nuclear plants, irrigation systems, and hydro plants. ICS is an integration of Distributed Control Systems (DCS), Supervisory Control and Data Acquisition (SCADA), and many smaller control systems such as Programmable Logic Controller (PLC) or Remote Terminal Units (RTU). These systems contain many controllers with different capabilities that accomplish numerous tasks with collaboration. Among many ICS components, one of the most popular is the Programmable Logic Controller (PLC) which is an industrial digital computer designed for controlling manufacturing processes like assembly lines, robotic activity, or any other activity which requires control with high reliability. It communicates through wired or wireless or both connections which are configured with the surrounding environment. PLC can operate continuously in a hostile environment with the help of its sensors and actuators that are connected to the physical world [81].

#### 8.2.1.2 Smart Grid Systems

Power grid is an electric grid that is built up as a network of transformers, transmission lines, and more that has been used for electricity generation, transmission, and distribution. The primary function of a smart grid is to deliver electricity from the power plant to the electric home appliances of customers as efficiently as possible so that it reduces the management cost for utility and power cost for the customer. The two major components of the smart grid are supporting infrastructure and power application [143]. The supporting infrastructure is the intelligent one that concerns mainly with monitoring and controlling the core operations of the smart grid. However, the core functions of the smart grid are done by the power application. The current smart grid of ours was built in the 1890s and has been improving with the advancements in technology through the years. Today, more than 9200 electric generating units are there with the generating capacity of more than 1 million megawatts that are connected to more than 300,000 miles of transmission lines [74].

#### 8.2.1.3 Medical Devices

For the betterment of health care services, medical devices have integrated with information technologies that are cyber-based. This allows the physician along with the patient to control the devices more conveniently and not compromising accuracy at the same time. Since decades, we are more interested making devices that have cyber capabilities and a better physical impact on the patients. These devices are either implanted inside a patient's body, or the patient wears it. The devices that are implanted are called Implantable Medical Devices (IMD) and which are worn by the patient are called wearable devices. Since medical cyber-physical devices (MCPS)

are context-aware, life-critical, and a networked system of the medical devices, the usage of these devices is increasing in the hospitals for continuous high-quality treatment of patients. These devices can be equipped with the wireless capabilities also, which allows the devices to communicate with each other or the programmer. For example, wearable devices can be controlled remotely by a physician through a smartphone. These devices need to be safe, efficient, and effective as a minor fault in it can be fatal for someone's life [131].

#### 8.2.1.4 Smart Cars

The passenger-carrying vehicles are evolving to become smarter, for which the electronic components that make these vehicles smart are introduced to new models continuously. Smart cars are those cars which are more fuel-efficient, environment-friendly, safe, and have more convenient features. Advanced entertainment units in smart cars are also desired, like advanced music player and even a video player. Besides these, comfort factors like automated tinted glass, window controllers, displays on the screen, cruise control, reverse camera and more are equally important. As this brings new features and benefits, it also brings security concerns. Numerous computers working together make these advancements possible. These computers are called Electronic Control Units (ECUs). ECUs monitor and control various functions of a smart car like brake control, engine emission control, entertainment units, and comfort features.

### *8.2.2 CPS Communications*

Communication is a major part of a cyber-physical system as the cyber and physical components communicate with each other the whole time that makes these systems run properly. Different CPS applications have different communication technologies. They use different protocols and technologies like open and proprietary, wired and wireless. Here we have discussed the common communication technologies used by each of the four applications.

#### 8.2.2.1 Communications in ICS

In ICS, there are two diverse types of communication protocols which are deployed. One of them controls and automates the Distributed Network Protocol (DNP3), Modbus, while the other interconnects the control centers of ICS, for example, Inter-Control Center Protocol (ICCP). These protocols are used in addition to the general-purpose protocols like TCP/IP.

#### 8.2.2.2 Communications in Smart Grid

Smart meters use two types of networks: one is using Modbus and DNP3 in field device communications and the advanced protocol IEC 61850 which is developed by the International Electrotechnical Commission (IEC). The other type of network used is control center communication. This is like ICS as it relies on ICCP. Additionally, smart meters and field devices also use wireless communications for sending measurements and receiving control from the control centers. Smart meters usually use short-range frequency signals like Zigbee to diagnose operations done by the technicians or the readings generated by digital smart readers.

#### 8.2.2.3 Communications in Medical Devices

To avoid surgical extraction, it is necessary to configure and update the IMDs wirelessly, which makes wireless communication the most common method of communication for medical devices. Though, the wearable devices and IMD use different types of technologies and protocols for communications. For example, for the communication with programmers, low-frequency (LF) signals are used by the IMD. These LF signals are specified by the Federal Communications Commission (FCC). This communication through low signals, specified by FCC, is called Medical Implant Communication Service (MICS), whereas the wearable devices use a different type of communication called Body Area Network (BAN). This wireless communication uses numerous wireless communication protocols and technologies like Zigbee and Bluetooth [18].

#### 8.2.2.4 Communications in Smart Cars

There are different types of communications in smart cars, which are Vehicle to Infrastructure (V2I), Vehicle to Vehicle (V2V), and in-vehicle communications. This paper focuses on the latter. As we mentioned above, there are around 70 computers connected in a smart car which are called ECUs. These ECUs communicate with each other through a bus network. These networks are also divided into bus networks which have their bus topology. The messages are exchanged among the subnetworks through a gateway. The gateway separates the messages according to the source and destination of the messages. This is not only for security concerns but also for the bandwidth. The most common protocols which are used are (1) Local Interconnect Network (LIN), (2) Controller Area Network (CAN), (3) FlexRay, and (4) Media-Oriented Systems Transport (MOST). LIN is used for comparatively low-speed applications like shutting windows on/off. CAN runs the soft real-time applications such as antilock braking system. Where the speed of transmission is critical, FlexRay is needed for hard real-time applications. MOST is usually used for in-car entertainment-oriented applications [160]. Some cars are also operated with wireless connections like cellular interfaces and Bluetooth.

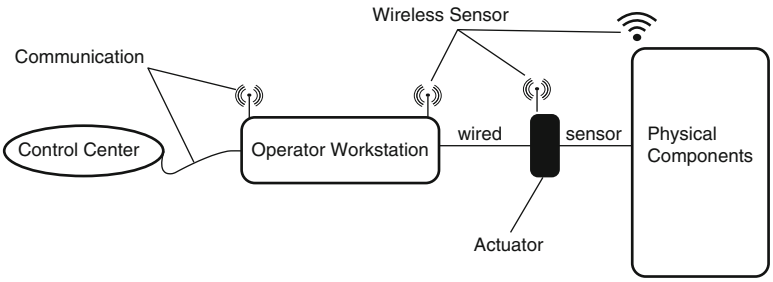



**Fig. 8.1** Aspects of a CPS

### *8.2.3 CPS Models and Aspects*

As shown in Fig. 8.1, there are mainly three categories of components in cyber-physical systems: (1) monitoring and manipulating, (2) computation and control, and (3) communication. The CPS is connected with the high-level systems like control centers and/or lower-level components which exist in the physical world through the wired or wireless communication channel. The intelligence is embedded in the computation and control part since all the control commands are sent, and the sensed measures are received here CPS is connected to the physical system by monitoring and manipulating components through the sensors and the actuators.

A Cyber-Physical System component can communicate with other CPS components or control centers. There are different security implications for each one of these components which may be the result of heterogeneous properties and capabilities of the components. For example, the physical world is not expected to get affected by the cyber world of a CPS, and yet the physical components might be damaged by unexpected attacks which may cause physical consequences. Similarly, the cyber components can also be affected through the communication channel by the exploitation of physical components.

Therefore, different aspects need to be distinguished properly and the respective security analysis must be done separately. The cyber aspects of CPS include those components which do not interact with the physical world directly such as data computations, monitoring communications, communication protocols, etc. Whereas other channels through which cyber world interacts with the physical world and vice-versa are considered as cyber-physical aspects. Finally, the components of a CPS that can be accessed, controlled, or monitored physically come into the physical aspects category.

#### 8.2.3.1 ICS

A scenario of the network of ICS is depicted in Fig. 8.2. The Corporate Network is the cyber part of the ICS which has no direct connection with the physical world. All the ERP servers and production management system are parts of the cyber world. The cyber components have wired/wireless communication with the cyber-physical

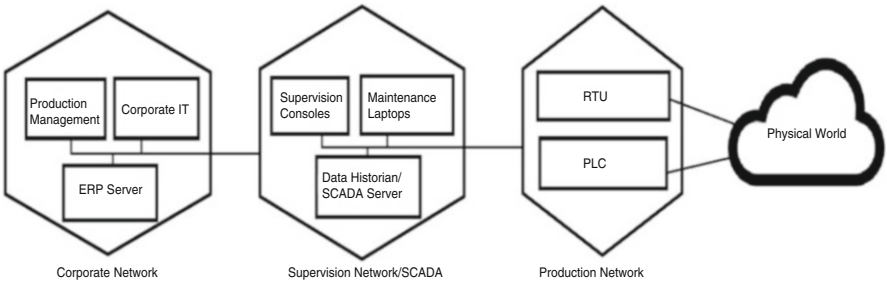



**Fig. 8.2** Aspects of industrial control system

components which, on the other hand, have connections with the physical world, too. In this part, the operations of the physical world are supervised, controlled, and the data regarding their functions, requirement are stored. This is mainly the control and supervisory section which interacts with the physical world through field devices such as PLC/RTU.

As an example, the PLC/RTU is used to control and monitor the temperature of an industrial plant. If the temperature exceeds a certain amount of temperature in any instrument, it notifies the PLC/RTU through the wireless connection and in return the PLC informs the control center about the undesired changes in temperature. As a response, the control center will instruct the PLC/RTU to initiate the cooling system to reduce the temperature.

#### 8.2.3.2 Smart Grid

In Fig. 8.3 the cyber-physical aspects of smart grids are shown. To every house, there is a smart meter attached to provide the utility companies more accurate data of electric consumption. It also makes more convenient for the customers to have a track on their usage. The smart meter, on the one hand, connects the house appliances with the Home Energy Management System (HEMS), and on the other hand, it interacts with the data collector components. Although wireless communication is the most convenient and common way of interaction for data collectors, wired communication is available too, i.e., Power Line Communications. There is a meter equipped with diagnosing port which relies on the short wireless range, and a meter with a diagnosing port which relies on the short wireless range, to make the access more convenient for the digital meter readers and diagnostic tools [71]. The measurements of the smart meter are sent to a collector which forwards those in an aggregated form to the distributed control center which is managed by the utility company. The AMI head-end stores this data and shares it with the Meter Data Management System (MMDS). The MMDS manages the data with other systems like demand response system, billing system, and historians. These connections with high-end sectors can be disconnected by remotely sending commands to smart meters. If many smart meters are sent signals to disconnect with the high-end systems, a large-scale blackout will occur.

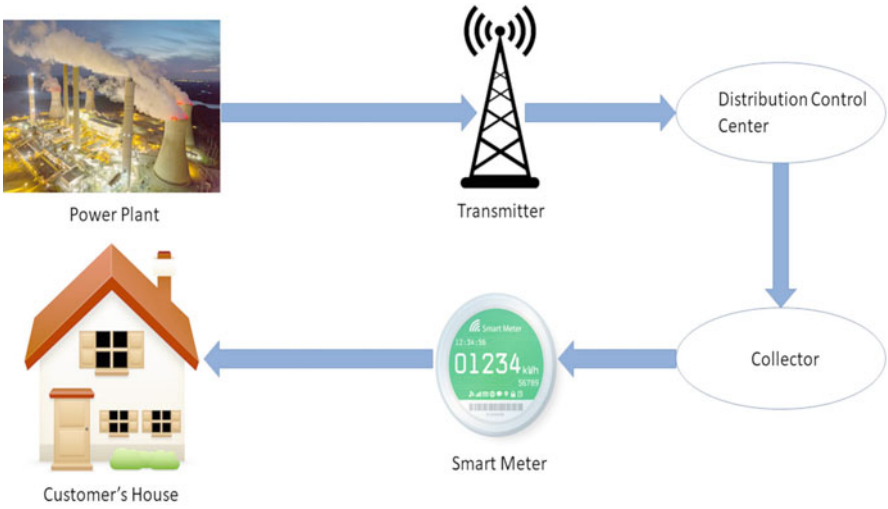



**Fig. 8.3** Aspects of smart grid cyber-physical system

In Fig. 8.3 we have described the overview of aspects of a smart meter. The cyber aspect is present in the control center where the data of the smart meter are stored, shared, and analyzed. The control center can also be a cyber-physical aspect when a connect/disconnect command is sent to the smart meter from high-end AMI. The cyber-physical component is very apparent in a smart meter as it can send data to the utility companies which is a cyber-operation, whereas it can also connect/disconnect electricity services on command which is an example of physical activity. Other field devices also have a high presence of cyber-physical aspects as they interact with the physical components very closely, such as the devices in the generation, transmission, automation, distribution, etc. The amount of energy used by any home appliance can be controlled by the utility companies based on the time when it is needed, which is a cyber-physical activity [113].

#### 8.2.3.3 Medical Device

Figure 8.4 portraits the networking of two of the most important Inter-operable Medical Devices (IMD)s—the Implantable Cardioverter Defibrillator (ICD) and the insulin pump. If a rapid heartbeat is detected, an electric shock is delivered to maintain a normal heartbeat rate. The role of an insulin pump is to inject insulin into the diabetic patients automatically or manually when its needed [52]. To get the measurement of blood sugar, the insulin pump uses Continuous Glucose Monitor (CGM). As both the devices have small needles which are injected into patient's body, the CGM sends the received blood sugar measurement from the patient to the insulin pump or other devices like a computer or any remote-control devices through wireless signals, and the pump decides whether it should inject the insulin or not based on the measurements. The remote-control devices are usually held by the

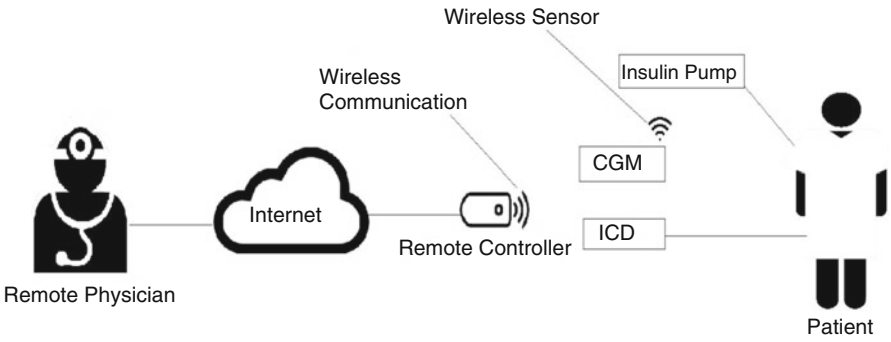



**Fig. 8.4** Aspects of medical cyber-physical system

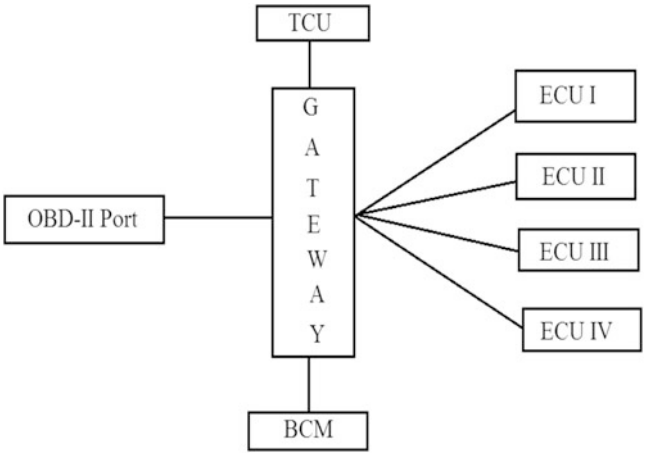



**Fig. 8.5** Aspects of smart car cyber-physical system

patient or the doctor. In the fig, four cyber aspects are embodied in the monitoring computers, whereas the cyber-physical aspects are those devices which directly interact with the patients' implant devices. In the end, the patient is considered as the physical aspect.

#### 8.2.3.4 Smart Cars

Figure 8.5 demonstrates the architecture of the network in a smart car. Each of the Electronic Control Units (ECUs) is connected to their respective sub-network according to the tasks expected from those. Each of the ECUs can interact with each other through gateways. In this chapter, we mainly discuss the Controller Area Network (CAN) bus. Behind this, there are primarily two reasons: (1) Maximum number of security issues are generated from CAN-based networks, and (2) since 2008 every vehicle in the United States requires CAN to be installed in it, so almost every car around us possesses it.

Figure 8.5 shows various aspects of a smart car network. The cyber aspects are held by those ECUs which don't communicate with the physical components directly, such as the Telematics Control Unit (TCU) and the media player. The TCU has wireless interfaces which allow remote software update by the car manufacturer, phone pairing, hands-free facilities which can be used by a connected smartphone. The ECUs which interact with the physical components such as the Remote Keyless Entry (RKE) and parking assist are considered as the cyber-physical aspect. The RKE sends wireless signals to have a physical impact on the car such as locking and unlocking. Finally, the engine, tires, etc. are the physical aspects.

### *8.2.4 Security in CPS*

We have motivated the importance of security in CPS in this section. We have illustrated four different examples based on this. Usually, security systems are linked with mechanisms like cryptography, intrusion detection, access control, and many other solutions which are used in IT field. Those mechanisms play a key role in securing the information and communication in the architecture. The reported attacks on the Cyber-Physical Systems are the result of sole dependency on these mechanisms as discussed in Sect. 8.5. Hence, the solution which takes these aspects into consideration is required and can be used along with the IT security solutions.

#### 8.2.4.1 Security in ICS

If the security of a CPS is compromised, it can bring a catastrophic consequence. For example, if a nuclear plant's security is breached, the result can be a worldwide threat. On the other hand, if a smart grid is violated, it will stop the services to the consumer and bring economic loss to the utility company since the use of CPS is very wide and its pervasiveness and the security of CPSs are very critical. In fact, even now it is advised that the ICS should not be connected to the internet because of the inherent security vulnerabilities which can bring possible catastrophic consequences [44].

#### 8.2.4.2 Security in Smart Grids

Inadequate security in smart meters creates the threat by remote attackers which could evolve to extensive blackout. The extended results of this could be data loss, malfunctioning of medical devices, and even an increased crime rate. Another possible result can be an ability of attackers to reveal customer's information.

#### 8.2.4.3 Security in Medical Devices

Security in IMDs and wearable devices makes them immune to the attacks which can compromise patient's privacy and safety. There are other different circumstances around the medical devices which mandate the need for definite security in them. The security goals of the medical devices, integrity, confidentiality, and availability, are initiated by Halperin et al. [53]. The security goals must allow the entities to access accurate data, configure identified data, update software, and maintain the availability of the devices. On the other hand, privacy deals with the security of the private information of the devices, such as their type, unique ID along with patient's information.

#### 8.2.4.4 Security in Cars

Car Manufacturing companies always thrive to come up with innovative ideas such as how to enhance the comfort and functionalities of the vehicle, etc. However, the safety issues are not concerned in the design phase. The safety of cars means their ability to function accurately in different favorable and adverse situations. The advanced features of a smart car like wireless communications and components escort more security vulnerabilities with which arise the importance of security issue.

## 8.3 CPS Security Threats

To secure a CPS, there are various challenges which need to be considered. Before securing a system, we must understand the possible threats upon that. We have analyzed different angles of potential threats on CPS in this section. At first, we have reviewed the general threats which are vulnerable to almost any CPS application. Then, we deal with various threats that are distinct with different CPS applications.

### *8.3.1 General CPS Threat Model*

The knowledge of from whom to secure a CPS is as important as the knowledge of its existing vulnerabilities and security mechanism. The definition of a security threat is *"a set of circumstances that has the potential to cause loss and harm"* [120]. The loss from an attack might be in confidentiality, safety measures, integrity, or availability of resources. On the contrary, the harm refers to harming people, systems, or the environment. We have identified the five factors which every threat has: source, target, motive, attack factor, and potential consequences.

- **Source**: Source of attack is where the attack initiates from. It is categorized into three types: Adversarial threats, accidental threats, and environmental threats.
- **Target**: Targets are on which the attacks have been made. In this paper, the targets are the CPSs and their components.
- **Motive**: The reasons behind the attack made. It could be political, personal revenge, criminal, terrorist, spying, or cyberwar [138, 152].
- **Attack Vector**: A successful attack must have one or more of the four mechanisms: interception, interruption, modification, or fabrication [120].
- **Consequence**: The outcome of a successful attack on a CPS, such as lose of confidentiality, data integrity, privacy, availability, or safety.

### *8.3.2 CPS Security Threats*

We consider the potential threats to each of the four applications of CPS using the proposed threat model. The threats specific to each application have been emphasized on the five factors of attack: source, target, motive, attack vector, and consequence.

#### 8.3.2.1 Threats Against ICS

- **Politically influenced cyberwar (motive):** A cyberwar could be initiated by a hostile nation (source) with another nation (target) by remotely attacking on the nation's critical infrastructure like nuclear plant, by injecting malware or controlling their field devices (vector) which can result in a complete shutdown of a plant, huge economic loss, or polluting the environment (consequences) [8, 75, 132, 149].
- **Politically influenced espionage (motive):** A party or even a nation can use their intelligence agencies (source) to attack any rival's critical infrastructure (target) by spreading malware in their system (vector) to access or have their critical confidential data (consequence) [101, 110].
- **Physical Threats (motive):** A sensor of any environment (target) can be attacked by an attacker (source) to falsely increase or decrease the temperature of the environment (vector) resulting in control center receiving false measurement (consequence).
- **Financially Motivated Threats (motive):** A skillful customer (source) can tamper with the utility physical devices or inject false data (vector) to reduce the utility bill which will affect the utility company (target) and lose financially (consequence) [150].
- **Criminal Attackers (motive):** A capable attacker who is familiar with a system (source) could utilize the wireless connection (vector) to control a CPS application (target) to interrupt its operations (consequence).

#### 8.3.2.2 Threats Against Smart Grids

- **Financially Influenced Threats (motive):** A customer (source) who wants to fabricate his utility data in smart meters (vector) which will result in reduction of his utility data (consequence) made by the utility company (target) [4, 97, 98, 108, 127]. On the other hand, to advertise their product, the utility companies (source) look into the private information of their customers (target) by looking into their usage and house appliances (vector) which result in privacy violation [24, 97, 123, 143]. An attacker (source) can also have control over many smart meters (target) by injecting malware (vector) and extort money in exchange for not shutting them down which will result in a wide blackout (consequence) [4].
- **Criminally motivated threat (motive):** A capable robber (source) can consider a person's smart meters (target) to access the utility measurements (vector) by which he can predict that whether the person is home or not to conduct a successful robbery (consequence) [143].
- **Politically motivated threat (motive):** An organization (source) can gain remote access to a set of smart meters (vector) to cause a blackout, economic loss, or any disturbance (consequence) to its rivals (target) [97].

#### 8.3.2.3 Threats Against Medical Devices

- **Criminal motivated threat (motive):** An Attacker (source) can inject modified data or retransmit the previously given command (vector) to the medical devices attached to the patient (target) to cause harm to the patient's health condition (consequence) [53]. Additionally, the attacker (source) can also jam the wireless communication of the medical device and the control center (vector) which are responsible for maintaining the desired health condition of the patient (target) will fail to conduct the accurate operations (consequence) [53, 54, 131].
- **Spying Motivated Threat (motive):** An attacker (source) can intercept the communications between the medical devices (vector) of a patient to access the confidential information of the patient (target) which results in privacy violation (consequence) [53]. On the other hand, medical devices send this information to the other control centers (target) also which contain a huge set of confidential information of numerous patients (motive) which can be accessed by a hacker (source) by intercepting the wireless communication (vector) which also results in privacy violation (consequence) [82].
- **Politically motivated threat (motive):** To harm a nation (target) politically, a hostile nation (source) can interpret and control the wireless communication of the medical devices (vector) of a political leader which can cause tremendous harm to him or even death (consequence) [153]. US Vice President Dick Cheney had the wireless communications of his pacemaker disabled as he was aware of his possible assassination [131].

#### 8.3.2.4 Threats Against Smart Cars

- **Criminal motivated Threats (motive):** A criminal hacker (source) can attack a car's ECUs (target) by utilizing the weakness in the wireless communications of the ECUs (vector) to cause a malfunction in the ECUs or an accident (consequence) [15].
- **Privacy invasion Motivated (motive):** A hacker (source) can hack into the TCU (vector) of a car (target) to listen to the private conversations going inside the car [15].
- **Tracking motivated threats (motive):** A law enforcement agent or a hacker (source) can hack into the GPS system (vector) to track the car (target) which is an example of privacy invasion (consequence) [7, 15].
- **Profiling motivated threat (motive):** Car manufacturing companies (source) can look into the detailed car logs stored in the ECUs of that particular car (vector) to gather detailed information about its usage and if it has violated any traffic rules or not (target), without the consent of the car owner which is a violation of privacy (consequence) [7, 60].
- **Politically motivated Threats (motive):** An unfriendly nation (source) can attack another nation's transportation system (target) by hacking into fully remote-control cars (vector) to cause large-scale accidents (consequence) [15].

## 8.4 CPS Security Vulnerabilities

We first identify the existing vulnerabilities in a cyber-physical system. Then, on each of the applications, the vulnerabilities are highlighted as different applications may have different kinds of vulnerabilities. Hence, for the suitable solutions, the generic and application-specific vulnerabilities need to be distinguished.

### *8.4.1 Causes of Vulnerabilities*

- **Isolation Assumption:** Usually the designing dependable and safe modules are focused, and the security factor is not emphasized much [93]. The logic behind this is if the system is isolated from the outside world, then automatically it is secure from outside attacks. For example, the security of ICS and Smart Grids relies on the assumption that the systems are isolated from the outside world, and they are controlled and monitored locally [11, 32, 92]. Even, the IMDs were also secured based on the assumption that they are isolated from the outside world [53] the same as the ECUs in smart cars [78]. But the ongoing development in CPS applications is no more restricted to the isolation concept. In fact, they are introducing more connections. With more connections, more access points are getting introduced to those systems which is making them more vulnerable.

- **Increased Connectivity:** With time, the connections inside a CPS are increasing at a significant rate. Unlike before, they are no more isolated from the external environment. Nowadays, they mostly rely on the open networks and wireless technologies. For example, control centers which are directly connected to the ICS and Smart Grids are also connected to the Internet which increases the chance of external attacks. In fact, it has been analyzed that a maximum number of attacks to an ICS was made locally until 2001, but after that most of them originated from outside sources [8]. This is a definite result of more connectivity. Some field devices are directly connected to the Internet for fast incident responses and more convenience which make them more vulnerable [85, 141].
- **Heterogeneity:** A CPS is built with different kinds of components which are manufactured by different entities. For example, COTS, proprietary, and third-party components are integrated to build a CPS application. Hence, a CPS is a result of more integrated such components rather than designed [32]. Each of the components has its security problems. This integration of those components invites respective inherent vulnerabilities [3]. For example, to access a computer running on Windows OS, one of the steps of Stuxnet attack was to harness the Siemens PLC's default password [101].

## *8.4.2 Vulnerabilities in ICS*

### 8.4.2.1 Cyber Vulnerabilities

- **Communication Vulnerabilities:** Although there have been many studies on the security issues of the TCP/IP and ICCP like popular protocols, they still have security issues as the design of these protocols was not intended to be secured [5, 56, 84]. Besides, the ICCP interconnects the control center, but it lacks basic security measure like authentication and encryption [114]. The well-known Stuxnet Attack uses the security vulnerabilities in Remote Procedure Call (RPC) protocol [99]. Besides this, the RPC has many more vulnerabilities which are used by the attackers to tamper with the ICS. The ICS which uses wired communications usually depends on the fiber optic and Ethernet. Since Ethernet uses the local area network and as a result, the components of an ICS are connected through the same medium, it is vulnerable to a Man-in-the-Middle (MITM) attack [43]. As an example, if attackers manage to access the connection between the components, they can easily intercept and manipulate all the data [118, 156]. In an ICS plant, usually, short-range wireless communications are performed under the assumption that no outsiders can get access to the communication medium. However, a malicious insider can still capture, analyze, or manipulate the traffic or even a well-skilled outsider can break into it [27]. Moreover, if employees connect their devices which are probably unsafe, to the wireless network, an outsider can use their Internet-connected devices as a

vector and get into the system [42]. Long-range communications like satellite, microwave, and others are also used in ICS, but their vulnerabilities in the context of ICS have not been studied yet. It concluded that wireless communications are more susceptible to cyber-attacks like unauthorized access, active and passive eavesdropping, replay attacks, and much more discussed rigorously in the literature as in [68, 157].

- **Software vulnerabilities:** The unauthorized access to the database where confidential data are stored is one of the most popular software-related vulnerabilities [118, 168]. Emails are also contributed to spread malware. Many attacks which exploited emails are demonstrated by experiments in [39]. To gain access to a secure ICS network, an attacker usually exploits the Internet connections of the devices (e.g., laptops, smartphones, tablets, etc.) which are connected to the desired network [13].

#### 8.4.2.2 Cyber-Physical Vulnerabilities

- **Communication Vulnerabilities:** For sending control commands from the control center to different components, ICS uses protocols like Modbus and DNP3. These protocols are used for monitoring also. The de facto standard for communication in the Modbus protocol lacks the basic standard of security like encryption which creates vulnerability for eavesdropping [1, 9, 29]. The lacking integrity in de facto makes the data integrity uncertain [9, 39]. Even, the authorization measures are not feasible enough which may result into the controllers receiving false data or manipulated data could be sent to the actuators [149, 168]. These vulnerabilities are caused by the DNP3 protocol also as it lacks these security measurements like encryption and authorization [31, 61]. At least 23 attacks were made using DNP3's vulnerabilities, such as lack of authorization, authentication, and encryption, which was analyzed by East et al. [31]. If the primary communication between the field devices such as PLCs, RTUs and the control centers is failed, usually there is a secondary connection which is directly connected (e.g., dial-up) to the sensors and the actuators [1]. It makes the system easy to be breached as the attacker does not need to exploit any other advanced communication.
- **Operating System:** In ICS field devices like PLCs and RTUs, the operating systems which are used are Real-Time Operating System, which does not have access control measure. As a result, all users get root access which is the highest privilege. This makes the devices vulnerable to miscellaneous attacks [168]. If the operating systems have vulnerabilities, the systems running them become vulnerable also which in return makes those devices the vector of an attack on the field devices. For example, the Stuxnet attack exploited the vulnerabilities in two operating systems. The first one was exploited in the Print Spooler Service which is a vulnerability in remote code execution over Remote Procedure Call (RPC) [100]. As a result, Stuxnet could copy itself to the vulnerable computer [17]. Similar to this, the other one was Windows Server Service which also used

remote code execution in which specially crafted RPC request was sent [99], and this makes the Stuxnet capable of copying itself to other computers [17]. Some attacks use the buffer overflow vulnerability in the operating system used at the control center [149, 168]. Among the most used vulnerabilities in the operating system, buffer overflow is the most popular according to ICS-CERT [62].

- **Software Vulnerabilities:** Among the programs which are used in the general-purpose operating system to monitor and control the field devices, WinCC is popular which is a Siemens product and used to control PLCs. The Stuxnet attacks the vulnerable computers which run WinCC. At first, the Stuxnet copies itself to the vulnerable computer and then installs a DLL file in the system which is used by both WinCC and the PLC. It allows the DLL to send rogue codes to the PLC. Lack of digital signature is the main vulnerability which allows this critical action [77]. As we discussed earlier, one of the main reasons of increased vulnerabilities is the presence of COTS in CPS, and an example of this in 200 PLC models is revealed in [85].

#### 8.4.2.3 Physical Vulnerabilities

The physical exposure of the ICS field devices, such as PLCs, RTSs, is vulnerable as they can be tampered or stolen due to lack of physical security. For example, a water canal had solar panels as its source of energy which were stolen and as a result, the control center lost all critical data of that canal for necessary operations [2].

### *8.4.3 Vulnerabilities in Smart Grid*

#### 8.4.3.1 Cyber Vulnerabilities

- **Communication Vulnerabilities:** The information infrastructure of a smart grid relies on certain protocols. The smart Grid's components use TCP/IP for the general-purpose Internet. As the vulnerabilities in TCP/IP are known, it is not used for the connection to the control center. However, sometimes due to misconfiguration accidentally the control center gets directly/indirectly connected to the Internet which itself is very vulnerable [25]. ICCP is used for the communication among control centers, but it contains critical buffer overflow vulnerabilities [168].
- **Software Vulnerabilities:** Along with the same software vulnerabilities present in the ICS, smart meters contain some more. Since the widely spread smart meters can be controlled remotely, it provides certain vulnerabilities for the attackers to exploit as they can control smart meters from either the meters individually or the control center. If injecting a software bug into one of the components in a smart grid is used as a vector, creating a wide blackout will become feasible [4]. More accessible smart meters in every household provide

more access points to the attackers [108] which sometimes are used as backdoors. Santa Marta [135] discovered such backdoor to a smart meter which could allow a capable customer to gain full control over the meter including modification of the utility bill. Additionally, there is a protocol named TELENET which can be used to connect the smart meter. This protocol can be exploited to perform multiple coordinated attacks on different smart meters in a grid.

- **Privacy Vulnerabilities:** The smart meters at the households and the utility companies have two-way communication which creates vulnerabilities. An attacker can intercept the traffic of vast data generated by the smart meters which will compromise the privacy of the customer [21]. Furthermore, they can access the information about the presence/absence and daily habits of the residence.

#### 8.4.3.2 Cyber-Physical Vulnerabilities

- **Communication Vulnerabilities:** The power grid infrastructure has the same kind of vulnerabilities as ICS in the context of the same protocols used in ICS, i.e., Modbus and DNP3. Smart grid has some additional protocols like IEC 61850 which provides some advanced communication between the substations. However, these advanced protocols do not have adequate security measures. Some of these could not provide encryption which creates the vulnerability for eavesdropping which will provide the details of customer's usage pattern to the attacker [96, 108]. The protocols which do not have significant authentication measure could be exploited to inject false data [124, 156]. One more result of this is the over-flooded network by injected bogus data which is an example of DoS attack [143, 162]. The heterogeneous components of a smart grid also create vulnerabilities. The generation plant of a smart grid communicates with the transmission plant which interacts with the distribution sector and this sector delivers the power to the customers. Each of these sectors has their security protocols which integrally are vulnerable to various kind of attacks due to lack of proper communication and collaboration [36, 58, 108].
- **Smart Meter Vulnerabilities:** As discussed earlier, due to two-way communication in smart meters components, many access points are created for the attackers to intercept the interactions [69]. This creates a backdoor in home appliances which could become an entrance for an attacker to the control center. The documentation of a smart meter was analyzed by Santamarta, and a "Factory login" account was discovered [135]. Unlike regular customers' accounts, this account would give any user complete privilege over the device. This may result into (1) Disrupted power supply, (2) attacks to other smart meters in the same network, and (3) tampered data of the collected data such as the utility bill [135].

#### 8.4.3.3 Physical Vulnerabilities

The field devices of smart grid face similar problems like ICS' components. As they are widespread the physical security to these field devices is inadequate [140]. They are vulnerable to the physical destructions. For instance, the power lines can be easily damaged by malicious, natural, or accidental causes. Once 50 million people suffered from large blackout due to the power line cut by overgrown trees [149].

### *8.4.4 Vulnerabilities in Medical Devices*

#### 8.4.4.1 Cyber Vulnerabilities

- **Obscurity Vulnerabilities:** Some medical device manufacturing company relies on the secrecy of the designing proprietary protocols due to lack of their security measures [82]. This is known as "security through obscurity." Although this has never been enough to thwart the attackers.
- **Communication vulnerabilities:** Medical devices usually communicate with their programmers through a wireless connection which is exploited for different kinds of attacks, such as injection, eavesdropping, replay attacks, and much more. Lack of encryption in the security measures allows replay attacks [52] along with the loss of confidentiality since the ICDs interact with their programmers through wireless channels. Also, patients wearing devices or with IMDs are vulnerable to privacy invasion-related attacks. Moreover, the patient can be tracked down if the unique information of the devices is inferred [53].
- **Software Vulnerabilities:** Along with the growing role of software in the medical devices, their vulnerabilities have also increased. Hence, more devices are recalled due to software-related defects [46, 54]. Since the role of these devices is to monitor and control the health of patients, a simple flaw can result in a critical health situation. The security analysis of the medical devices was publicly analyzed by Hanna et al. [54] for the first time. They discovered that a medical device named Automated External Defibrillator (AED) has four vulnerabilities such as (1) random code execution because of buffer over flow, (2) improper storage for credentials, (3) inadequate authentication mechanism, (4) firmware update without any authorization due to improperly deployed Cyclic Redundancy Check (CRC). Also, Li et al. [87] also showed that how a random CRC check in code can lead to various dangerous attacks like replay attack, unauthorized injection of data, and sending out unapproved commands.

#### 8.4.4.2 Cyber-Physical Vulnerabilities

- **Communication Vulnerabilities:** The dependency of medical devices, like wearable devices and IMDs, on wireless communication invites more vulnerabilities. If a medical device fails to send or receive accurate data, the patient will suffer from an undesired health condition. The chances of jamming attacks, replay attacks, eavesdropping increase. As an example of jamming attack, if an insulin pump does not receive the periodic updates from the Continuous Glucose Monitoring (CGM) device associated with the patient, the pump may not decide the accurate amount of insulin that needs to be injected, which may cause improper health conditions [87, 125]. Some attacks on computational or communication devices drain out the battery resource and the devices fail to communicate [52, 131]. Another vulnerability is, by exploiting wireless communication, the attacker could inject specially crafted data which result in undesired operations by the medical devices. Halperin et al. [52] and Gollakata et al. [48] explained that the wireless communication vulnerabilities held by ICDs could be utilized for injection attacks. Moreover, Li et al. [87] demonstrated that by intercepting the wireless communication of the insulin pump with its remote control, an attacker could gain the ability to control the device remotely. For authorization, the injected package requires the serial number of the device. Radcliffe et al. [125] showed that an attacker who retrieved the serial number could inject an unauthorized inappropriate command to the device. For the replay attack, the attacker does not have to be knowledgeable about the underlying protocols. All he has to do is capture legitimate command packets which can be retransmitted later. Li et al. [87] showed that the vulnerable insulin pump which allows replay attack may receive incorrect readings of the glucose level which will command the insulin pump to inject wrong amount of insulin to the patient which will cause undesired health problems. It was revealed by Radcliffe et al. [125] that CGM is also vulnerable to replay attacks.
- **Device Authentication:** As a result of implied trust to everyone using commercial programmer, an attacker without any technical knowledge can use those without any authorization [52]. Halperin et al. [52] showed that even Universal Software Radio Peripheral (USRP) is capable to replace a programmer and deliver malicious packet.

#### 8.4.4.3 Physical Vulnerabilities

Both wearable and implantable devices are vulnerable to physical attacks. For example, an attacker, who can get close to the medical devices, can tamper with those and inject malicious commands which will cause undesired health problems for the patient. The serial number of the device will also be revealed which is convenient for other attacks. Thus, there should be adequate physical security around the medical devices as recommended by Hanna et al. [54]. Nonetheless,

since the designer of the devices cannot control the surroundings of the patients' wearing those, the patients, along with the devices, are vulnerable to physical attacks in an insecure location. These types of attacks are usually politically motivated [153].

### *8.4.5 Vulnerabilities in Smart Cars*

#### 8.4.5.1 Cyber Vulnerabilities

- **Communication Vulnerabilities:** To enable hands-free operations in a smart car and to provide car's manufacturing company the control to do certain operations remotely like software update, crash report and stolen car recovery, etc., cellular interfaces are used. These cellular communication channels are provided to the cars by TCU. Since Global Positioning System (GPS) and microphone are parts of these TCU, major security concerns regarding TCU are arising. The connection can be used as a vector to track down the vehicle or even become a spying tool for eavesdropping on the conversation going inside the car [15, 155].

  Among the vulnerable vectors used for an attack on a smart car, Bluetooth is the most important one [155]. To pair a device with the smart car via Bluetooth, the Telematics Control Unit (TCU) generates a PIN which has to be entered for authentication. However, this measurement is not enough since an attacker can brute-force the PIN or even inject a modified PIN by faking the Bluetooth Software. If the Bluetooth's Media Access Control (MAC) is extracted by the attackers, the car will be vulnerable to the traceability attacks as the MAC is unique and traceable [15].
- **Software Vulnerabilities:** Smart cars are the result of integration of different types of ECUs which are operated by different software. The reliance of smart cars on the ECUs has increased rapidly and as a result the possibility of a software bug, and other vulnerabilities, in the ECUs have also escalated [59]. Malicious code, injected in a software running ECU, can expose the entire car to various types of attacks. Jo et al. [66] showed that how a TCU running on Android Operating System was exploited to unlock the doors and even the GPS was tracked due to the vulnerabilities in the software. In fact, if the media player has the vulnerability it can be exploited to attack other ECUs as the player has the ability to connect to the CAN bus directly. As identified by Checkoway et al. [15], the media player can be used as vector to attack the other ECUs by injecting a malicious code installed CD and the player is vulnerable to other arbitrary codes because of its ability to resolve different media files.

#### 8.4.5.2 Cyber-Physical Vulnerabilities

- **Communication Vulnerabilities:** Due to inadequate security mechanisms, smart cars are vulnerable to different types of attacks [78]. Among the in-car communication protocols, Controller Area Network (CAN) and Local Interconnect Network (LIN) are used the most. However, we will review the vulnerabilities of CAN since it is used more than LIN. CAN lacks proper encryption, authentication, and authorization measurements. Due to lack of encryption, Tire Pressure Monitoring System (TPMS) is vulnerable to attacks like eavesdropping, data injection, and spoofing [130]. Using the unique ID stored in the TPMS, tracing a car becomes possible. Even more, the broadcasting nature of CAN creates vulnerabilities for DoS attacks [73]. DoS exploits the error handling mechanism of CAN bus network [22]. Another vulnerability in the security property is non-repudiation which makes it difficult to trace the source of the attack [60].
- **Vulnerabilities in ECUs:** The ECUs are getting equipped with advanced technologies for the betterment of safety and comfort. ECUs like Collision Prevention, Adaptive Cruise Control, etc., provide safety. And Comfort is delivered by the ECUs like RKE and Comfort Part Assist. These are all parts of AN network which is vulnerable to many attacks. For example, if a ECU is attacked then the attacker can also attack the other vulnerable ECUs in the same network at the same attempt [155]. ACC is the next-generation Cruise Control Driving. If the ECU is tampered externally or even internally by exploiting other vulnerable ECUs such as RKE, TPMS, and TCU, incorrect data can be provided to the ACC, as a result the car will change its speed unexpectedly which can turn into a collision.
- **Vulnerabilities in X-by-wire technology:** Nowadays, there is a trend called X-by-wire in smart cars. This technology replaces the components like steering, brakes which are controlled mechanically by electronic and electromagnetic components. It allows the driver to control the mechanical and electromechanical components by pushing some buttons. For example, Steer-, Shift-, Break- are used in this trend [145]. However, this implies more threats to the vehicle. This technology counts on FlexRay protocol which is very costly and doubtful to be widely used in the near future [145].

#### 8.4.5.3 Physical Vulnerabilities

A car is physically vulnerable to various types of attacks. Such as if the TPMS part of a car is destroyed, the designated ECU will not receive the air pressure from the TPMS. A mechanic has physical access to a car. This opportunity makes many internal parts directly accessible to an attacker through OBD-II port [160]. Even the exterior mirror is exploited sometime as the backdoor to the internal parts [60].

## 8.5 Real-World CPS Attacks

In this section, we have reviewed the attacks reportedly made on each of these applications, using the vulnerabilities discussed in Sect. 8.4. The attacks have been categorized into cyber, cyber-physical, and physical attacks based on the locations of the damages made by these attacks. The attacks which do not have impact on the actuators/sensors are considered as cyber-attacks while the attacks which directly strike the physical components are contemplated as physical attacks. Finally, the attacks which hit the physical components exploiting cyber components are considered as cyber-physical attacks. Generally, the attacks which are publicly known are very rare [121] and to find the attacks utilizing all the above-mentioned vulnerabilities is infeasible. With the brief review of real-world attacks on CPS applications, we have also provided four different taxonomy proposed by Yampolskiy et al. [164] based on attacks on ICS, Smart Grids, Medical Devices, and Smart Cars. Tables 8.1, 8.2, 8.3, and 8.4 demonstrates the real-life attacks on ICS, Smart Grids, Medical Devices, and Smart Cars, respectively. Here are some brief definitions of items used in the taxonomy:

- **Influenced object:** The object on which is attacked.
- **Influence:** The resulting changes on the attacked object.
- **Affected Elements:** Elements which got affected indirectly.
- **Impact:** Changes in the entire CPS.
- **Method:** The way the attack took place.
- **Precondition:** The pre-attacks made to make the attack successful.

### *8.5.1 Attacks on Industrial Control System (ICS)*

#### 8.5.1.1 Cyber-Attacks

- **Communication Protocols:** Most of the attacks on ICS are made exploiting the vulnerabilities in communication protocols. An example of Address Resolution Protocol being spoofed was demonstrated on Supervisory Control and Data Acquisition (SCADA) system [42, 151].
- **Espionage:** Many attacks have been made to the ICS with motive of spying. DuQu and Flame are two examples of these kind of attacks [20, 110]. Flame had targeted many ICS in the Middle East world and was not discovered till 2012. The objective of this malware is to collect private data like emails, network traffic of the corporations [110]. Similarly, a group of hackers called Dragonfly attacked many corporations in the USA and Europe in 2003. Their motive was to collect classified information of those corporations. They sent fishing emails containing malware PDFs to the employees of those organizations. After opening these emails, the vector changed into water hole vulnerabilities which

**Table 8.1** ICS cyber-physical attacks

| Name | Influenced element | Influence | Affected element | Impact | Method | Precondition | Reference |
|---|---|---|---|---|---|---|---|
| Web-based attacks and field devices | Web interface of the field devices | Components which are controlled by the attacked devices | Field devices become unable to be controlled properly by the authorized users | Disable the connection of the control center to the field devices | The Internet connections of the devices are exposed. | | [150] |
| Web-based attacks and field devices | Web interface of the field devices | Components which are controlled by the attacked devices | The configurations of the field devices are lost | Inject malicious code into the devices | Induced vulnerabilities into the TCP/IP protocol due to COTS implementation | | [150] |
| Maroochy | Sewage pumps | Malfunction in the pumps | The configuration of the pump station | Polluted environment and monetary loss due to sewage flood | The configurations of the pumps are manipulated | Well informed insider who is familiar with the system | [141] |
| Modbus worm | Communication network of ICS | Damage to the communication network | Components connected to the network | False data injection and even rebooting the entire system | Inject malicious code into the system | Communication traffic of ICS is accessible | [39] |
| Stuxnet | Centrifuges of PLCs | Exaggerated rotation of centrifuges | Rotors attached to the centrifuges | Physically damaged for a long term | Unauthorized commands sent to the centrifuges from the PLCs | Installation of Stuxnet in PLCs | [17, 77, 111, 164] |

**Table 8.2** Smart grid cyber-physical attacks

| Name | Influenced element | Influence | Affected element | Impact | Method | Precondition | Reference |
|---|---|---|---|---|---|---|---|
| Cyber extortion | Smart meter attached to the household | No power supply to the household | Residence of the house | Extortion in exchange for power supply | Control the smart meters through its Internet connection | Intercepted two-way communication with the utility company through Internet | [112] |
| Aurora experiment | Circuit breakers of generators | Explosion of the generator | Power Generators and utilities fed by those generators | Power cut due to explosion of the generators | Rapidly open and close circuit breakers | Ability to communicate with the device | [134, 165] |
| False data injection | Smart meters | Manipulated billing information | Utility companies | Financial loss | Inject false data to the smart meters installed in the household | Physical access to smart meters | [156] |
| Theft | Metal and copper wire | Theft of equipment | Smart grid | Disconnection of the field devices | Steal the metallic parts from the field devices | Physical access to the field devices | [122] |
| Jamming attack | Wireless communication layer | Delay of communication | Smart grid | Malfunction of various components | Inject many unnecessary unauthorized packets in the communication channel | Access to the communication channel. | [90, 91] |
| Vandalism | Transformers | Damage to transformers | Smart grid | Wide blackout | Damage transformers | Physical exposure of transformers | [41] |

**Table 8.3** Medical devices cyber-physical attacks

| Name | Influenced element | Influence | Affected element | Impact | Method | Precondition | Reference |
|---|---|---|---|---|---|---|---|
| DoS | A certain medical device | Shutting down of the device or inability to function accurately | Patients with those attached medical devices | Unhealthy condition of the patient due to miss-treatment | Replay the previously intercepted commands | Intercept the commands sent to the attached medical devices from the control device | [52] |
| Replay attack | Continuous glucose monitoring (CGM) device | Incorrect measurement of glucose delivered to the pump | Patients along with the attached insulin pumps | Delivery of incorrect amount of insulin to the patient | Replay the previously intercepted commands | The communication between the CGM and insulin pump gets intercepted | [125] |
| False data injection | Insulin pump | Manipulated data transferred to the insulin pump | Patient's wrong treatment | Patients' health conditions | Imitate the CGM to send the inject false data to insulin pump | The communication between the CGM and insulin pump gets intercepted | [87] |
| Unauthorized commands injection | Insulin pump | Improper actions by the insulin pump | Patient's wrong treatment | Patients' health condition | Inject unauthorized commands to the insulin pump | The communication between the insulin pump and its remote controller gets intercepted | [87] |

**Table 8.4** Smart cars cyber-physical attacks

| Name | Influenced element | Influence | Affected element | Impact | Method | Precondition | Reference |
|---|---|---|---|---|---|---|---|
| DoS | Body control module | Random drop in speedometer | Instrument panel cluster (IPC) | The whole IPC freezes | Disabling the communication of CAN from/to the BCM | Physical access to controller area network (CAN) bus | [73] |
| DoS | Windows of car | Unable to control open/close windows | ECUs connected to the windows controlling ECU | Passengers' safety is compromised, and discomfort | Send back previously eavesdropped packet, and reverse engineering | Physical access to controller area network (CAN) bus | [60, 73] |
| Malware injection | Bluetooth ECU | Ability to connect to other ECUs | Transmission control unit (TCU) with other ECUs with cyber and physical capabilities | The entire vehicle safety | Exploiting Bluetooth ECU, send malware to another ECU | Control over the Bluetooth paired devices | [15] |
| Malware injection | Telematics control unit | Ability to track the vehicle | Other ECUs which are connected to the TCU through cellular channel | Safety of the entire vehicle can be compromised due to possible large-scale accidents | Connection of the Bluetooth paired device is exploited, and a malicious payload is sent to the TCU | Vulnerabilities in the connection of the Bluetooth paired device | [15] |

(continued)

**Table 8.4** (continued)

| Name | Influenced element | Influence | Affected element | Impact | Method | Precondition | Reference |
|---|---|---|---|---|---|---|---|
| Malware injection | Any specific ECU | Packets containing malware are spread to the other ECUs | CAN bus which is connected to the ECU | Other ECUs in the same CAN bus network | Spread malware through CAN bus network | Access to the wireless connections | [60, 73] |
| Replay attack | Lights of the vehicle | Unexpected turn-off/on of the lights | The targeted vehicle with its passengers along with the other surroundings | Life-risking consequences | Retransmit previously eavesdropped packets | Ability to access the CAN bus network | [73] |
| Spying attack | TCU | Eavesdropping the in-car communications | All other ECUs connected to TCU | Compromised privacy of the passengers | Injecting malware to the TCU exploiting wireless connections | Vulnerabilities in the cellular network of the car | [15] |
| Relay attack | RKE system | Ability to open the car without using the key fob | Entire vehicle | Theft of the entire vehicle | Relaying the captured LF beacon signals sent from vehicle to key fob and resulting UHF signal sent from key fob to vehicle | Attacker must be equipped with tools such as antennas and amplifiers to relay | [45] |

would redirect the readers to a malicious website hosted by those attackers. This malware allowed the attackers to access confidential information stored in those systems [147].

- **Accidental Attack:** Software updates in the systems are necessary to maintain less vulnerabilities. However, if a software is updated with rebooting the system and the system has not completed the backup, all the critical data stored in that system will be erased. If that system is one of the control center systems, other components will also suffer from this and operate abnormally which may even cause plant shutdown [11].
- **Web-based Attacks:** In 2011, a number of oil and energy companies were attacked by Night Dragon which extracted private information from these plants. The attack exploited many vulnerabilities and combined different types of web-based attacks like SQL and malware injection [1, 101].

#### 8.5.1.2 Cyber-Physical Attacks

- **Communication Channels:** As mentioned above, dial-up connections connect the field devices directly which give the attackers direct access to the devices through the dial-up connection. Once in 2005, billing documentation of a water utility pump was modified by exploiting this dial-up connection in the canal system [150]. Although no physical damage was made, if intended, the attacker could have done critical physical damage also.
- **Resentful Insiders:** The workers of a company are familiar with the architecture and networking of the system. Once an ex-employee disrupted the functions intentionally of a sewage treat system of Maroochy Water Services located in Australia in 2000. The attacker used the knowledge as an ex-employee and exploited the vulnerabilities. As a result, the company faced a huge financial loss and the environment got tainted as many streets were flooded [141].
- **Modbus Worm:** Fovino et al. [39] did a tremendous work targeting the malware which is very alarming. A malware was crafted by them which could exploit the vulnerabilities like lack of authentication and integrity present in the Modbus protocol. This worm performs two kinds of attacks: (1) Identify the actuators and sensors, and then DoS including message, and (2) send unauthorized commands to the actuators and the sensors.
- **Malware:** Software vulnerabilities are exploited as a vector to target the physical devices. Stuxnet will be a better example for this. It targets the physical devices through software vulnerabilities [164]. Stuxnet's attack is categorized into two phases: (1) identify the object to target, and (2) hijack the PLC [77]. The first step is achieved with the help of two vulnerable computers running on Windows OS, i.e., shared printing server and Windows Server Service. Remote code execution using RPC would be possible through both vulnerabilities. The first one helps Stuxnet to install itself, whereas the next one allows it to spread to another computer. This process could affect millions of computers. However, because of specific targeted PLCs, the computers which are connected to that specific PLC

will be influenced. Once the Stuxnet is installed it looks for the software which monitors and controls the PLCs, and that would be Siemens WinCC. To find the accurate Siemens WinCC for the targeted PLC, a thorough analysis is done by the Stuxnet [77]. Once the software is determined, the next step is injecting malware to disrupt the operations of the PLCs. We would refer [111] for detailed analysis on Stuxnet.

- **Web-Based Attacks:** A web-based interface is exploited by attackers as a vector to attack the PLCs. They open up multiple connections and leave them open until they cannot be accessed by the authorized users which results in DoS attacks. Sometimes they send a link to the authorized users which contains malicious Java script which injects bug into the TCP/IP protocols and the controller gets affected as a result [150].

#### 8.5.1.3 Physical Attacks

- **Unintended Attack:** Zotob Worm is not intended to attack ICS, although it caused several manufacturers to shut down their plants. Once, the US company, Daimler Chrysler, was forced to shut down their 13 plants as a side effect of the attack by this worm [149]. This influenced a lot of researchers to analyze the detailed consequences of this attack. Among those, Fovino et al. [39] showed how critical could be the collateral damages of this attack. The consequences include rebooting of ICS servers, creation of vulnerabilities of arbitrary code injection, stimulating DoS attacks, and infecting personal computers.

### *8.5.2 Attacks on Smart grids*

#### 8.5.2.1 Cyber-Attacks

- **DoS Attacks:** In a smart grid, time is a critical variable. If there is too much delay in the flow of instructions, the components will operate undesirably. If different layers of the network of a smart grid are flooded, it will result in a DoS attack. Lu et al. [90] have analyzed the effect of a DoS attack in a smart grid. In addition, the deployment of wireless communication in the physical layer of smart grid has significantly increased which opens up the vulnerabilities for jamming attacks as shown in [91].
- **False Data Injection:** Injecting false data in the smart grid components leads to disrupted operations by the components. Liu et al. [88] have demonstrated a simulation in which they injected a set of false data and analyzed the consequences. They assumed that the attack had the pre-intrusion of the attackers to the control center and injected false data into the system and as a result, various components operate improperly. Moreover, the operating utilities will also face significant economic loss [156].

- **Untargeted Malware:** Sometimes, other components also get affected because of attacks to the targeted components. In 2003, the traffic between the substations and the field devices got disrupted by the Slammer worm and consequently the energy sector had the impact [8].
- **Customer's information:** Analyzing the interaction between a smart meter located at a house and the control center, the attackers can extract classified information of the customer. Such as the attackers can predict the lifestyle of the customer, when the customer is present at home, when they sleep, which house appliances are preferred, and many more [109].

#### 8.5.2.2 Cyber-Physical Attacks

- **Cyber Extortion:** Taking control over a smart meter, the attacker can extort money from the customer in return of not doing any large-scale damage to his households [112].
- **Blackouts:** If a smart grid is targeted for a DoS attack, mostly the consequence will be a blackout. Idaho National Laboratory experimented on a generator in 2007. The purpose of the experiment was to see what will be the impact on the generator due to a cyber-attack, and the results were infeasible [26]. In 2003, Ohio and Florida had experienced a wide-scale blackout which is believed to be the consequence of attacks done by the People's Liberation Army [55]. The USA had experienced over 800 blackouts in 2014 but the reasons are still unknown [122]. Although it is suggested by some speculation that such blackouts were the results of some cyber-physical attacks on the smart grids [55].

#### 8.5.2.3 Physical Attacks

- **Natural Incidents:** The physical exposure of the field devices is the reason behind physical attacks. Natural calamities are unpredictable and uncontrolled which can cause widespread damage to smart grid's field devices. For example, an ice storm in Philadelphia had caused a broad blackout for several days which affected over 500,000 people [122]. Growing trees, wild animals, and severe storms can easily damage the field devices and since the power transmission is widely spread, the consequences will be affected on a vast population. In 2014, wild animals caused around 150 blackouts in the USA by damaging the power cables [122].
- **Theft:** The field devices are made of metal and copper wires which have a good value in the market. Thieves steal those equipment which cause disconnection and as a result people suffer from blackouts. As an example, over 3000 people in Virginia had suffered from a blackout due to theft of equipment [122].
- **Car Accidents:** There were 356 outages due to car accidents which damaged transmission towers, power poles, or transformers [122].

- **Vandalism:** Attackers can intentionally damage the field devices of smart meters. Once a sniper shot over 100 rounds at a substation in California damaging around 17 transformers [41].
- **Terrorist Attacks:** A group of terrorists can attack a transmission control of smart grid to cut off communication due to lack of power. As an example, in 2014, terrorists blew a large section of transmission control in Yemen using a rocket launcher which resulted in a nationwide blackout affecting over 24 million people [64].

### *8.5.3 Attacks on Medical Devices*

#### 8.5.3.1 Cyber-Attacks

The attacks on the medical devices are mostly performed in an experimental environment. We have reviewed the attacks on some limited medical devices such as insulin pump, IMDs. The successful attacks made on the insulin pump could be a possible case for the other medical devices due to the similar hardware components and communication channels.

- **Privacy Invasion:** For a successful attack on a medical device, the attacker should be equipped with the device type, its PIN, and the authorized commands to disrupt the device. The authors of [87] have implemented a successful attack and showed that three factors must be known by the attacker: existence of the device, it's type, and PIN. Halperin et al. [52] have also revealed the classified information of the patient along with the device unique number by experimenting an attack on an ICD medical device.
- **Replay Attacks:** If the PIN of a medical device is intercepted by an attacker, it can be exploited in the future to replay the eavesdropped packet [87]. As a result, the insulin pump will operate based on misinformed decisions [125].

#### 8.5.3.2 Cyber-Physical Attacks

- **Replay Attacks:** An ICD was turned off when it was supposed to be working accurately, by Halperin et al. [52]. They replayed the "turn off" command which was used earlier as the commands to turn the device off. Any replay attack can retransmit the previously given commands to the CGM and insulin pump if the software vulnerabilities for replay are exploited by the attacker [87, 125].
- **False Data Injection (FDI):** Li et al. [87] experimented to inject false data into an insulin pump and they could control the pump remotely such as shutting down and resuming the pump.

#### 8.5.3.3 Physical Attacks

It is not a challenging task to physically access medical devices. The type and unique serial number of the devices can be obtained by a third party which is an example of physical attacks to medical devices [125].

### *8.5.4 Attacks on Smart Cars*

#### 8.5.4.1 Cyber-Attacks

Most of the researchers have done theoretical or simulation-based experiments on smart cars. Only a few have experimented on the actual real cars [15, 60, 73, 102]. To attack a car's internal network an attacker must go through the OBD port II, media player, or wirelessly connected devices, like smartphones. Once the attacker can access the internal network of the car, a lot of opportunities to launch an attack successfully are open.

- **DoS Attack:** A demonstration of DoS attack was done by Koscher et al. [73]. They disabled the interaction of CAN with the Body Control Module (BCM), resulting in the speedometer to drop from 40 to 0 mph instantly and also freeze the whole Instrument Panel Cluster. The freezing of IPC is like, if the driver increases the speed of the vehicle it will not be shown in the speedometer. As a result, the driver will be unaware of the increased speed and the chances of a severe accident will rise critically.
- **False Data Injection (FDI):** The BCM constantly sends the package to the speedometer containing updated speed of the vehicle which keeps the driver aware of the accurate velocity of the car. An attacker might intercept the data transmission between the BCM and the speedometer, which would modify those and forward the incorrect speed [73]. Another possibility is the correct status of the airbags installed in the car can be modified and, even if they are not in the correct state, they may appear healthy due to data modification [60]. In [51] the authors have shown how a customer can manipulate the data received by the insurance dongle to estimate the rate so that it will show lower insurance price.
- **Privacy Invasion:** The in-car conversation can be eavesdropped if the cellular interface in the TCU is exploited by an attacker as demonstrated by Checkoway et al. [15]. They can extract many other private information regarding the vehicle and its owner. Also as Ed Markey, a US senator, had reported, the manufacturing companies store many private information such as driving history and the performance of the car [95].

#### 8.5.4.2 Cyber-Physical Attacks

- **DoS Attack:** As examples of DoS attacks, passengers are not able to close any open windows, the theft alarm system of the car doesn't work when it is needed [60], etc. Jamming RKE signals is an example of DoS attacks on the wireless communication.
- **Malware Injection through Bluetooth:** Exploiting the wirelessly connected devices, an attacker can control the other ECUs which are connected through the same network to which the ECU of the Bluetooth connectivity is connected. A successful attack via Bluetooth was conducted by Checkoway et al. [15] in which they compromised a connected smartphone device. The connectivity of the device with the smart car's TCU, which is the Bluetooth ECU, was exploited as a malware named Trojan horse that was injected in the smartphone. After the connection is compromised, the malware sends a malicious payload to the TCU and in result the attacker gets the ability to control other connected ECUs such as Antilock Braking System (ABS). Another attack was shown by Woo et al. [161] that how a mobile device app can be exploited to attack a car's OBD-II port.
- **Malware Injection through OBD-II port:** The OBD-II port is the gateway for the attacker to control the internal networks of various ECUs. For example, Hoppe et al. [60] demonstrated that if an attacker equipped with the OBD-II port of a car injects malicious command, it will develop into DoS attack. The consequences are preventing passengers from opening or closing windows, showing the incorrect status of the airbags, false information regarding air pressure of the tires, and many more.
- **Replay Attacks:** This attack requires the attacker to intercept the CAN network traffic when certain functions are being done by specific ECUs so that those can be replayed to reactivate those functions. Koscher et al. [73] could disable the interior and the exterior lights of a vehicle by delivering previously eavesdropped packets.
- **Packet Injection:** The first step for this attack is to get access to the CAN network. Once an attacker gets access to the CAN network, an enormous number of attacks become feasible. For example, many functions of the car's engine can be disrupted by exploiting the OBD-II port such as increasing the Revolution Per Minute (RPM), disabling the engine's cylinder or even the whole engine. Electronic Brake Control Module controls releasing and locking of the brake. If random modified packets are sent to EBCM, the driving will become unsafe [73]. By injecting arbitrary packets, Lee et al. [83] was able to disrupt many functions of multiple vehicles by performing fuzzing attack. This attack is about capturing the CAN ID and flooding the network by sending arbitrary packets having the same ID.

#### 8.5.4.3 Physical Attacks

- **Relay Attacks:** Many cars have keyless entry nowadays which is like unlocking/locking the car from the key fob. In this attack, the RKE is being targeted

where the vehicle communicates with the key fob. At first the periodical LF Beacon signal, sent by the vehicle to its key fob, is exploited by the attacker to know if the key fob is in close range or not. The attacker relays the communication to send an "Open" Ultrahigh-Frequency signal to unlock the car and in this way the attacker can even start the engine. Eight different manufacturers have implemented this attack successfully on ten different vehicles [45]. This is also an example of Man-in-The-Middle (MITM) attack [158]. Garcia et al. [47] have also implemented these successful attacks by exploiting simple cryptographic measures in the physical layer communication to access the vehicle by coning the car key.

- **ABS Spoofing:** Shoukrey et al. [139] have demonstrated such successful attacks where the target is ABS wheel speed sensor. They have installed a malicious actuator which produces a different magnetic field disrupting the original magnetic field generated by the ABS wheel speed sensor and sends incorrect information to the ABS ECU.

## 8.6 Security Control and Solutions

In this section, we describe different solutions in CPS controls. The different solutions can be classified into three different types. Some are application-specific solutions, some are general solutions regardless of application, and some solutions are cross-domain.

### *8.6.1 General CPS Controls*

In this, we review general solutions to secure CPS regardless of the application. The first step of each solution is addressing the causes of vulnerability.

- **Superfluous Connectivity:** Security measures should be taken to prevent unauthorized access to the access point. The protocols used for these communications are well-known proprietary protocols (Modbus, DNP3) or open protocols such as TCP/IP. The proprietary protocols are full of vulnerabilities because of isolation from public testing [3].
- **Communication:** Improved security solutions at communication level in ICS. Intrusion Detection System (IDS) should be designed in such a way that long-delays become intolerable. Mitchell and Chen [103, 106, 107] focus on designing improved IDS which are time-critical. **Device Verification:** The software running on CPS should be authentic. One such verification process is Trusted Platform Module (TPM). TPM is hardware-based solutions providing physical security, which is infeasible to provide in some ICS and smart grids. Therefore, there is the need for revised TPMs considering limited CPS resources.

### *8.6.2 Application-Specific Controls*

#### 8.6.2.1 ICS Controls

- **Modern design:** ICS needs security solutions which are specific to a system. For such solutions, numerous factors should be taken into consideration, such as cyber-physical interactions, and heterogeneity of components and protocols. Most ICS aims at providing reliability to the system during non-malicious failures as suggested by Cardenas et al. [12], but in the current scenarios, cyber-attacks are more common than before. Therefore, security should be taken into consideration besides reliability when designing innovative solutions.
- **Add-on security for Protocols:** Various modifications have been proposed to modify current protocols such as Modbus, DNP, and ICCP to improve the prior security measurements of the traditional IT solutions. For providing non-repudiation, authentication, and preventing replayed attacks, Secure Modbus framework was proposed by Fovino et al. [40]. To add integrity, authenticity, and confidentiality to the security measures, DNPSec has been suggested by Majdalawieh et al. [94].
- **IDS:** It is less complex to design an IDS for ICS compared to traditional security framework for IT. The ability to predict the traffic and the static networking is the reason behind this [75]. A set of goals to be monitored by IDS in ICS is presented by Zhu et al. [167] which are (1) ability to detect any access to the communication links of sensor/actuators and controllers, (2) detection of any kind of customization in the settings of sensors, and (3) physical tampering of actuators. Also, WildCat, a solution to control the physical exposure of ICS plants' wireless network, is presented by D'Amico et al. [27]. This WildCat should be installed in the security guard's car to detect any suspicious wireless activities going on in the perimeter of a CPS plant. The collected information is sent to the control center where it will be analyzed and will send the location of the activity source to the guards. For further details about current IDS solutions for a ICS plant, we will recommend [6, 16, 75, 76, 107, 126, 167].
- **Remote Access:** The field devices must be controlled by only authorized personnel remotely, as suggested by Fernandez et al. [35]. To secure the field device, a designated laptop should constantly be operating through VPN to detect any unauthorized activity. To avoid web-based DoS attacks, Turk et al. [150] suggested that any idle connection should be closed which also helps to reduce the complexity of multiconnection.
- **Encryption and Key Management:** Encryption is a fundamental requirement for ICS systems and the delay in it could be very crucial in a time-sensitive environment. To solve this, a new key management was designed by Choi et al. [23] for specifically ICS which does not cause any delay. ICS environments are widely spread and to protect them, Cao et al. [10] have come up with a new layered security protocol which is based on Hash Chains. This layered security protocol (1) divides the ICS into two different zones based on high-security and

low-security levels, and (2) manages a lightweight key mechanism. This way, even though an attacker can break through the security of low-level security-based ICS components, accessing the high-level security-based components would not be possible.

- **Software Control:** To continuously update the modified security measurements into the system is a very rigorous and complex process. To prevent the Stuxnet attack, Windows have released a security patch which focuses on Stuxnet-related attacks [77]. Similarly, the manufacturers of ICS components must keep up with modification of security measures of the system and then manufacture the compatible devices. This way it will be ensured that no old vulnerabilities are there in the system [67].
- **Standardization:** The leading bodies like the National Institute of Standards and Technology should focus on the security measures of ICS significantly as both the technical and operational controls are critical. Neglecting either one can be very critical for the entire system. Stouffer et al. [144] have provided a solution regarding this issue. They proposed guidelines for technical problems, like IDS, firewalls; and operational control such as awareness, security, and training of the employers. The operational control is as important as technical problems. For example, as per the report by ICS-CERT, most of the attacks made to ICS are done by phishing [1]. If the employers, without awareness, open those malicious emails, the entire system could become vulnerable [114]. According to the comparison done by Sommestad et al. [142], standardization bodies normally focus on either technical or operational problem, but it is highly required to be focused on both.

#### 8.6.2.2 Smart Grid Controls

- **DoS Controls:** Attacks like DoS at the network layer are prevented by filtering malicious packets, rate-limiting, and reconfiguring network architecture. Unlike the first two, due to the static nature, the third one will not be easy for smart grids. Also, security techniques in smart grids usually focus on the wireless jamming kind of attacks. The techniques which prevent DoS attacks are divided into four categories: packet-based, signal-based, hybrid, and proactive detection [156].
- **IDS:** Due to the enormous size of smart grids and the heterogeneous components, designing IDS for smart grids is very difficult [143]. The design of IDS for the smart grids must be different from the ones built for traditional IT systems to reduce the possibility of false data injection. An IDS is proposed by Jin et al. [65] which is anomaly-based and uses artificial ants and invariant detection, with Bayesian reasoning approach, to detect any malicious activity. In addition, another IDS was suggested by Mitchell and Chen [105] that works on behavior rule which protects the cyber-physical devices of smart grids, such as subscriber energy meters, data aggregation points, etc. Another significant contribution made by Liu et al. [89], who presented an IDS, which prevent the ICS from false data injection.

- **Authorization and authentication:** Employees usually have access to certain field devices with authorization and authentication. The problem is that the smart grid is spread vastly and most of the field devices share the same authorization credentials. This makes any malicious employee capable of tampering with any of those field devices, and the identity of the attacker could not be tracked as other employees also have the authorized access. Hence, Vaidya et al. [154] came up with a mechanism which strengthens the authentication and authorization of field devices. This mechanism provides legitimate employees the ability to access the field devices remotely from the automation systems in the smart grids, and it relies on elliptic curve cryptography.
- **Modern designs:** Each of the aspects of a CPS needs to be approached differently due to constantly evolving security issues. Mo et al. [108] have proposed the area "cyber-physical security" for the first time. This approach considers the details of both cyber and physical aspects of security. They have demonstrated their approach on two different kinds of attacks: stealthy deception and replay attack. The importance of each of the two aspects, cyber and physical, has been emphasized in the above-mentioned literature. However, the approaches are enhancing the existing protocols which are a temporary solution. Hence a whole bottom-up redesign of the system is desired.
- **Add-on Security:** To prevent the advanced attacks, the trend is about merging the required additional security with the existing one. For example, secure DNP3, which has basic security measurements, such as authentication, confidentiality services, and encryption, is an advanced version of simple DNP3. In the simple version, the add-on security is added by placing another layer of security in the communication level of these protocols [156].
- **Privacy-preserving Controls:** When the data flow from the smart meter to the utility company, due to lack of confidentiality, the usage and patterns of the consumer can be intercepted as well as the data can be modified due to lack of integrity in the security protocols which may result in disrupted billing information [97, 143]. To solve these problems, a certain number of techniques have emerged to provide better privacy when the data is in transit between smart meters and the utility companies [34, 156]. Attacking the smart meters attached to a household, the occupancy of the house can be predicted to break in successfully. As a solution to this problem, Chen et al. [19] have proposed a mechanism named combined heat and privacy (CHP) which makes the usage data look like the house is always occupied by tricking the occupancy detection techniques.
- **Standardization:** To secure the communications among the smart grids, certain standardizations have been introduced by several bodies like NIST and IEC. For example, such guidelines for smart grids are developed by NIST in the report 7268 [116]. In addition, standards like TC57, 6235 are developed by IEC [25].
- **Preventing disabling of smart grids:** To prevent the exploitation of disabling feature of a smart grid, Anderson and Fuloria [4] have suggested that the manufacturers should program smart meters in this way that they could let the customer know, in enough time, in advance before the malicious command takes

effect and the meters get disabled. This may help in the detection of DoS attack also.

- **Physical Security:** To prevent the physical tampering with smart meters, NIST standard states that all the meters should have cryptographic access to the meter and must be sealed inside tamper-resistant units [116].

#### 8.6.2.3 Medical Devices Control

- **Authentication:** Authentication: To prevent unauthorized access to the IMDs, Halperin et al. [52] have proposed a cryptographic-based mechanism along with a key exchange procedure which improves the authentication. Both mechanisms rely on external radio frequency, as a source of energy, instead of batteries. In addition, another protocol named Out of Band (OOB) was deployed to improve the authentication measurements. This authentication protocol uses a different channel than the one which is used for communication [131]. For advanced key generation in encrypted communication, heart rate, glucose level, electrocardiograms can also be used in Body Sensor Network (BSN) [131, 136]. The movement of a patient can also be used for key generation [117].
- **Intrusion Detection System:** A mechanism that alarms the patient whenever it detects any unauthorized attempts to interact with IMDs was proposed by Halperin et al. [52]. Similarly, Gollakota et al. [48] also proposed a mechanism named Shield which detects and prevents any unauthorized attempt to connect IMDs wirelessly. In addition, there was an attempt by Mitchell and Chen [104] on how to prevent any posed threats by disrupted actuators and sensors. The mechanism proposed by them can detect the affected actuators and sensors by behavior rule-based Intrusion Detection System but it is not designed for the IMDs or wearable devices. Their technique is applicable for stand-alone devices which work solely such as cardiac device and vital sign monitor.
- **Location-Based Control:** Some security technique relies on protocols based on distance bounding. This protocol prevents an attacker to attack remotely. Various techniques such as received signal strength, ultrasound signals, electrocardiography, etc. determine the limit of distance [166]. However, since these techniques do not provide any authorization, other mechanisms are needed to be incorporated [131].
- **Thwarting Active and Passive Attacks:** Body Coupled Communication thwarts most of the active and passive attacks made on the insulin pump as stated by Li et al. [87]. They have experimented with this communication and showed that since this communication of an insulin pump uses the human body as their medium to communicate, instead of any wireless communication, it thwarts the possible attacks by exploiting the wireless channel. To tamper the insulin pump, the attacker needs to reach very close to the patient which is a bar for most of the attackers.
- **Shifting Security to Wearable Devices:** The IMDs and wearable devices have their risk management security measurements, and this can be challenging for

shifting the security measurements to another device. For example, to replace a patient's IMD with a more secure device, the process can be life-risking at first. Even if it is not life-risking, regarding battery and computational resources, the shifting of security techniques could be still expensive. Hence, the optimal solution is to deploy another device which will solely operate on the security issues. Xu et al. [163] have designed a device called IMD Gaurd which defends the medical devices against spoofing and jamming attacks. Similarly, as previously discussed, Gollakota et al. [48] proposed the shield device which will defend the wearable device from any unauthorized attempt to interact.

- **Cross-Domain Solutions:** Here is a certain similar limitation in both smart cars and medical devices such as constraints of data and power. Hence to defend from eavesdropping and replay attacks, the rolling code encryption of smart cars is implemented in medical devices as suggested by Li et al. [87].
- **Standardization and recommendations:** The leading body in the standardization of medical devices is the Food and Drug Administration (FDA). There are several standards and guidelines issue by the FDA for the manufacturers of the medical devices. They have suggested in 2005 that the usage of COTS creates the maximum number of vulnerabilities since it has the capability to be accessed remotely [37]. Another standard regarding cyber security was posted by them in 2014 [38]. However, they lack the intensity that mandates following of these guidelines which allow the manufacturers to follow their preferred guidelines. To resolve this issue, the latest BAN standard, IEEE 802.15.6, has been implemented to stop the manufacturers from the production of less secure medical devices [63].
- **Allowing vs. disallowing remote functionalities:** To prevent the attackers from intercepting the channel between the patients' medical devices and the remote physicians, the manufacturers should limit this remote access of the medical devices. So, it is suggested the medical device should not receive remote commands from the physician but should only send the patients' log and health status to the physician. Although this will prevent the medical devices from unauthorized commands, it will also limit the complete usability of the device [82]. Hence, Hayajneh et al. [57] came up with a cryptographic system named Rabin public key which prevents the medical system from unauthorized commands even if they are passed to the medical devices.

#### 8.6.2.4 Smart Cars Controls

- **Unimplemented promising controls:** There are several promising controls which have not been implemented yet. For example, Wolf et al. [160] have proposed authentication gateway, firewalls, and encryption to secure the bus network. Another security paradigm named defense-in depth, i.e., detection, prevention, countermeasures, deflections, and recovery, was suggested by Larson et al. [78] as the replacement design of the security measure in the cars.
- **Cryptography:** Although, cryptography adds advanced authorization, integrity, and authentication, the computational cost of these will be high due to the

limited functionalities of cars' components. Thus Wolf et al. [159] presented Hardware Security Module which is cost-efficient as it is hardware based. This mechanism secures the communication channels of the ECUs in a car along with V2V communication channel. In addition, a standard for equipping the car with better security measurements, Escherich et al. [33] designed Secure Hardware Extension which adds secure boot and secret key protection to the cars' ECUs.

- **Redefining Trust:** To disable the arbitrary ECUs from performing operations to diagnose and reflash, a trust-related control was presented by Koscher et al. [73]. They also implemented another trust-related control which requires authentication and authorization for the ECUs which are allocated for the reflashing and diagnostic operations. For the successful implementation of these, trusted platforms along with remote verification are required [70].
- **Restricted Critical Commands:** Physical access to the cars can be a gateway for attackers. There are certain commands which require physical access to cars for implementation, and this could be critical as any benign malicious command could be a serious attack. Koscher et al. [73] emphasized on this part that physical access to the car is always dangerous. If the number of commands requiring physical access is restricted the convenience and flexibility of the car will be affected. So, such security mechanism should be implemented which will balance both sides.
- **Bluetooth:** The TCU controls Bluetooth connectivity of the cars which is also connected to the other ECUs. Hence, attackers can exploit connected electronic devices, which are connected to the cars via Bluetooth, and attack other ECUs of the car through TCU as well [15]. For the need for extra security layer to secure Bluetooth connectivity, Dardanelli et al. [28] proposed a mechanism which is applicable for two-wheeler vehicles but should be efficient for cars also. To process thorough authentication of the smartphones before connecting to cars via Bluetooth, a mechanism was also suggested by Woo et al. [161]. This helps to reduce the number of attacks exploiting vulnerabilities in Bluetooth connected smartphones.
- **IDS:** Most of the Intrusion Detection Systems are designed for CAN network protocols, whereas a very few are designed for other protocols such as LIN and FlexRay. A specification-based IDS is designed by Larson et al. [79] which is installed in every single ECU. Another behavior-based IDS is designed for both FlexRay and CAN network by Stefan and Roman [137]. As cost is an important factor for the implementation of security mechanism, a very cost-efficient mechanism of anomaly-based IDS is designed by Miller and Valasek[155]. Another anomaly-based IDS which uses time as a constraint is very effective for detecting intrusion or any anomaly. It is designed by Cho and Shins [22], and this mechanism utilizes the measurements of the time intervals of periodic messages to uniquely identify each of the ECUs. Taylor et al. [148] have proposed an IDS mechanism which compares the frequency of currently sent packets with the historically sent packet which had strict frequencies. This way it can detect anomalies in the frequency of delivered packets.

## 8.7 Security Challenges

### *8.7.1 Challenges in General CPS*

- **Security by Design:** Most of the CPS model is not secured enough in their design model because they are not considered enough due to their isolation from other systems in a physically secured environment, such as no Internet connection which makes the physical security measure the most important one [144].
- **Cyber-Physical Security:** To provide optimal cyber-physical security, the cyber and physical aspects have to be considered separately with the same importance. This way the cyber-attacks with physical consequences will be better predicted and prevented [49]. The solutions of the attacks on CPS will be focused on cyber only unless the fundamental differences between the physical and cyber aspects are properly contemplated as suggested by Neuman et al. [113]. A new field named "Cyber-Physical Security" was proposed by Mo et al. [108]. They have also described some novel solutions regarding cyber-physical security, especially for smart grids. The systems' ability to survive under an attack carries the same importance as the security challenges. A set of security solutions including the systems' survivability are discussed in [12].
- **The Real-Timeliness Nature:** The absence of real-time requirement affects the security model [14, 113] since the real-time decision is very crucial for the attacked CPS system. Hence, contemplation of the interactions between cyber and physical aspects gives the full picture of the CPS model with which arises the importance of risk assessment [14].
- **Uncoordinated Change:** A CPS usually have many stakeholders among which most of them are somehow related to the system such as manufacturers, operators, and implementers. Hence, while implementing any changes in the system, the coordination among them is necessary. Otherwise, due to lack of coordination in security measures, the heterogeneous components of a CPS will become vulnerable [3, 92].

In addition, with the above mentioned general challenges with the CPS security, we have briefed about the challenges with each of the four applications.

### *8.7.2 Challenge in ICS*

- **Change Management:** The components of ICS are diversely spanned geographically which are needed to be updated, repaired, removed, or replaced at some point. For example, a perfect planning is required to update any component of ICS, or else other components can experience unexpected failure. Once, in a nuclear plant, an unexpected failure occurred due to an update in the computer system [12]. Moreover, the large number of stakeholder can also cause unexpected failure unintentionally. Hence the coordinated change management is very crucial to managing the security-related changes in the system [92, 144].

- **Insider Threat:** The insiders of a system have the detailed knowledge about the components. They can attack the system intentionally or unintentionally exploiting the trust given to them. The Maroochy incident is an example of this. They can also help the remote attackers by giving them the access or confidential information. This kind of threat needs more serious considerations [75].
- **Secure Integration:** ICS is inherently vulnerable by the vulnerabilities of its legacy systems. Thus, the integration of the components with their legacy vulnerabilities must be done very securely so that they do not create any new vulnerabilities. However, since there are many components in an ICS it is practically infeasible to replace all of them at one time due to financial concerns [75], but, meanwhile, short-term security updates must be implemented to reduce the potential risks [12].

### *8.7.3 Challenges in Smart Grids*

- **Two-Way Communication:** The advanced metering system allows the smart meters attached to the households to communicate to the utility companies directly which increases the physical attack. Unlike power grid, smart meters are physically accessible which is a threat to the utility companies also [69].
- **Access Control Mechanism:** As smart grid is widely spread, the mechanism for access control to the field devices must be considered strictly [3]. There must be proper control and mechanisms at every possible access point.
- **Privacy Concerns:** The communication traffic contains consumers' privileged information also. Besides the encryption of data, there must be an anonymization technique induced to prevent several types of attacks from deducing patterns of the encrypted information [72, 109]. A homomorphic encryption was proposed by Li et al. [86]. This mechanism protects the privacy of the consumers while the low overhead of the smart grids is also maintained. However, this encryption is not enough to prevent an attacker from injecting false data or impersonating a legitimate smart meter [156].
- **Explicit Trust:** There must be proper mechanisms and security measures to detect false data. Since the size of smart grid is large, it is very difficult to detect false data injection and unauthorized commands by the security measure which are designed to detect faults only [29, 88].
- **Comprehensive Security:** The security levels at the lower levels of smart grids, like field devices, are less compared to high-level components, such as control centers, due to the less capabilities of the smart grid. Because of the additional maintenance cost, the security solutions should be lightweight and cost-effective [71].
- **Change Management:** The management of changes in a smart grid is as challenging as ICS or even more as smart grids are more diverse and the number of its stakeholders are also more. Since the capabilities of change management are very limited in smart grids, it is intensely required to make the grids more secure [143].

### 8.7.4 Challenges in Medical Devices

- **Usability vs. Security:** Too much security can be a serious problem for medical devices. For example, if a patient with critical health condition needs urgent care by IMD, but the IMD needs assistance from another person, who does not have the access privileges or the cryptographic credentials, the unavailability could be dangerous for the patient [53, 129]. So, it is required to focus on the usability along with security. The ideal solution would be allowing usability during emergencies while providing security as much as possible. Denning et al. [30] have proposed an optimal solution which uses fail-open/safety wristband. The patient wears a wristband which prevents any unauthorized person to interact with the IMDs, and when the patient removes the band, the IMD can be accessed by unauthorized persons.
- **Increased code for add-on security:** The limited power of the medical devices could be affected if addition functions are required due to additional codes of security. This may kill the original purpose of the device [131]. Hence, it is advisable that the medical devices should be focusing on their priority operation where the security measurements will be managed by some external device [48].
- **Limited Resources:** The power resources are critical for these small devices which require limited energy. The additional mechanism for security requires extra energy [53, 129]. The power of these devices must be maintained for several years. As one of the first efforts, Halperin et al. [52] proposed a security which has battery-free power consumption. It solely relies on RF as the energy source.

To drain the battery resources of these medical devices, attackers can flood the network with unnecessary commands which result in DoS attack [131]. Although if a medical device refuses to interact with unauthorized components, sending and receiving unnecessary commands will consume a certain amount of energy. Hence, new security controls must be developed to prevent the devices to respond to any illegitimate activity.

### 8.7.5 Challenges in Smart Cars

- **Secure Integration:** In a smart car, the COTS and the third-party components are integrated by the manufacturer. Sometimes the integrated components mismatch due to lack of detailed information about the COTS and the third-party components. According to Sagstetter et al. [133], it is advisable that we should use the formal techniques, like a model-based design, where in every step the correctness of the information about the COTS and the third-party components are verified. Also, the measurements of access control need to be improved to prevent any unauthorized access to the components which are usually originated from the mismatch between the COTS and the third-party components [115].

- **Effective Separation:** The traffic of CAN network is separated into high and low frequencies by the gateway ECUs; however, many attacks were still able to bypass these gateways and attack various ECUs [73]. As a solution, some manufacturers deployed some techniques where the critical ECUs are in a totally separated network [155]. Another solution is to replace the simple ECUs with Master-ECUs [133], though it is very costly.
- **Heterogeneity of Components:** It is very normal that different components of a car cannot be manufactured by only car manufacturing companies. So, the distinct types of components are normally manufactured by Original Equipment Manufacturers (OEM). So, the accord among different parties must be maintained as the security measurements and capabilities are different for each component. Hence, security engineers who are familiar with early design phase must be incorporated by the manufacturers [70].
- **In-Car Communication:** Due to the assumption of isolation, CAN network is vulnerable inherently. Hence, there is a need for new protocols which assumes that there are potential attackers who can exploit these vulnerabilities. Therefore, a highly secure platform was proposed by the OVERSEE project [50] which will revolutionize the CAN network by replacing it. In addition, firewall and IDS kind of temporary solutions can be used as a part of the gateway ECU or even a separated ECU. Currently, there is a potential project going on state-full IDS which helps to detect the legitimacy of each packet sent to the vehicle during its different states as driving, parking, and much more [146].
- **New Vulnerabilities:** In the near future, there will be more challenges regarding the new security issues, V2V, and V2I communications as discussed in [119].

## 8.8 Conclusion

This chapter builds a good reference for those who need to get an overview of the recent studies and real-life issues of the CPS' security measures. We have discussed various possible threats to CPS according to different motives of the attacker. This helps to emphasize the awareness of the popularly used security measures of the CPS applications and can nurture great danger upon us. This chapter contains what are the common vulnerabilities in CPSs that can be exploited by the attackers and will allow the readers to learn about areas of the CPS that need more emphasis. For a better understanding of the exploitation of such vulnerabilities, we have presented several attacks that have taken place in the real world and succeeded. A taxonomy of these real-world attacks is also included for a quick study about successful attacks exploiting existing vulnerabilities. Some of the previous and current studies which are done on the security issues of CPS are covered in this chapter which will assist the readers having a desire to pursue further research in this field.

## References


1. Alcaraz C, Zeadally S (2013) Critical control system protection in the 21st century. Computer 46(10):74-83
2. Amin S, Litrico X, Sastry SS, Bayen AM (2010) Stealthy deception attacks on water scada systems. In: Proceedings of the 13th ACM international conference on hybrid systems: computation and control. ACM, New York, pp 161–170
3. Amin S, Schwartz GA, Hussain A (2013) In quest of benchmarking security risks to cyber-physical systems. IEEE Netw 27(1):19–24
4. Anderson R, Fuloria S (2010) Who controls the off switch? In: 2010 first IEEE international conference on smart grid communications (SmartGridComm). IEEE, Piscataway, pp 96–101
5. Bellovin, SM (1989) Security problems in the TCP/IP protocol suite. ACM SIGCOMM Comput Commun Rev 19(2):32–48
6. Briesemeister L, Cheung S, Lindqvist U, Valdes A (2010) Detection, correlation, and visualization of attacks against critical infrastructure systems. In: 2010 eighth annual international conference on privacy security and trust (PST). IEEE, Piscataway, pp 15–22
7. Brooks RR, Sander ST, Deng J, Taiber J (2008) Automotive system security: challenges and state-of-the-art. In: Proceedings of the 4th annual workshop on cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead (CSIIRW '08). ACM, New York, pp 26:1–26:3
8. Byres E, Lowe J (2004) The myths and facts behind cyber security risks for industrial control systems. In: Proceedings of the VDE Kongress, vol 116
9. Byres E, Franz M, Miller D (2004) The use of attack trees in assessing vulnerabilities in scada systems. In: Proceedings of the international infrastructure survivability workshop
10. Cao H, Zhu P, Lu X, Gurtov A (2013) A layered encryption mechanism for networked critical infrastructures. IEEE Netw 27(1):12–18
11. Càrdenas AA, Amin S, Sastry S (2008) Research challenges for the security of control systems. In: Proceedings of the 3rd conference on hot topics in security (HOTSEC)
12. Càrdenas A, Amin S, Sinopoli B, Giani A, Perrig A, Sastry S (2009) Challenges for securing CPSs. In: Workshop on future directions in cyber-physical systems security
13. Cardenas AA, Roosta T, Sastry S (2009) Rethinking security properties, threat models, and the design space in sensor networks: a case study in SCADA systems. Ad Hoc Netw 7(8): 1434–1447
14. Càrdenas AA, Amin S, Lin ZS, Huang YL, Huang CY, Sastry S (2011) Attacks against process control systems: risk assessment, detection, and response. In: Proceedings of the 6th ACM symposium on information, computer and communications security. ACM, New York, pp 355–366
15. Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H, Savage S, Koscher K, Czeskis A, Roesner F, Kohno T (2011) Comprehensive experimental analyses of automotive attack surfaces. In: USENIX security symposium
16. Cheminod M, Durante L, Valenzano A (2013) Review of security issues in industrial networks. IEEE Trans Ind Inf 9(1):277–293
17. Chen TM, Abu-Nimeh S (2011) Lessons from Stuxnet. Computer 44(4):91–93
18. Chen M, Gonzalez S, Vasilakos A, Cao H, Leung VC (2011) Body area networks: a survey. Mobile Netw Appl 16(2):171–193
19. Chen D, Kalra S, Irwin D, Shenoy P, Albrecht J (2015) Preventing occupancy detection from smart meters. IEEE Trans Smart Grid 6(5):2426–2434
20. Chien E, OMurchu L, Falliere N (2012) W32.duqu: the precursor to the next Stuxnet. In: Presented as part of the 5th USENIX workshop on large-scale exploits and emergent threats. USENIX, Berkeley
21. Cho S (2014) Privacy and authentication in smart grid networks. Doctoral dissertation, Ph. D. dissertation, Department of Computer Science, State University of New York, Incheon, South Korea

22. Cho KT, Shin KG (2016) Error handling of in-vehicle networks makes them vulnerable. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, New York, pp 1044–1055
23. Choi D, Kim H, Won D, Kim S (2009) Advanced key-management architecture for secure SCADA communications. IEEE Trans Power Delivery 24(3):1154–1163
24. Chow R, Uzun E, Càrdenas AA, Song Z, Lee S (2011) Enhancing cyber physical security through data patterns. In: Workshop on foundations of dependable and secure cyber-physical systems (FDSCPS), p 25
25. Cleveland FM (2012) IEC 62351 security standards for the power system information infrastructure. http://iectc57.ucaiug.org/wg15public/Public%20Documents/White%20Paper%20on%20Security%20Standards%20in%20IEC%20TC57.pdf
26. CNN (2007) Sources: staged cyber-attack reveals vulnerability in power grid. http://www.cnn.com/2007/US/09/26/power.at.risk/
27. D'Amico A, Verderosa C, Horn C, Imhof T (2011) Integrating physical and cyber security resources to detect wireless threats to critical infrastructure. In: 2011 IEEE international conference on technologies for homeland security (HST), pp 494–500
28. Dardanelli A, Maggi F, Tanelli M, Zanero S, Savaresi SM, Kochanek R, Holz T (2013) A security layer for smartphone-to-vehicle communication over bluetooth. IEEE Embed Syst Lett 5(3):34–7
29. Das SK, Kant K, Zhang N (2012) Handbook on securing cyber-physical critical infrastructure. Elsevier, Amsterdam
30. Denning T, Kramer DB, Friedman B, Reynolds MR, Gill B, Kohno T (2014) CPS: beyond usability: applying value sensitive design based methods to investigate domain characteristics for security for implantable cardiac devices. In: Proceedings of the 30th annual computer security applications conference. ACM, New York, pp 426–435
31. East S, Butts J, Papa M, Shenoi S (2009) A taxonomy of attacks on the DNP3 protocol. In: Palmer C, Shenoi S (eds), Critical infrastructure protection III. IFIP advances in information and communication technology, vol 311. Springer, Berlin, pp 67–81
32. Ericsson, GN (2010) Cyber security and power system communication 2014; essential parts of a smart grid infrastructure. IEEE Trans Power Delivery 25(3):1501–1507
33. Escherich R, Ledendecker I, Schmal C, Kuhls B, Grothe C, Scharberth F (2009) SHE: secure hardware extension-functional specification, version 1.1. Hersteller Initiative Software (HIS) AK Security
34. Fang X, Misra X, Xue G, Yang D (2012) Smart gridthe new and improved power grid: a survey. IEEE Commun Surv Tutorials 14(4):944–980
35. Fernandez JD, Fernandez AE (2005) SCADA systems: vulnerabilities and remediation. J Comput Sci Coll 20(4):160–168
36. Fleury T, Khurana H, Welch V (2008) Towards a taxonomy of attacks against energy control systems. In: Papa M, Shenoi S (eds) Critical infrastructure protection II. The international federation for information processing, vol 290. Springer, New York, pp 71–85
37. Food and Drug Administration (FDA) (2005) Cybersecurity for networked medical devices containing off-the-shelf (OTS) software. http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm077823.pdf
38. Food and Drug Administration (FDA) (2014) Cybersecurity for networked medical devices containing off- the-shelf (OTS) software. www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM356190.pdf
39. Fovino IN, Carcano A, Masera M, Trombetta A (2009) An experimental investigation of malware attacks on SCADA systems. Int J Crit Infrastruct Prot 2(4):139–145
40. Fovino IN, Carcano A, Masera M, Trombetta A (2009) Design and implementation of a secure Modbus protocol. In: Critical infrastructure protection III. Springer, Berlin, pp 83–96
41. FOX News Network (2014) Threat to the grid? Details emerge of sniper attack on power station. http://www.foxnews.com/politics/2014/02/06/2013-sniper-attack-on-power-grid-still-concern-in-washington-andfor-utilities/

42. Francia G III, Thornton D, Brookshire T (2012) Wireless vulnerability of SCADA systems. In: Smith RK, Vrbsky SV (eds), ACM southeast regional conference. ACM, New York, pp 331–332
43. Francia III G, Thornton D, Brookshire T (2012) Wireless vulnerability of SCADA systems. In: Proceedings of the 50th annual southeast regional conference, 2012 Mar 29. ACM, New York, pp 331–332
44. Francia III GA, Thornton D, Dawson J (2012) Security best practices and risk assessment of SCADA and industrial control systems. In: Proceedings of the international conference on security and management (SAM), 2012 Jan 1. The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Las Vegas, p 1
45. Francillon A, Danev B, Capkun S (2011) Relay attacks on passive keyless entry and start systems in modern cars. In: Proceedings of the network and distributed system security symposium (NDSS)
46. Fu K, Blum J (2013) Controlling for cybersecurity risks of medical device software. Commun ACM 56(10):35–37
47. Garcia FD, Oswald D, Kasper T, Pavlid'es P (2016) Lock it and still lose it–on the (in) security of automotive remote keyless entry systems. In: 25th USENIX security symposium (USENIX Security 16)
48. Gollakota S, Hassanieh H, Ransford B, Katabi D, Fu K (2011) They can hear your heartbeats: non-invasive security for implantable medical devices. SIGCOMM Comput Commun Rev 41(4):2–13
49. Gollmann D (2013) Security for cyber-physical systems. In: Mathematical and engineering methods in computer science. Springer, Berlin, pp 12–14
50. Groll A, Holle J, Ruland C, Wolf M, Wollinger T, Zweers F (2009) Oversee a secure and open communication and runtime platform for innovative automotive applications. In: 7th embedded security in cars conference (ESCAR)
51. Guan L, Xu J, Wang S, Xing X, Lin L, Huang H, Liu P, Lee W (2016) From physical to cyber: escalating protection for personalized auto insurance. In: Proceedings of the 14th ACM conference on embedded network sensor systems CD-ROM (SenSys '16). ACM, New York, pp 42–55
52. Halperin D, Clark SS, Fu K, Heydt-Benjamin TS, Defend B, Kohno T, Ransford B, Morgan W, Maisel WH (2008) Pacemakers and implantable cardiac defibrillators: software radio attacks and zeropower defenses. In: IEEE symposium on security and privacy (SP 2008), pp 129–142
53. Halperin D, Heydt-Benjamin TS, Fu K, Kohno T, Maisel WH (2008) Security and privacy for implantable medical devices. IEEE Pervasive Comput 7(1):30–39
54. Hanna S, Rolles R, Molina-Markham A, Poosankam P, Fu K, Song D (2011) Take two software updates and see me in the morning: the case for software security evaluations of medical devices. In: Proceedings of the 2nd USENIX conference on health security and privacy, health (SEC '11). USENIX Association, Berkeley, p 6
55. Harris S (2008) China's cyber militia. National Journal Magazine, 31 May
56. Harris B, Hunt R (1999) TCP/IP security threats and attack methods. Comput Commun 22(10):885–897
57. Hayajneh T, Mohd BJ, Imran M, Almashaqbeh G, Vasilakos AV (2016) Secure authentication for remote patient monitoring with wireless medical sensor networks. Sensors 16(4):424
58. Hieb JL (2008) Security hardened remote terminal units for SCADA networks. University of Louisville, Louisville
59. Hoglund G, McGraw G (2004) Exploiting software: how to break code. Pearson Education India, New Delhi
60. Hoppe T, Kiltz S, Dittmann J (2011) Security threats to automotive can networks — practical examples and selected short-term countermeasures. In: Proceedings of the 27th international conference on computer safety, reliability, and security (SAFECOMP '08). Springer, Berlin, pp 235–248

61. Huitsing P, Chandia R, Papa M, Shenoi S (2008) Attack taxonomies for the modbus protocols. Int J Crit Infrastruct Prot 1:37–44
62. ICS-CERT (2010) Common cybersecurity vulnerabilities in industrial control systems. http://ics-cert.us-cert.gov/sites/default/files/recommendedpractices/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf
63. IEEE 802.16 Working Group et al (2004) IEEE standard for local and metropolitan area networks. part 16: air interface for fixed broadband wireless access systems. IEEE Std 802: 16–2004
64. Investment Watch (2014) First time in history, a terrorist attack on the electric power grid has blacked-out an entire nation in this case Yemen. http://investmentwatchblog.com/first-time-in-history-a-terrorist-attack-onthe-electric-power-grid-has-blacked-out-an-entire-nation-in-thiscase-yemen
65. Jin X, Bigham J, Rodaway J, Gamez D, Phillips C (2006) Anomaly detection in electricity cyber infrastructures. In: Proceedings of the international workshop on complex networks and infrastructure protection (CNIP '06)
66. Jo, HJ, Choi W, Na SY, Woo S, Lee DH (2016) Vulnerabilities of android OS-based telematics system. Wirel Pers Commun 92(4):1511–1530
67. Johnson RE (2010) Survey of SCADA security challenges and potential attack vectors. In: 2010 international conference for internet technology and secured transactions (ICITST). IEEE, Piscataway, pp 1–5
68. Karygiannis T, Owens L (2002) Wireless network security. NIST special publication, 800:48
69. Khurana H, Hadley M, Lu N, Frincke DA (2010) Smart-grid security issues. IEEE Secur Priv 8(1):81–85
70. Kleidermacher D, Kleidermacher M (2012) Embedded systems security: practical methods for safe and secure software and systems development. Elsevier, Amsterdam
71. Knapp ED, Samani R (2013) Applied cyber security and the smart grid: implementing security controls into the modern power infrastructure, Newnes
72. Komninos N, Philippou E, Pitsillides A (2014) Survey in smart grid and smart home security: issues, challenges and countermeasures. IEEE Commun Surv Tutorials 16(4):1933–1954
73. Koscher K, Czeskis A, Roesner F, Patel S, Kohno T, Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H, Savage S (2010) Experimental security analysis of a modern automobile. In: 2010 IEEE symposium on security and privacy (SP), pp 447–462
74. Krishnamurti T et al (2012) Preparing for smart grid technologies: a behavioral decision research approach to understanding consumer expectations about smart meters. Energy Policy 41:790–797
75. Krotofil M, Gollmann D (2013) Industrial control systems security: what is happening? In: 2013 11th IEEE international conference on industrial informatics (INDIN). IEEE, Piscataway, pp 670–675
76. Krotofil M, Larsen J, Gollmann D (2015) The process matters: ensuring data veracity in cyber-physical systems. In: Proceedings of the 10th ACM symposium on information, computer and communications security (ASIA CCS '15). ACM, New York, pp 133–144
77. Langner R (2011) Stuxnet: dissecting a cyberwarfare weapon. Secur Priv IEEE 9(3):49–51
78. Larson UE, Nilsson DK (2008) Securing vehicles against cyber-attacks. In: Proceedings of the 4th annual workshop on cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead (CSIIRW '08). ACM, New York, pp 30:1–30:3
79. Larson UE, Nilsson DK, Jonsson E (2008) An approach to specification-based attack detection for in-vehicle networks. In: 2008 IEEE intelligent vehicles symposium. IEEE, Piscataway, pp 220–225
80. Lee EA (2008) CPSs: design challenges. In: 2008 11th IEEE international symposium on object oriented real-time distributed computing (ISORC). IEEE, Piscataway
81. Lee EA, Seshia SA (2011) Introduction to embedded systems: a cyber-physical systems approach. University of California, Berkeley

82. Lee I, Sokolsky O, Chen S, Hatcliff J, Jee E, Kim B, King A, Mullen-Fortino M, Park S, Roederer A et al (2012) Challenges and research directions in medical cyber–physical systems. Proc IEEE 100(1):75–90
83. Lee H, Choi K, Chung K, Kim J, Yim K (2015) Fuzzing can packets into automobiles. In: 2015 IEEE 29th international conference on advanced information networking and applications. IEEE, Piscataway, pp 817–821
84. Leverett EP (2011) Quantitatively assessing and visualising industrial system attack surfaces. University of Cambridge, Darwin College
85. Leverett E, Wightman R (2013) Vulnerability inheritance in programmable logic controllers. https://ics-cert.us-cert.gov/content/cyber-threat-source-descriptions
86. Li F, Luo B, Liu P (2010) Secure information aggregation for smart grids using homomorphic encryption. In: 2010 first IEEE international conference on smart grid communications (SmartGridComm). IEEE, Piscataway, pp 327–332
87. Li C, Raghunathan A, Jha NK (2011) Hijacking an insulin pump: security attacks and defenses for a diabetes therapy system. In: 2011 13th IEEE international conference on e-health networking applications and services (Healthcom), pp 150–156
88. Liu Y, Ning P, Reiter MK (2011) False data injection attacks against state estimation in electric power grids. ACM Trans Inf Syst Secur 14(1):13
89. Liu T, Sun Y, Liu Y, Gui Y, Zhao Y, Wang D, Shen C (2015) Abnormal traffic-indexed state estimation: a cyber-physical fusion approach for smart grid attack detection. Futur Gener Comput Syst 49:94–103
90. Lu Z, Lu X, Wang W, Wang C (2010) Review and evaluation of security threats on the communication networks in the smart grid. In: Military communications conference (MILCOM' 10). IEEE, Piscataway, pp 1830–1835
91. Lu Z, Wang W, Wang C (2011) From jammer to gambler: modeling and detection of jamming attacks against time-critical traffic. In: Proceedings IEEE INFOCOM. IEEE, Piscataway, pp 1871–1879
92. Luallen, ME (2011) Critical control system vulnerabilities demonstrated - and what to do about them. A SANS Whitepaper
93. MacDonald D, Clements SL, Patrick SW, Perkins C, Muller G, Lancaster MJ, Hutton W (2013) Cyber/physical security vulnerability assessment integration. In: 2013 IEEE PES innovative smart grid technologies (ISGT). IEEE, Piscataway, pp 1–6
94. Majdalawieh M, Parisi-Presicce F, Wijesekera D (2006) DNPSEC: distributed network protocol version 3 (DNP3) security ramework. In: Advances in computer, information, and systems sciences, and engineering. Springer, Berlin, pp 227–234
95. Markey E (2015) Tracking and hacking: security and privacy gaps put American drivers at risk. http://www.markey.senate.gov/imo/media/doc/2015-02-06MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
96. Mashima D, Càrdenas AA (2012) Evaluating electricity theft detectors in smart grid networks. In: Proceedings of the 15th international conference on research in attacks, intrusions, and defenses (RAID'12). Springer, Berlin, pp 210–229
97. McDaniel P, McLaughlin S (2009) Security and privacy challenges in the smart grid. Secur Priv IEEE 7(3):75–77
98. Metke AR, Ekl RL (2010) Security technology for smart grid networks. IEEE Trans Smart Grid 1(1):99–107
99. Microsoft Security Tech Center (2008) Microsoft security bulletin ms08-067 - critical. https://technet.microsoft.com/library/security/ms08-067
100. Microsoft Security TechCenter (2010) Microsoft security bulletin summary for September 2010. https://technet.microsoft.com/library/security/ms10-sep
101. Miller B, Rowe D (2012) A survey scada of and critical infrastructure incidents. In: Proceedings of the 1st annual conference on research in information technology. ACM, New York, pp 51–56
102. Miller C, Valasek C (2013) Adventures in automotive networks and control units. A SANS whitepaper

103. Mitchell R, Chen IR (2011) Survivability analysis of mobile CPSs with voting-based intrusion detection. In: 2011 7th international wireless communications and mobile computing conference (IWCMC), pp 2256–2261
104. Mitchell R, Chen R (2012) Behavior rule based intrusion detection for supporting secure medical CPSs. In: 2012 21st international conference on computer communications and networks (ICCCN). IEEE, Piscataway, pp 1–7
105. Mitchell R, Chen R (2013) Behavior-rule based intrusion detection systems for safety critical smart grid applications. IEEE Trans Smart Grid 4(3):1254–1263
106. Mitchell R, Chen IR (2013) Effect of intrusion detection and response on reliability of CPSs. IEEE Trans Reliab 62(1):199–210
107. Mitchell R, Chen IR (2014) A survey of intrusion detection techniques for cyber-physical systems. ACM Comput Surv 46(4):55
108. Mo Y, Kim THJ, Brancik K, Dickinson D, Lee H, Perrig A, Sinopoli B (2012) Cyber-physical security of a smart grid infrastructure. Proc IEEE 100(1):195–209
109. Molina-Markham A, Shenoy P, Fu K, Cecchet E, Irwin D (2010) Private memoirs of a smart meter. In: Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building. ACM, New York, pp 61–66
110. Munro K (2012) Deconstructing flame: the limitations of traditional defences. Comput Fraud Secur 2012(10):8–11
111. Falliere N, Murchu LO, Chien E (2011) W32. stuxnet dossier. White Paper (Symantec Corporation, Security Response) 5(6):29
112. Nakashima E, Mufson S (2008) Hackers have attacked foreign utilities, CIA analyst says. Washington Post, 19 January
113. Neuman C (2009) Challenges in security for cyber-physical systems. In: DHS workshop on future directions in cyber-physical systems security, Citeseer
114. Nicholson A, Webber S, Dyer S, Patel T, Janicke H (2012) SCADA security in the light of cyber-warfare. Comput Secur 31(4):418–436
115. Nilsson DK, Phung PH, Larson UE (2008) Vehicle ECU classification based on safety-security characteristics. In: Road transport information and control-RTIC 2008 and ITS United Kingdom members' conference. IET, Stevenage, pp 1–7
116. NIST (2010) NISTIR. 7628: guidelines for smart grid cyber security. Technical report
117. Oberoi D, Sou WY, Lui YY, Fisher R, Dinca L, Hancke GP (2016) Wearable security: key derivation for body area sensor networks based on host movement. In: 2016 IEEE 25th international symposium on industrial electronics (ISIE). IEEE, Piscataway, pp 1116–1121
118. Paukatong T (2005) SCADA security: a new concerning issue of an inhouse egat-scada. In: Transmission and distribution conference and exhibition: Asia and pacific, IEEE/PES, pp 1–5
119. Petit J, Shladover SE (2015) Potential cyberattacks on automated vehicles. IEEE Trans Intell Transp Syst 16(2):546–556
120. Pfleeger CP, Pfleeger SL (2006) Security in computing, 4th edn. Prentice Hall PTR, Upper Saddle River
121. Pièrre-Cambacèdès L, Tritschler M, Ericsson GN (2011) Cybersecurity myths on power control systems: 21 misconceptions and false beliefs. IEEE Trans Power Delivery 26(1): 161–172
122. Bauman K, Tuzhilin A, Zaczynski R (2017) Using social sensors for detecting emergency events: a case of power outages in the electrical utility industry. ACM Trans Manag Inf Syst (TMIS) 8(2–3):7
123. Quinn, EL (2009) Privacy and the new energy infrastructure. SSRN 1370731
124. Rad AHM, Leon-Garcia A (2011) Distributed internet-based load altering attacks against smart power grids. IEEE Trans Smart Grid 2(4):667–674
125. Radcliffe J (2011) Hacking medical devices for fun and insulin: breaking the human scada system. In: Black Hat Conference presentation slides, vol 2011
126. Barbosa RR (2014) Anomaly detection in SCADA systems: a network based approach
127. Rahman MA, Bera P, Al-Shaer E (2012) Smartanalyzer: a noninvasive security threat analyzer for AMI smart grid. In: Proceedings IEEE INFOCOM, pp 2255–2263

128. Rajkumar RR et al (2010) Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th design automation conference. ACM, New York
129. Rostami M, Burleson W, Koushanfar F, Juels A (2013) Balancing security and utility in medical devices? In: Proceedings of the 50th annual design automation conference. ACM, New York, p 13
130. Roufa RMI, Mustafaa H, Taylora SOT, Xua W, Gruteserb M, Trappeb W, Seskarb I (2010) Security and privacy vulnerabilities of in-car wireless networks: a tire pressure monitoring system case study. In: 19th USENIX security symposium, Washington, pp 11–13
131. Rushanan M, Rubin AD, Kune DF, Swanson CM (2014) SoK: security and privacy in implantable medical devices and body area networks. In: IEEE symposium on security and privacy
132. Ryu DH, Kim HJ, Um K (2009) Reducing security vulnerabilities for critical infrastructure. J Loss Prev Process Ind 22(6):1020–1024. Papers presented at the 2007 and 2008 international symposium of the Mary Kay O'Connor process safety center and papers presented at the fWCOGIg 2007
133. Sagstetter F, Lukasiewycz M, Steinhorst S, Wolf M, Bouard A, Harris WR, Jha S, Peyrin T, Poschmann A, Chakraborty S (2013) Security challenges in automotive hardware/software architecture design. In: Proceedings of the conference on design, automation and test in Europe, EDA Consortium, San Jose, pp 458–463
134. Salmon D et al (2009) Mitigating the aurora vulnerability with existing technology. In: Proceedings of the 36th annual western protective relay conference
135. Santamarta R (2012) Here be backdoors: a journey into the secrets of industrial firmware. https://media.blackhat.com/bh-us-12/Briefings/Santamarta/BH_US_12_Santamarta_Backdoors_WP.pdf
136. Seepers RM, Weber JH, Erkin Z, Sourdis I, Strydis C (2016) Secure key-exchange protocol for implants using heartbeats. In: Proceedings of the ACM international conference on computing frontiers. ACM, New York, pp 119–126
137. Seifert S, Obermaisser R (2014) Secure automotive gateway– secure communication for future cars. In: 12th IEEE international conference on industrial informatics (INDIN). IEEE, Piscataway, pp 213–220
138. Miciolino EE, Bernieri G, Pascucci F, Setola R (2015) Communications network analysis in a SCADA system testbed under cyber-attacks. In: Telecommunications Forum Telfor (TELFOR), 2015 23rd-2015 Nov 24. IEEE, New York, pp 341–344
139. Shoukry Y, Martin P, Tabuada P, Srivastava M (2013) Non-invasive spoofing attacks for anti-lock braking systems. In: Cryptographic hardware and embedded systems-CHES 2013. Springer, Berlin, pp 55–72
140. Shoukry Y, Martin P, Yona Y, Diggavi S, Srivastava M (2015) PYCRA: physical challenge-response authentication for active sensors under spoofing attacks. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, New York, pp 1004–1015
141. Slay J, Miller M (2007) Lessons learned from the maroochy water breach. In: Critical infrastructure protection, pp 73–82
142. Sommestad T, Ericsson GN, Nordlander J (2010) SCADA system cyber security—a comparison of standards. In: 2010 IEEE power and energy society general meeting. IEEE, Piscataway, pp 1–8
143. Sridhar S, Hahn A, Govindarasu M (2012) Cyber– physical system security for the electric power grid. Proc IEEE 100(1):210–224
144. Stouffer KA, Falco JA, Scarfone KA (2011) SP 800-82. Guide to industrial control systems (ICS) security: supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other control system configurations such as programmable logic controllers (PLC). Technical report, Gaithersburg
145. Studnia I, Nicomette V, Alata E, Deswarte Y, Kaˆaniche M, Laarouchi Y (2013) Survey on security threats and protection mechanisms in embedded automotive networks. In: 43rd annual IEEE/IFIP conference on dependable systems and networks workshop (DSN-W). IEEE, Piscataway, pp 1–12

146. Studnia I, Nicomette V, Alata E, Deswarte Y, Kaaniche M, Laarouchi Y (2013) Security of embedded automotive networks: state of the art and a research proposal. In: SAFECOMP 2013-workshop CARS (2nd workshop on critical automotive applications: robustness & safety) of the 32nd international conference on computer safety, reliability and security
147. Symantec Security Response (2014) Dragonfly: western energy companies under sabotage threat. http://www.symantec.com/connect/blogs/dragonfly-western-energy-companies-under-sabotage-threat
148. Taylor A, Japkowicz N, Leblanc S (2015) Frequency based anomaly detection for the automotive can bus. In: 2015 world congress on industrial control systems security (WCICSS). IEEE, Piscataway, pp 45–49
149. Tsang R (2010) Cyberthreats, vulnerabilities and attacks on scada networks. University of California, Berkeley
150. Turk RJ (2005) Cyber incidents involving control systems. Idaho National Laboratory (INL), Idaho Falls
151. Urias V, Leeuwen BV, Richardson B (2012) Supervisory command and data acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed. In: Military communications conference, 2012 - MILCOM, pp 1–8
152. US-CERT (2009) Cyber threat source descriptions. https://ics-cert.us-cert.gov/content/cyber-threat-source-descriptions
153. Vaas L (2013) Doctors disabled wireless in Dick Cheney's pacemaker to thwart hacking. https://nakedsecurity.sophos.com/2013/10/22/doctors-disabled-wireless-in-dick-cheneys-pacemaker-to-thwart-hacking/
154. Vaidya B, Makrakis D, Mouftah HT (2013) Authentication and authorization mechanisms for substation automation in smart grid network. IEEE Netw 27(1):5–11
155. Valasek C, Miller C (2014) A survey of remote automotive attack surfaces. Black Hat USA 2014
156. Wang W, Lu Z (2013) Cyber security in the smart grid: survey and challenges. Comput Netw 57(5):1344–1371
157. Welch D, Lathrop S (2003) Wireless security threat taxonomy. In: IEEE systems, man and cybernetics society information assurance workshop. IEEE, Piscataway, pp 76–83
158. Wetzels J (2014) Broken keys to the kingdom: security and privacy aspects of rfid-based car keys. Preprint arXiv:1405.7424
159. Wolf M, Gendrullis T (2012) Design, implementation, and evaluation of a vehicular hardware security module. In: Information security and cryptology-ICISC 2011. Springer, Berlin, pp 302–318
160. Wolf M, Weimerskirch A, Paar C (2004) Security in automotive bus systems. In Proceedings of the workshop on embedded security in cars (ESCAR'04)
161. Woo S, Jo HJ, Lee DH (2015) A practical wireless attack on the connected car and security protocol for in-vehicle can. IEEE Trans Intell Transp Syst 16(2):993–1006
162. Xie L, Mo Y, Sinopoli B (2010) False data injection attacks in electricity markets. In: 2010 first IEEE international conference on smart grid communications (SmartGridComm). IEEE, Piscataway, pp 226–231
163. Xu F, Qin Z, Tan CC, Wang B, Li Q (2011) Imdguard: securing implantable medical devices with the external wearable guardian. In: Proceedings IEEE INFOCOM, pp 1862–1870
164. Yampolskiy M, Horvath P, Koutsoukos XD, Xue Y, Sztipanovits J (2013) Taxonomy for description of cross-domain attacks on CPS. In: Proceedings of the 2nd ACM international conference on high confidence networked systems. ACM, New York, pp 135–142
165. Zeller M (2011) Myth or reality? Does the aurora vulnerability pose a risk to my generator? In: 2011 64th annual conference for protective relay engineers. IEEE, Piscataway, pp 130–136
166. Zheng G, Fang G, Shankaran R, Orgun MA (2015) Encryption for implantable medical devices using modified one-time pads. IEEE Access 3:825–836

167. Zhu B, Sastry S (2010) Scada-specific intrusion detection/prevention systems: a survey and taxonomy. In: Proceedings of the 1st workshop on secure control systems (SCS)
168. Zhu B, Joseph A, Sastry S (2011) A taxonomy of cyber-attacks on scada systems. In: Proceedings of the 2011 international conference on internet of things and 4th international conference on cyber, physical and social computing (ITHINGSCPSCOM '11). IEEE Computer Society, Washington, pp 380–388

# Index





S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6