A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics

Dimitrios Pliatsios, *Member, IEEE*, Panagiotis Sarigiannidis, *Member, IEEE*,
Thomas Lagkas, *Senior Member, IEEE*, and Antonios G. Sarigiannidis, *Member, IEEE*

***Abstract*—Supervisory Control and Data Acquisition (SCADA) systems are the underlying monitoring and control components of critical infrastructures, such as power, telecommunication, transportation, pipelines, chemicals and manufacturing plants. Legacy SCADA systems operated on isolated networks, that made them less exposed to Internet threats. However, the increasing connection of SCADA systems to the Internet, as well as corporate networks, introduces severe security issues. Security considerations for SCADA systems are gaining higher attention, as the number of security incidents against these critical infrastructures is increasing. In this survey, we provide an overview of the general SCADA architecture, along with a detailed description of the SCADA communication protocols. Additionally, we discuss certain high-impact security incidents, objectives, and threats. Furthermore, we carry out an extensive review of the security proposals and tactics that aim to secure SCADA systems. We also discuss the state of SCADA system security. Finally, we present the current research trends and future advancements of SCADA security.**

***Index Terms*—SCADA, cybersecurity, protocols, security, smart grid, trends.**

I. Introduction

The facilities, systems, processes, networks, and services, that are crucial to the security, safety and economic well-being of the people and organizations, are considered as critical infrastructures. Such infrastructures include power, telecommunications, transportation, pipelines, chemicals and manufacturing plants.

The Industrial Control Systems (ICSs) include both Supervisory Control and Data Acquisition (SCADA) systems and Distributed Control Systems (DCSs)/Process Control Systems (PCSs), as well as other control system configurations [1]. In particular, SCADA systems are the underlying monitoring components of many critical infrastructures, while DCSs/PCSs interconnect the distributed sensors and actuators, and manage the control process. The ICSs enable real-time monitoring and control of the process, by providing access to remote and local operators through the Human-Machine Interfaces (HMIs).

A typical SCADA system is composed of a central controller, and a number of distributed field devices, such as sensors and actuators. The data exchange between the controller and the field devices is enabled by certain communication protocols, that have been specifically developed for industrial applications.

Legacy SCADA systems operated on isolated networks that made them less exposed to Internet threats. In addition, the limited availability of technical details regarding the utilized protocols increased the security of the systems. Nowadays, these networks are being interconnected to common networks such as the Internet, in order to leverage the robustness of common network protocols, facilitate remote accessibility and reduce the capital and operating expenditure. However, the interconnection of SCADA systems with the Internet introduces severe security issues.

A cyber attack against a SCADA system can have devastating consequences. The continuous and reliable operation of SCADA systems can have a crucial effect on public safety and health. As a consequence, any security incidents on these systems may threaten public health and safety. For example, an attacker can compromise a SCADA system and shut down electricity, gas, and water services, or destroy critical military infrastructure.

A. Related Work and Contribution

There are several existing works that discuss and review the current state of SCADA security. Igure *et al.* [2] provide an overview of the security state in SCADA networks. The authors discussed the security threats and vulnerabilities in common SCADA networks. They also presented the research challenges and discussed the ongoing work in several SCADA security areas. The authors in [3] review research proposals in the area of SCADA security and provide an overview of the vulnerabilities, risks and countermeasures. An overview of smart grid communication technologies is provided in [4]. In addition, the authors summarize the substantial security requirements of the smart grid communication infrastructure. The authors in [5] present a comprehensive survey of security issues for the smart grid. Additionally, they present the security requirements, the vulnerabilities,

Manuscript received January 17, 2019; revised September 22, 2019, January 27, 2020, and March 20, 2020; accepted April 5, 2020. Date of publication April 14, 2020; date of current version August 21, 2020. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme (SPEAR) under Grant 787011. *(Corresponding author: Panagiotis Sarigiannidis.)*

Dimitrios Pliatsios and Panagiotis Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, 50100 Kozani, Greece (e-mail: psarigiannidis@uowm.gr).

Thomas Lagkas is with the Department of Computer Science, International Hellenic University, 65404 Kavala, Greece.

Antonios G. Sarigiannidis is with the Research and Development Department, Sidroco Holdings Ltd., Limassol 1077, Cyprus.

Digital Object Identifier 10.1109/COMST.2020.2987688

and the potential countermeasures. Furthermore, they discuss the state-of-art security solutions and provide future research directions. Leszczyna and Egozcue [6] provide an overview of ICS and carry out a survey to identify the main concerns of ICS security, as well as potential security solutions. Based on the survey findings, the authors provide recommendations for the protection of ICS [7]. The authors in [8] review SCADA standards and communication infrastructures, and discuss security issues and solutions. Furthermore, they list several SCADA security schemes. McLaughlin *et al.* [9] explore the ICS cyber security landscape. They review the vulnerability assessment process, the emerging SCADA threats and discuss ICS testbeds for vulnerability analysis. The authors in [10] provide a comprehensive survey of tools and techniques that assess the vulnerabilities and evaluated the security of SCADA systems. Finally, Giraldo *et al.* [11] present a taxonomy of security surveys regarding cyber-physical systems. The presented taxonomy, classifies the reviewed surveys based on the application domains, security and privacy attacks, and counter measures. A survey of security in SCADA networks is provided in [12]. The authors present the communication architecture, as well as a classification of potential threats and attacks. In addition, various novel security schemes are categorized into detection and prevention of SCADA attacks.

However, the aforementioned related works have certain shortcomings. The work in [2] lists the common SCADA protocols, without describing their specifications. The works in [4] and [5] only consider the communication infrastructure that is used in the smart grid, while [3], [8], and [12] only provide a description of the SCADA architecture, without presenting details about the communication protocols. Moreover, only a general description of security counter-measures is provided. The work in [9] review the SCADA security of each layer, without considering specific protocol threats and countermeasures. Finally, there are no previous surveys that provide an extensive description of common SCADA protocol specifications, and a thorough and up-to-date review of the SCADA security measures.

Motivated by the aforementioned remarks, we present this survey aiming to address these shortcomings, provide further specification and implementation details about a wide variety of SCADA protocols, and offer an up-to-date analysis on the state of SCADA security along with trends and advancements. Specifically, the following contributions are included in this survey:

- An overview of the general SCADA architecture and supported communication protocols. Firstly, we present the general SCADA architecture and its main components. Following, we list the well-known SCADA communication protocols, along with their specifications, supported topologies, data rates, and packet structure.
- A discussion of SCADA security incidents, objectives, and threats. We report certain incidents that had a significant impact in order to show the importance of security in the SCADA systems. Then, we present the security objectives of a SCADA system and describe common attack types against those systems. We also provide a description of SCADA testbeds, that have been developed to assist security researchers.
- A thorough review of the security proposals that aim to secure SCADA systems. We have categorized the reviewed proposals into four groups based on the utilized SCADA protocol. We also list the proposed approaches and methodologies, the challenges that the authors addressed and the evaluation results.
- A presentation of the research trends and advancements of the SCADA security. Those trends include novel SCADA protocols, that are being designed to support the requirements of the emerging Industry 4.0, the integration of the Internet of Things concept, the leverage of virtualization technologies such as Software Defined Networking and Network Function Virtualization, the leverage of Big Data analytics in securing SCADA systems, and finally, the adoption of a SCADA cyber hygiene framework.

The rest of the paper is organized as follows. Section II presents the general architecture of a SCADA network and briefly describes its main components. Moreover, it provides a detailed overview of commonly used SCADA communication protocols and their specifications. In Section III, we report certain well-known SCADA security incidents, we discuss the security issues of SCADA networks, describe the common attacks against SCADA systems, and review the SCADA security testbeds. Section IV provides a thorough review of security proposals based on SCADA protocols, such as Modbus, DNP3, and Profinet. Furthermore, a discussion about the state of SCADA security is provided. Section V discusses the survivability and resilience of SCADA systems in presence of cyber and physical threats. In Section VI we provide future trends and advancements in SCADA systems and we conclude this paper in Section VII.

II. SCADA Systems

A. Architecture

SCADA systems are extensively used in industrial applications to control and monitor the process systems. As shown in Fig. 1, a typical SCADA system consists of the following components:

The **Operator**, who is responsible for monitoring the system, addressing alerts and performing the necessary control operations. The operator can be located in the premises of the organization or he can access the system remotely through the Internet.

The **HMI**, which facilitates the interaction between the operator and the SCADA system. The HMI collects information from the Master Terminal Unit and translates the control commands appropriately.

The organization's **Intranet**, that consists of computational, networking, and storage components located within the organization. It facilitates the operation of the system by running analytics on the data collected from field devices.

The **Master Terminal Unit (MTU)**, which is responsible for gathering data from remote terminals, transmitting them

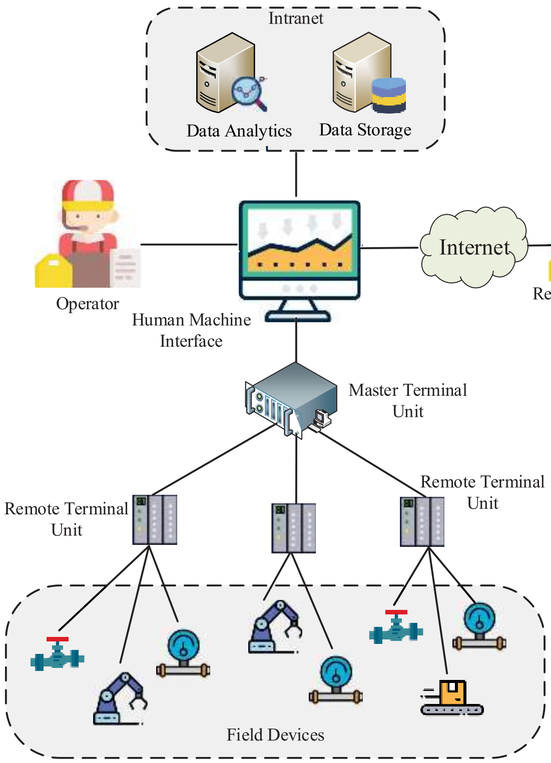


Fig. 1. SCADA System Architecture.

to the HMI, and sending control signals. It also provides the high-level control logic for the system.

The **Remote Terminal Units (RTU)**, which exchange data and commands with the MTU and the send specified control signals to the field devices.

The **Field Devices**, which are distributed across the organization and consist of devices that monitor and control the industrial process. For example, a number of sensors is used to gather data, while actuators perform the control actions.

In order to realize continuous, reliable and efficient communication between the aforementioned SCADA components, certain communication protocols have been devised. Those protocols take into account the processing capabilities of the components and the communication requirements of the industrial applications. The rest of the section provides an overview of the well-known protocols, along with technical specifications such as supported topologies, data rates, packet structure, and network layer technologies. Table I provides a summary of the protocol specifications that are described in this section. The Network Infrastructure column shows the underlying communication technology that each protocol uses, while each of the supported topologies are listed in the Topologies column. The Data Rates and Maximum Distance columns list the supported data rates and the maximum device distance from its controller, respectively.

B. Fieldbus-Based Protocols

Fieldbus is a network system for real-time distributed control in industrial applications. It enables the connection of field devices such as sensors, motors, and actuators, with their associated controllers. Fieldbuses differ according to the topology, the transmission medium, and the transmission protocols. They also differ in regard to the maximum cable length and the maximum data size per telegram.

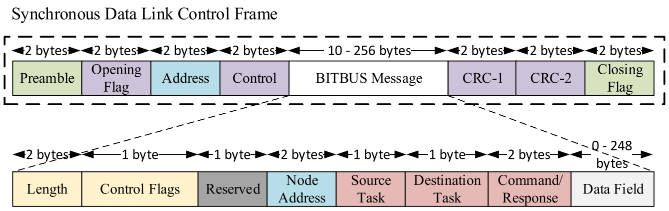


Fig. 2. BITBUS Frame Structure.

Fieldbus offers certain advantages compared to parallel wiring. As it uses a single cable running through all devices, the networks can be designed and deployed more quickly. The short path between the devices increases the availability and reliability of the network. The use of standardized protocols enables the connection of equipment of different manufacturers. Finally, the network can be easily modified and extended, in order to adapt to future requirements.

BITBUS is an open implementation of the Fieldbus protocol. It can extend up to 1200 meters, while the supported data rates are 62.5 Kbps, 365 Kbps, and 1.5 Mbps, depending on the distance. The interconnection is based on RS485 specification, using twisted pairs cable. BITBUS is based on the bus topology, where a maximum of 28 nodes can be connected in a bus segment. The number of nodes can be extended up to 250, by using repeaters and decreasing the data rate. Each device is assigned a unique address in the form of a number ranging from 1 to 249. The address 255 is reserved as the broadcast address.

Fig. 2 presents the structure of a BITBUS message, which is encapsulated in a Synchronous Data Link Control (SDLC) frame. The frame starts with 16-bit preamble along with a unique Opening flag (1 byte). The Address field (1-2 bytes) contains the recipient of the message, while the Control field (1-2 bytes) determines the type of the frame. The length field specifies the size of the message, while the MT, SE, DE, and TR fields are reserved for routing information. The node address ranges from 1-249 and specifies the destination node. The Source and Destination Tasks identify the task that has generated the command and the reply respectively. The Command/Response field contains the command that has to be executed. The Data Field has a variable length from 0 to 248 bytes. Finally, two CRC fields and a Closing flag are appended by the SDLC frame.

Foundation Fieldbus H1 [13] is a bi-directional communications protocol used for communications among field devices and the control system. Each communication point of the controller can connect with up to 32 nodes using either twisted pair or fiber. The data rate is fixed to 31.25 Kbps and the maximum distance between the master and a slave is 1900 meters. Using up to 4 repeaters the distance can be extended to 9500 meters. Each device is assigned an address in the form of a number ranging from 1 to 255. The protocol does not support broadcasting functionalities.

TABLE I
SCADA COMMUNICATION PROTOCOLS

Protocol	**Network Infrastructure**	**Topologies**	**Data Rates**	**Maximum Distance**
BITBUS	Fieldbus	Bus	62.5 Kbps, 375 Kbps, 1.5 Mbps	1200m
DC-BUS	2-wire cable	Line	115.2 Kbps up to 1.3 Mbps	100 km
Distributed Network Protocol 3	Ethernet	Line, Peer-to-Peer	100 Mbps, 1 Gbps	100m
EtherCAT	Ethernet	Ring, Line, Daisy-chain	100 Mbps	100m
Ethernet Powerlink	Ethernet	Tree, Line, Star, Peer-to-Peer	100 Mbps	100m
Foundation Fieldbus H1	Fieldbus	Point-to-point, Bus with Spur, Daisy-chain, Tree	31.25 Kbps	1900 m without repeater, 9500 m with up to 4 repeaters
Foundation HSE	Ethernet	Tree, Line, Star, Peer-to-Peer	100 Mbps	100m
HART	2-wire cable	Point-to-point, Multi-drop	1.2 Kbps	3 km
IEC 60870	Serial, Ethernet	Ring, Tree, Line, Star	N/A	N/A
IEC 61850	Ethernet	Ring, Tree, Line, Star	N/A	100m
Modbus	Serial, Ethernet	Line, Star, Ring, Mesh (MB+)	100 Mbps, 1 Gbps	N/A
PROFIBUS	Fieldbus	Point-to-point, Bus with spur, Daisy-chain, Tree	9.6 Kbps to 12 Mbps	100 to 1200m, 15km for optical channel
PROFINET	Ethernet	Ring, Tree, Line, Star	100 Mbps, 1 Gbps	100m
RAPIEnet	Ethernet	Line, Ring	100 Mbps	100m
SERCOS III	Ethernet	Line, Ring	100 Mbps, 1 Gbps	N/A
Unitronics PCOM	Serial, Ethernet	Ring, Line, Star	100 Mbps	100m
WorldFIP	Fieldbus	Bus	31.25 Kbps, 1 Mbps, 2.5 Mbps, 5 Mbps	1km

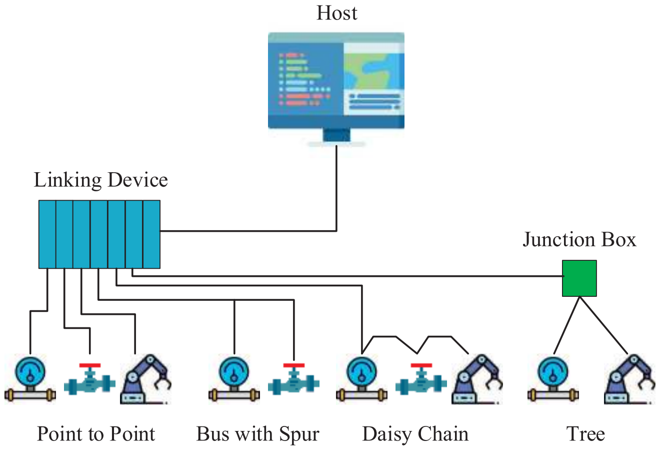


Fig. 3. Foundation Fieldbus H1 Topologies.

The supported topologies are shown in Fig. 3. A Linking Device acts as an interface between the host and the field devices. Different topologies can be realized, such as Point-to-Point, Bus with Spurs, Daisy Chain and Tree. In the Point-to-Point topology, each field device is connected directly to the Linking Device. The Bus with Spurs topology uses a single bus to which the field devices are connected. In the Daisy Chain topology, the field devices are connected in series with each other. In the Tree topology, a Junction Box is used as a concentrator, where several field devices connect to it. After concentrating the data from the devices the junction box forwards them to the Linking Device.

Foundation Fieldbus H1 supports three communication methods. The Publisher/Subscriber method is used for continuous, real-time data acquisition and it is scheduled at specific time intervals. The Client/Server method is mainly used when the operator accesses a specific device to modify variables, manages alarms and runs diagnostics. The Report Distribution method is used for alarms.

The five-layer architecture and packet structure of the Fieldbus Foundation H1 are shown in Fig. 4. The Data from the User Application Layer are encapsulated with a Fieldbus Message Specification (FMS) Protocol Control Information Field, in order to form a FMS Protocol Data Unit (PDU). Similarly, the FMS PDU is encapsulated with the Fieldbus Access Sublayer (FAS) PCI to form the FAS PDU. The Data Link Layer (DLL) PCI and Frame Check Sequence fields encapsulate the FAS PDU, in order to form the DLL PDU. Finally, the DLL PDU is encapsulated with the Physical Layer fields, namely Preamble, Start Delimiter, and End Delimiter.

PROFIBUS [14] is a Fieldbus-based industrial communication protocol, that was developed by PROFIBUS & PROFINET International. PROFIBUS specifies the Application, Data Link and Physical layers of the Open Systems Interconnection (OSI) model. In the Application layer, three service levels are defined. The first level provides the basic cyclic exchange of data and diagnostics. The second level provides acyclic data exchange and alarm handling, while the third level provides interval and broadcast data exchange.

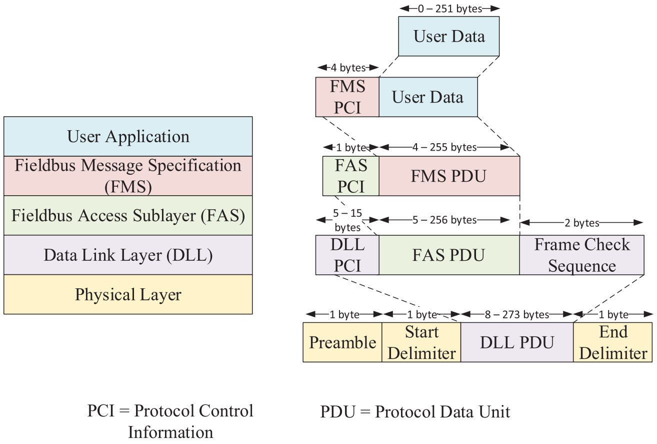


Fig. 4. Fieldbus Foundation H1 Network Stack.

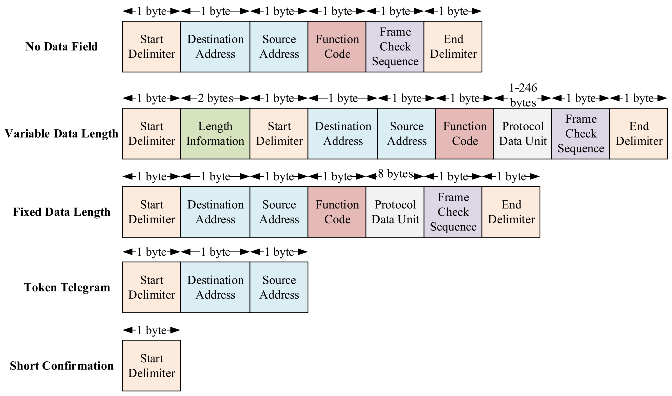


Fig. 5. Profibus Telegram Structure.

The Data Link layer provides a hybrid access method, combining token passing and master-slave schemes, is used for the transmission control. Finally, the Physical layer transmits the bits using twisted pair cables or fiber optics.

As shown in Fig. 5, the five telegrams are used for transmission control. The Start and End Delimiters mark the beginning and end of the telegram respectively. The Destination and Source Addresses are numbers ranging from 1 to 126, while the address 127 is used for broadcast addressing. The Function Code is used to select the function to be executed, while the Frame Check Sequence is used to check the integrity of the telegram. The application data are stored in the PDU field which has either variable or fixed length.

WorldFIP [15] is a Fieldbus network protocol designed to link SCADA devices and controllers. WorldFIP can be used in both synchronous and asynchronous applications. It specifies the application, the data link, and the physical layer. The physical layer relies on the bus topology and allows four transmission speeds, namely 31.25 Kbps, 1 Mbps, 2.5 Mbps, and 5 Mbps. The maximum wire length per segment is 1km and up to 64 nodes can be connected to it.

The structure of the WorldFIP frame is illustrated in Fig. 6. The Frame Start Sequence (FSS) marks the beginning, while the Frame End Sequence (FES) marks the end of the frame respectively. The Control field indicates the type of the frame. The Destination and Source addresses are 24-bit numbers, that are used to identify the devices, while broadcast addressing is not supported. Finally, the integrity of the frame is verified using the Frame Check Sequence (FCS).

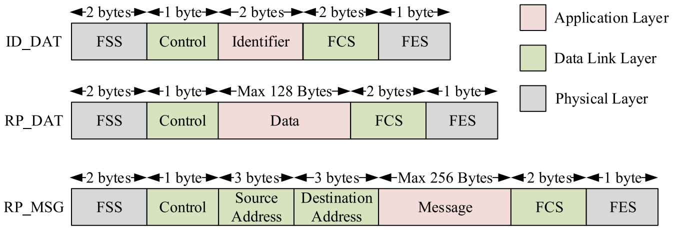


Fig. 6. WorldFIP Frame Structure.

C. Ethernet-Based Protocols

Ethernet [16] is one of the most acclaimed networking technologies. The ubiquity, cost efficiency and high flexibility of Ethernet are urging many industrial communication protocols to incorporate it into their solutions. To satisfy the very low latency requirement of industrial applications, the Industrial Ethernet standard was developed which utilizes a modified Media Access Control (MAC) layer.

Industrial Ethernet offers significant advantages over other types of industrial networks. It offers extremely increased speed compared to legacy serial communications, leveraging the capacities of Cat5e/Cat6 cables and optical fiber. Moreover, the error detection and correction functionalities of the Ethernet allow for increased connection distances. The ubiquity of the Ethernet enables the use of common network equipment such as access points, switches, and routers. In spite of the modified MAC layer, the conventional MAC addresses can be used for identifying the devices. In addition, the MAC broadcast address (FF:FF:FF:FF:FF:FF) can be used to send broadcast packets to all the devices of the network. Finally, Ethernet has the capability to form peer-to-peer architectures, which will replace the legacy master-slave ones.

The **Distributed Network Protocol 3 (DNP3)** [17] enables communication between components in process automation systems. It was developed for facilitating data exchange between various monitoring and control devices. It has a crucial role in SCADA systems, as it facilitates communications between Master Stations, RTUs and IEDs. The original protocol used a slow serial interface, but the latest versions support TCP/IP-based operation, which improves its more robustness, efficiency, and interoperability, at the cost of higher implementation complexity.

Fig. 7 shows the DNP3 layers within the OSI model. The Application layer organizes the transmitted data in fragments, which is a block of bytes that contains request or response data. Fig. 8 illustrates the structure of the DNP3 Fragment. The header starts with the Application Control field that contains information on how to construct and reassemble multiple fragments. The Function Code field specifies how the fragment should be processed by the receiver. The Response header contains an additional field named Internal Indications. Following the Application Request/Response Header, a number of DNP3 objects are, along with their associated headers

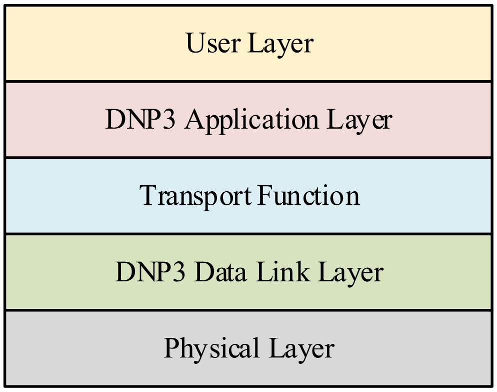


Fig. 7. DNP3 Network Stack.

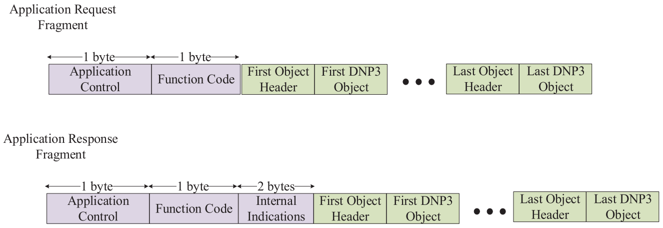


Fig. 8. DNP3 Fragment Structure.

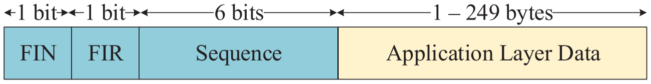


Fig. 9. DNP3 Transport Segment Structure.

are included into the fragment. They provide supplementary data that are required to complete the operation. In the Request fragment, only the DNP3 headers are included, as the master does not send any data. The Response fragment contains the same DNP3 headers followed by the associated DNP3 objects, which contain the data. The object header contains information on the data types and values, such as analog input value, binary event value, counter and time values.

The Transport Function is considered a sublayer of the Application layer that fits above the Data Link layer. The size of the DNP3 Application layer fragment may be larger than the size of the Data Link frame. The Transport Function is responsible for breaking the fragments into segments. A Transport segment (Fig. 9) is composed of the header and the application data. The header is composed of the FIN and FIR fields, which indicate whether it is the final or first fragment respectively, and the Sequence field, which is used to differentiate subsequent fragments.

Finally, the Data Link layer provides an interface between the physical media and the Transport Function and it is suitable for both User Datagram Protocol (UDP)/Internet Protocol (IP) and Transmission Control Protocol (TCP)/IP communication systems. The Data Link layer structures the Transport Function segments into data link frames and forwards them to the communication channel for transmission. In case of data reception, the transport segments are extracted from the incoming frames and passed to the upper layers. Moreover, Data Link layer manages data link frame synchronization, flow control, error handling, and link status probing. The frame format of the Data Link layer is shown in Fig. 10. The frame is composed of a header block and a number of data blocks. The header consists of the following fields: the Start field, which marks the start of the frame, the Length field, which indicates the size of user data, the Control field, which contains information regarding the flow control, and the type of the frame, and two MAC Address fields for Destination and Source respectively. A Cyclic Redundancy Check (CRC) field is appended at the end of the header and each data block.

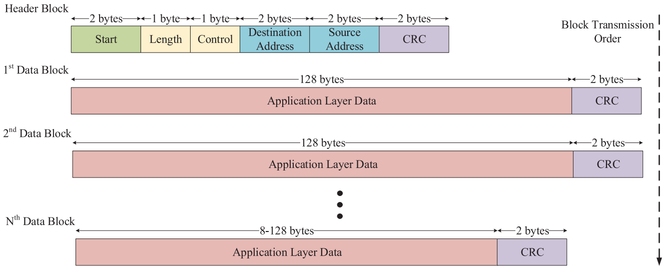


Fig. 10. DNP3 Data Link Frame.

EtherCAT [18] is an Ethernet-based protocol that is suitable for industrial real-time computing requirements. The use of plain Ethernet in automation technology has specific shortcomings. Firstly, the very high bandwidth is wasted, as each field device only sends and receives a few bytes of data. Moreover, the low computing capacity of the field devices is insufficient for embedding an Ethernet controller within the device. Finally, Ethernet has certain limitations regarding its real-time capabilities.

The main advantage of EtherCAT is that it does not require a specialized interface. Any commercially available Ethernet controller can be used as an EtherCAT master. Another important advantage is the conformity with the Ethernet standard. This enables EtherCAT to operate with standard network components such as Ethernet switches. Finally, the very short cycle time ($<100\ \mu s$) enables new applications with more accurate control.

EtherCAT considers the bus as a single large Ethernet device, which interconnects a number of EtherCAT slaves. The data transfer procedure is shown in Fig. 11. The master node initiates the data transfer by transmitting an Ethernet frame. A slave node extracts its own data from the frame, carries out the received command (such as reading data), insert new data to the frame and forwards it to the next node in the bus. The last node sends a frame, containing data from all the nodes, back to the master completing the cycle.

Fig. 12 shows the structure of a basic EtherCAT frame, compared to a basic Ethernet Frame. The size of the Ethernet frame ranges between 64 and 1522 Bytes and includes the Ethernet Header, the Ethernet Data, and the Frame Check Sequence field. In the case of the EtherCAT frame, the Ethernet Header includes the destination and source 48-bit addresses and the EthernetType value which indicates the encapsulated protocol in the payload. The value 0x88A4 is registered to Internet Assigned Numbers Authority (IANA) for the EtherCAT.

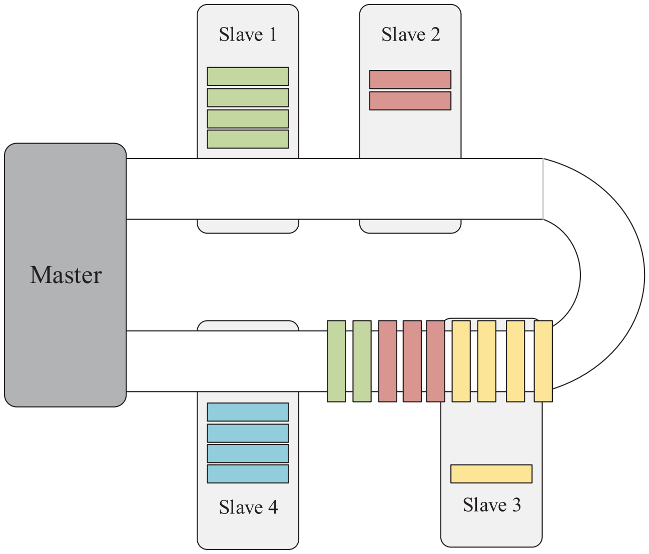
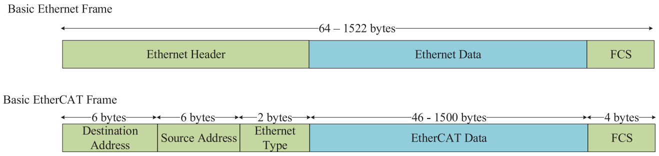


Fig. 11. EtherCAT Data Transfer Procedure.



Basic Ethernet Frame

64 – 1522 bytes		
Ethernet Header	Ethernet Data	FCS

Basic EtherCAT Frame

6 bytes	6 bytes	2 bytes	46 - 1500 bytes	4 bytes
Destination Address	Source Address	Ethernet Type	EtherCAT Data	FCS

Fig. 12. EtherCAT Frame.

EtherCAT supports four addressing methods, namely physical, logical, multiple and broadcast addressing. In physical addressing, which is mainly used for transferring parameter data, the telegram is precisely addressed to a single slave device. In logical addressing, slave devices are not addressed individually, and the logical address can contain any number of slaves. The master device maps the physical MAC addresses to logical addresses and the configuration is transmitted to the Fieldbus Memory Management Units (FMMU) of the slaves. In the multiple addressing method, physical address areas of several slaves can be addressed, by setting the multiple read flag in the telegram. The broadcast address is used to address all the slave devices of the network.

Foundation High-Speed Ethernet (HSE) [19] is an implementation of the Foundation Fieldbus H1 protocol that uses the Ethernet protocol. As it is shown in Fig. 13 both protocols can be incorporated in the same network, by using a Linking Device (LD), which acts as a bridge between the Foundation H1 and HSE devices. The Foundation Fieldbus H1 data are encapsulated in an Ethernet frame. Each device of the network is addressed using its MAC address.

The network stack and frame structure of the Foundation HSE is shown in Fig. 14. The highest layer contains the user application and data. The Field Device Access is an interface between the user layer and the field devices. The TCP/IP protocol is used at the transport and network layers, while the Data Link Layer is based on Ethernet. The frame consists of the Preamble, a Start Delimiter, the Destination and Source Addresses, the Length of the payload, the Payload, and the CRC.

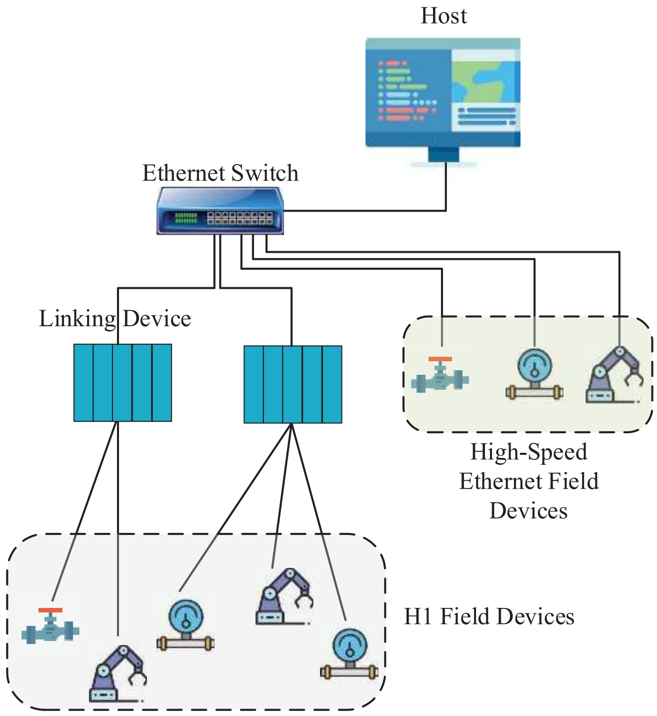
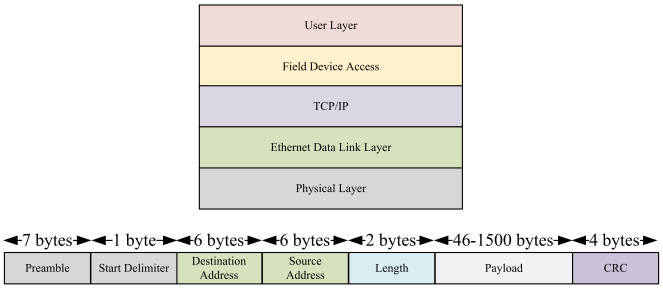


Fig. 13. Foundation Fieldbus H1 and HSE joint architecture.



User Layer
Field Device Access
TCP/IP
Ethernet Data Link Layer
Physical Layer

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes
Preamble	Start Delimiter	Destination Address	Source Address	Length	Payload	CRC

Fig. 14. Foundation HSE Stack and Frame Structure.

International Electrotechnical Commission (IEC) 61850 is an international communications standard, that defines a set of services and functions that enable data exchange between the SCADA HMI and the field devices. It is a higher layer protocol that defines a hierarchical, object-oriented, data representation model. Each node in the model consist of data and attributes such as configuration information, naming, and diagnostic information. This data model introduces an abstract layer, which enables a client to browse and retrieve data from a device without knowing details and implementation of the device. MAC addresses are used to address the devices of the network.

The network stack of IEC 61850 is shown in Fig. 15. The time-critical messages are mapped directly to Ethernet frames using non-IP protocols. These messages include the Sampled Measured Values (SMV), the Generic Object Oriented Substation Events (GOOSE), the Generic Substation State Events (GSSE), and the Manufacturing Messaging Specification (MMS). MMS can be also transferred through TCP/IP connections, while the time synchronization

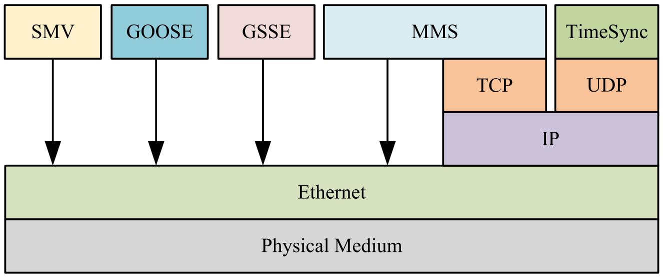


Fig. 15. IEC 61850 Network Stack.

(TimeSync) messages are transferred through UDP/IP connections.

PROcess FIeld NET (PROFINET) [20] is an industrial standard for data exchange over Industrial Ethernet, aiming to enable data collection and equipment control, under tight time constraints. It is a higher layer protocol that defines the Application and Presentation layers of the OSI model. It supports three different communication methods: a) Non-Real-Time, which is used for non-time critical data with cycle times in the range of 100ms, b) Real-Time which is used for time-critical data, by utilizing a communication channel with small cycle times of 10ms, and c) Isochronous Real-Time which supports cycle times lower than 1ms, by dividing the communication cycle and reserving the slots to specific nodes. The communication is based on Ethernet technology, while the MAC address is used for device addressing.

SERCOS III [21] is a standardized open digital interface for the communication between industrial controls, input/output devices and standard Ethernet nodes. It operates in master/slave configuration exchanging cyclic data between nodes. Sercos III uses two types of telegrams in order to accomplish the data exchange: the Master Data Telegram (MDT), which contains information sent by the master to the slaves and the Acknowledge Telegram (AT), which is issued by the master and the slaves insert the appropriate response data in it. Each device is equipped with two ports, namely P1 and P2.

SERCOS III supports two main network topologies, which are illustrated in Fig. 16. The line topology is a simpler and cheaper topology, as all devices are connected using a single cable, but it provides no redundancy. In the line topology, the master's P1 port is unconnected, while P2 port is connected to the first slave. The master initiates the data exchange by sending the telegram to the first slave. The slave reads the telegram, executes the required functions, inserts its data in the telegram and forwards it to the next slave. The final slave detects that its second port is unconnected and reverses the telegram forwarding procedure until it reaches the master.

In the ring topology, the master's P1 port is connected to the P2 port of the last slave. The master automatically detects the existence of the ring topology and transmits two counter-rotating telegrams. This topology enforces tighter synchronization, as well as automatic infrastructure redundancy. In case of a link or device failure, the infrastructure will automatically be reshaped to line topology, as the last slave will have its second port unconnected. This scenario is illustrated in Fig. 16, where the link between two slaves is severed and the infrastructure is reshaped into two line topologies.

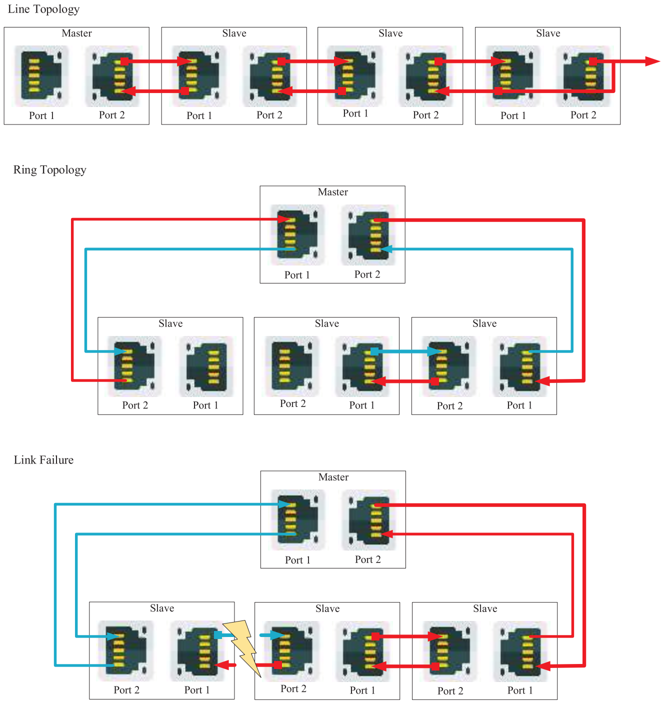


Fig. 16. SERCOS III Network Topologies.

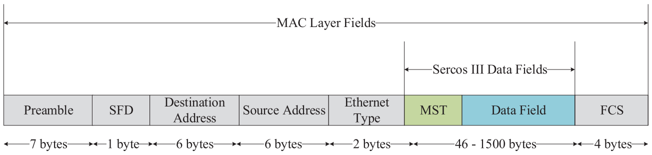


Fig. 17. SERCOS III Telegram Structure.

The structure of the SERCOS III telegram is shown in Fig. 17. The frame starts with the preamble, followed by the destination address, which is set to the Ethernet broadcast MAC address, and the source address, which is set to the MAC address of the master. The Ethernet type is set by the Field Registration Authority to 0x88CD. The encapsulated telegram consists of the SERCOS III header, which contains status and control information and the varying data field, that stores the variables for each device. Finally, the Forward Checking Sequence (FCS) field is appended for error detection.

Ethernet Powerlink [22] is a loyalty-free real-time industrial communication protocol, managed by the Ethernet Powerlink Standardization Group. Ethernet Powerlink implements mixed polling and time-slicing mechanisms over the traditional Ethernet in order to provide a guaranteed transmission of time-critical data, a high precision time synchronization of the devices, and an asynchronous channel dedicated to the transmission of less time-critical data.

The Ethernet Powerlink communication cycle consists of two phases, namely the isochronous and the asynchronous phases. A Start of Cyclic frame is used in order to synchronize all the devices. The node synchronization mitigates the

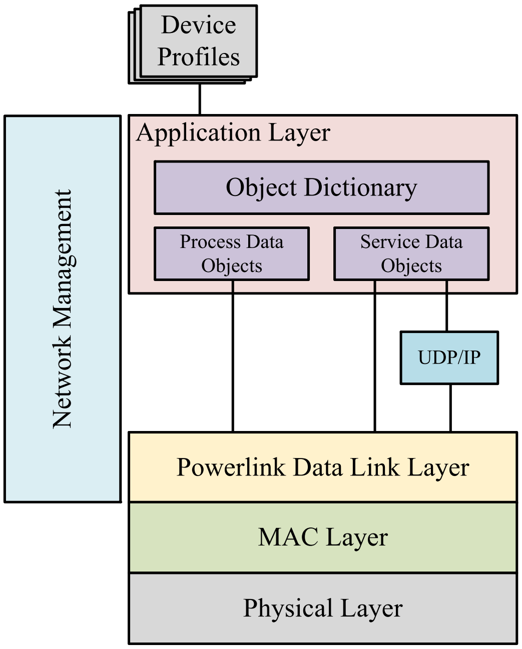


Fig. 18. Ethernet Powerlink Stack.

frame collision and ensures real-time communication. In the isochronous phase, the master device, called Managing Node, polls the Controlled Nodes cyclically. After all the Controlled Nodes have been polled, the asynchronous phase starts to allow the transmission of less-time critical data. As Ethernet Powerlink is based on the traditional Ethernet, each device has a unique MAC address. In addition, a logical node ID is assigned to each device.

The stack of the Ethernet Powerlink is shown in Fig. 18. The highest layer includes the device profiles, which define the properties of each device. The Application Layer contains: a) the Object Dictionary, which enables the application to expose the data, parameters, and services to the network, b) the Process Data Objects, which contain the values of the objects and they are cycled among network devices in the isochronous phase, and c) Service Data Objects, which are used to establish an asynchronous connection between the nodes. The Powerlink Data Link Layer is responsible for establishing communication between the network nodes. It also defines the Managing Node that is in charge of moderating access to the shared medium. Moreover, it provides isochronous and asynchronous communication channels as well as time synchronization and network management services to the upper layers. Finally, the rest of the layers are the same as in the traditional Ethernet.

Fig. 19 shows the structure of the Ethernet Powerlink frame, which is encapsulated in a traditional Ethernet frame, with an EtherType value of 0x88AB. The Powerlink frame consists of the Message type, which determines the purpose of the frame, the Powerlink destination and source addresses, and the Powerlink Payload.

Real-time Automation Protocols for Industrial Ethernet (RAPIEnet) [23] is an international standard for real-time data transmission that was developed in South Korea. RAPIEnet

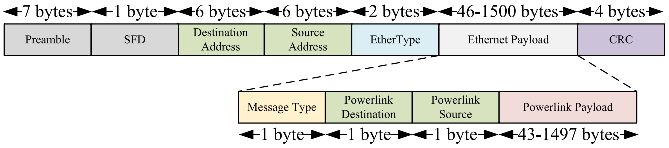


Fig. 19. Ethernet Powerlink Frame Structure.

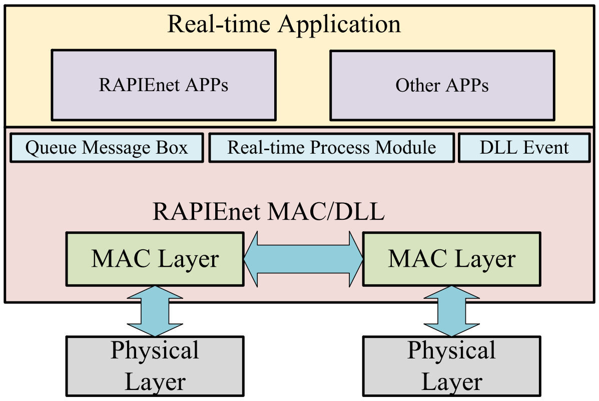


Fig. 20. RAPIEnet Protocol Stack.

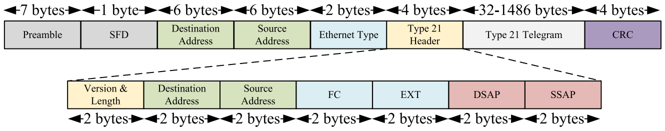


Fig. 21. RAPIEnet Frame Structure.

supports unicast, multicast, and broadcast addressing. Each RAPIEnet device features an embedded Ethernet switch with two ports in order to enable the daisy-chain and ring topologies. Figs. 20 and 21 show the stack and the frame structure of the RAPIEnet protocol, respectively. The RAPIEnet frame starts with the Preamble and the Start Frame Delimiter (SFD), followed by the Destination and Source addresses. The Ethernet Type field is used to select the type of the frame. The Type 21 header includes the protocol version and length of the telegram, the Destination and Source addresses, the requested Function Code (FC) and the corresponding function extension (EXT). The Destination Service Access Point (DSAP) and Source Service Access Point (SSAP) are appended to the end of the header. After the Type 21 Header, the Type 21 telegram is included, followed by the CRC.

D. Serial-Based Protocols

The **IEC 60870** is a set of standards which define the systems used for remote control and monitoring in electrical applications and power system automation. The IEC 60870-5 specification document defines the communication specifications and consists of a set of companion standards.

The IEC 60870-5-101 companion standard is mainly used in the energy sector. It mainly utilizes the asynchronous V.24 interface, which supports data rates of up to 9600 bps, while the X.24 and X.27 interfaces enable data rates up to 64000 bit/s.

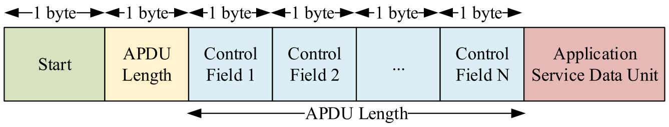


Fig. 22. APCI Frame Structure.

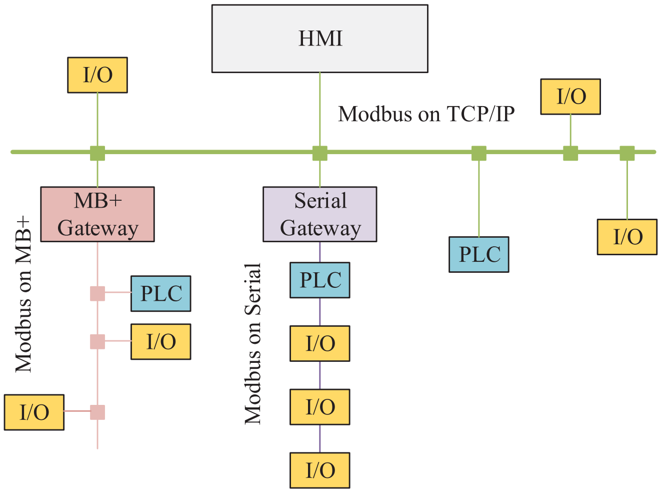


Fig. 23. Modbus Network Architecture.

The IEC 60870-5-103 companion standard mainly utilizes the asynchronous V.24 (RS232) and RS485 interfaces, which feature data rates of up to 19200 bps. The companion also includes specifications regarding interfaces that support fiber optics.

The IEC 60870-5-104 defines the Application layer of the OSI model and uses the conventional Ethernet transport technology. Various network types can be realized within TCP/IP, such as X.25, ATM (Asynchronous Transfer Mode), FR (Frame Relay), and serial point-to-point (X.21). Data are stored in an Application PDU (APDU), while the APDU along with an optional Application Service Data Unit (ASDU) are encapsulated in an Application Protocol Control Information (APCI) frame, as shown in Fig. 22. The APCI frame starts with a Start byte, followed by a field denoting the length of the APDU. A number of control fields are appended based on the APDU length.

Modbus [24] is one of the most used communication protocols for the interconnection of industrial devices, due to its industrial focus, the easy deployment and maintenance, and open specifications. Modbus also enables communication between devices on different network infrastructures. Fig. 23 shows a reference architecture of a Modbus network. Different types of field devices (e.g., Programmable Logic Controller (PLC), HMI, I/O devices) connect to the same network by using different Modbus variants such as Modbus+, Serial and TCP/IP. The MB+ and Serial Gateways are used as converters between the Modbus variants.

Fig. 24 illustrates the structure of a Modbus frame. The general frame form, called Application Data Unit (ADU), encloses a PDU along with fields reserved for device addressing and error checking. The PDU consists of the function code field,

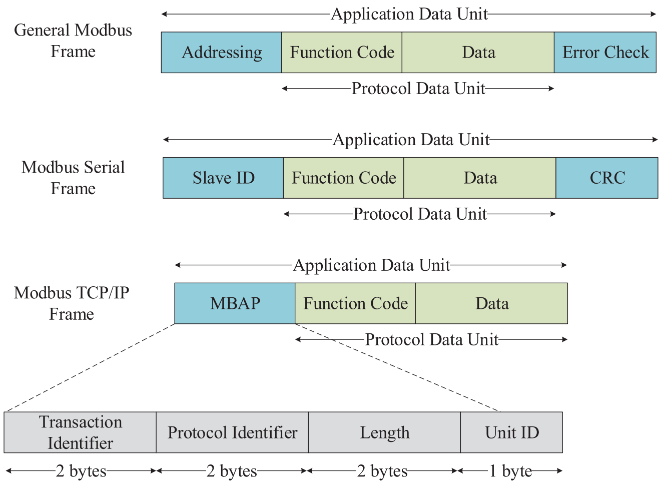


Fig. 24. Modbus Frame Structure.

which is used to select the operation, while the Data field size depends on the selected function. The addressing and error checking fields vary depending on the transport technology. In the Modbus serial variant, the addressing field contains the Slave ID and utilizes CRC for error detection. In case of Modbus TCP/IP, the addressing field is replaced by the Modbus Application Protocol (MBAP) Header, while the error check field is removed as the error detection capabilities of the TCP/IP protocol are leveraged. The complete Modbus ADU is encapsulated into the data field of a standard TCP/IP frame.

Serial Modbus enables message exchange between master and slave devices over serial communication mode. The master device coordinates the communication and can directly address the devices. The slave devices monitor the channel for requests from the master and respond accordingly.

In TCP/IP communication, the Modbus TCP/IP ADU contains the MBAP header and the PDU. The header includes the following fields: The Transaction Identifier, is used for logically pairing the transactions that are carried out in the same TCP stream. The Protocol Identifier, is always set to 0, while the Length field indicates the size of the remaining fields. Finally, the Unit Identifier is used to identify hosts that belong to networks, for example in case of bridging TCP/IP and serial networks.

Unitronics PCOM [25], [26] is a communication protocol that enables applications to communicate with PLC devices, based on requests and responses. The applications poll the PLC using command codes to identify the type of operation (e.g., read memory register). PCOM supports inter-PLC communication in master-slave schemes, where the master forwards the request/replies to/from the slave PLCs. In addition, PCOM also supports administrative operations that can be used to manage and reprogram the PLC.

The message structure of the PCOM protocol is shown in Fig. 25. It supports two message modes, namely ASCII and Binary. In the ASCII mode, only one type of operand per request is allowed, contrary to the Binary mode, where multiple types of operands are allowed. The STX and ETX fields denote the start and the end of the transmission,

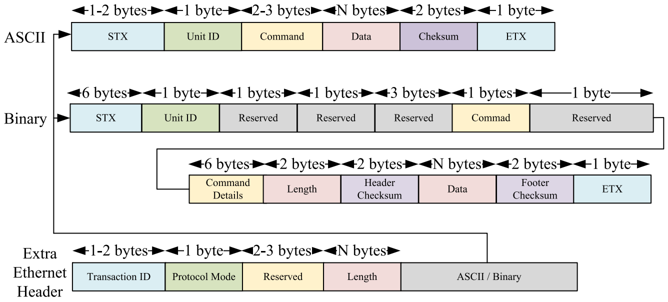


Fig. 25. PCOM Message Structure.

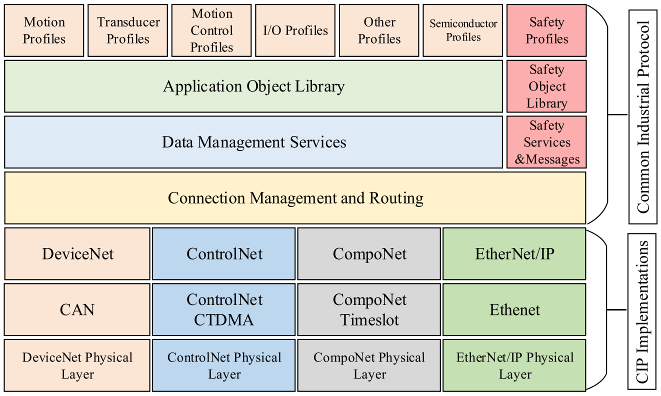


Fig. 26. Common Industrial Protocol Stack.

respectively. The Unit ID is used to address the PLC device, while the Command field is used to select the command to be executed. Finally, the Checksums are used for error checking. PCOM can also support Ethernet-based communications, by adding an extra header between the Ethernet header and the ASCII or Binary message.

E. Common Industrial Protocol

The **Common Industrial Protocol (CIP)** [27] is a peer to peer protocol that provides communications infrastructure for industrial applications. The stack of the protocol is shown in Fig. 26. The top level of the stack includes a number of device profiles, which are defined in order to increase device interoperability and consistency across multiple device vendors. The Application Object Library provides an application interface, where each object has as set of attributes (data), services (commands), and behaviors (reactions to events). The Data Management Services define the addressing models for the CIP entities, along with the supported data types. The Connection Management and Routing layer defines the mechanisms that enable the transmission of messages across multiple networks, and acts as an interface between the higher and lower protocol layers. In order to enhance the protocol's security, three additional higher level layers are included, namely Security Profiles, Safety Object Library, and Safety Services & Messages. Concerning the lower layers, four network and transport layer protocols are supported by CIP.

DeviceNet was the first implementation of CIP and it is based on the Controller Area Network (CAN) protocol [28]. The nodes are connected in a trunkline/dropline topology. In this topology, there is a main trunkline running across the field. A node is added to the network, by using a tap to insert the device in the trunkline. A 11-bit number is used for addressing the network devices, while broadcasting is not supported.

ControlNet is a digital communications protocol that provides high-speed transport of time-critical data. It forms a Producer/Consumer network that supports multiple communication hierarchies and message prioritization. ControlNet uses a Concurrent Time Domain Multiple Access (CTDMA) mechanism in order ensure the precise time for message delivery. For the addressing, each device is assigned a number ranging from 1 to 99, while broadcasting is not supported.

CompoNet provides high-speed communication among controllers, sensors and actuators. It forms a master/slave network, where the communication is scheduled in timeslots. A 16-bit number is used for addressing the network devices, while broadcasting is not supported.

EtherNet/IP is CIP implementation that is based on the Ethernet standard. EtherNet/IP is a data link layer protocol that encapsulates the CIP messages in an Ethernet frame, while MAC addresses are used for the device addressing. It employs TCP/IP for flow control, fragmentation reassembly and message acknowledgment, and UDP for transporting messages that contain time-critical control data.

The **Highway Addressable Remote Transducer (HART)** is an industrial communication protocol that supports both analog and digital communications. The data are modulated using Frequency Shift Keying (FSK). The digital signal consists of two frequencies, 1.2 KHz and 2.2 KHz for bits 1 and 0 respectively. The analog signal is superimposed with the waves of these two frequencies in order to provide simultaneous analog and digital communication.

It supports Point-to-Point and Multi-drop topologies in Master/Slave configuration. In the Point-to-Point topology, both the analog and digital signals are used. The 4-20 mA analog signal is used for reading a single value, while the digital one is used for accessing multiple values, and maintenance and diagnostic operations. In the Multi-drop topology, a two-wire system is used for connecting the field devices. The analog signal is used for powering the field devices and the data exchange is completely digital. For the device addressing, a 4-bit number is used, while in newer protocol versions 38-bits are used. HART does not support message broadcasting.

The structure of the HART packet is illustrated in Fig. 27. A preamble is used for carrier detection and synchronization. The Start field marks the beginning of the packet, while the Address field specifies the address of the master and slave devices. The Command byte represents the command to be executed by the slave devices. The Data Size field specifies the size of the user data. Finally, the Checksum byte is a XOR operation of all the bytes beginning from the Start field up to the last byte of the Data field.

DC-BUS is an analog protocol that enables reliable communication over noise Direct Current (DC) or Alternating Current (AC) power lines. DC-BUS operates on the physical layer and enables the transmission of data over the power lines even if the signal is extremely attenuated due to the line noise.

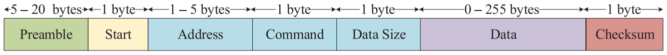


Fig. 27. HART Packet Structure.

DC-BUS enables byte-oriented and message-oriented communication, while a sleep mechanism is implemented in order to enable low power consumption.

Byte-oriented communication transfers a single Universal Asynchronous Receiver/Transmitter (UART) bytes over high frequency noisy channels, with a datarate up to 115.2 Kbps. A unique narrowband signal modulation is used, based on the combination of phase changes. Message-oriented communication offers a datarate up to 1.3 Mbps. In addition, a collision detection mechanism is used, while a narrowband carrier is selected in order to communicate over the power lines.

III. SCADA Security Incidents, Objectives and Threats

In this section, we report certain high-impact security incidents that affected SCADA systems, we discuss the security objectives and threats, and we provide a detailed overview of several SCADA security testbeds.

A. Security Incidents in SCADA Systems

Reports in [29] and [30] are showing an increasing number of security incidents and cyber attacks against critical SCADA infrastructure. Consequently, security considerations for SCADA systems are gaining higher priority and consideration than those for traditional IT systems due to the potential impact on the physical safety of employees, customers, or communities.

The Repository of Industrial Security Incidents (RISI) [31] contains 228 reported incidents dating from 1982 to 2014. Each data entry contains the year, title, industry type, country, and information about the incident and its impact. RISI tracks all incidents of cybersecurity nature that affect SCADA systems and control processes. Therefore, RISI includes events such as accidental cyber-related incidents, as well as deliberate events such as internal and external attacks, Denial of Service (DoS) attacks, virus/worm infiltrations, remote access attacks, and any other cyber incident that impacted the process environment. Table II lists the number of reported incidents in each country, based on the RISI.

A list of certain high-impact SCADA security incidents is provided in Table III. The list is ordered by the year, when the incident took place. The table also lists the methods that were used to launch the attack, and the result or aim of the attack.

In 2000, an employee of Maroochy Shire in Queensland gained unauthorized access to the waste management system and spilled a large amount of raw sewage into rivers and parks, resulting in loss of marine life [32]. The Hunter Watertech PDS Compact 500 RTU was installed in each of the pumping stations, that was capable of receiving instructions and transmitting alarm and data signals to the control center. This particular RTU utilizes the DNP3 communications protocol.

TABLE II
Reported Incidents by Country

Country	Number of Incidents
United States	123
United Kingdom	32
Unknown	16
Canada	14
Australia	12
Japan	5
Switzerland	4
Iran, Russia	3
France, India, Ireland, Israel, New Zealand	2
Brazil, Chad, Germany, Guam, Guyana, Other	1

In 2003, The Slammer worm [33] disabled monitoring system of the Ohio Davis-Besse nuclear power plant [34]. The worm infection originated from the unsecured network of a third-party collaborating company and penetrated the Davis-Besse's network by exploiting a vulnerability in the Microsoft Structured Query Language (SQL) Server 2000 through the network port 1434. The worm was discovered when the operators noticed the network's slow performance. The power plant was out of commission, so the incident had not any hazardous consequences.

In 2003, the SoBig virus [35] was responsible for shutting the system that manages the train signals in Florida, U.S., [36]. hlThe virus managed to widely spread through e-mail attachments and infect the computers that control the SCADA systems. SoBig contained its own implementation of the Simple Mail Transfer Protocol (SMTP) and managed to quickly propagate. In addition, a variant of the virus established a connection through UDP port 8998 and downloaded the WinGate proxy server. Except for the train delays, no major incidents were caused.

In 2007, a malicious software was installed on the Tehama Colusa Canal Authority SCADA system [37] by a former employee. No details were published about the malicious software, the infected systems, and the damages caused.

Chinese and Russian spies were reported to have hacked the U.S. electrical power grid in 2009 [38]. The spies aimed to gain information about critical infrastructure specifications using network mapping tools. The communications throughout the U.S. power grid are enabled by various well-known protocols such as Modbus, DNP3, and IEC 61850 [39]. However, the technical details of the attack still remain vague.

A Carrell Clinic, Dallas security contractor in 2009 installed malicious software on clinic computers causing disruption of heating, ventilation, and air-conditioning (HVAC) systems [40].

The Stuxnet computer worm [41] was identified by VirusBlokAda, a security firm based in Belarus [42], in June of 2010. Stuxnet's aim was to sabotage the uranium enrichment facility at Natanz, where the centrifuge operational capacity had dropped over the past year by 30 percent [43]. The attack may have caused the destruction of fast-spinning centrifuges, however, this has never been confirmed. The worm exploited

TABLE III
SCADA SECURITY INCIDENTS

Year	Name	Method	Result/Aim
2000	Maroochy Water System	User Compromise	Operation disruption
2003	Davis-Besee Nuclear Power Plant	Worm	Network disruption
2003	Florida Train Signaling Systems	Virus	Train scheduling disruption
2007	Tehama Colusa Canal Authority	User Compromise	Unkown
2009	US Electricity Grid	Infrastructure Mapping	Unkown
2009	Dallas Carrell Clinic	User compromise	HVAC Equipment disruption
2010	Stuxnet	Worm, Root Compromise, Trojan	Disruption of operations, Equipment destruction
2011	Night Dragon	Social Engineering, User Compromise, Root Compromise, Spear Phising, Windows-based Exploits	Unauthorized access to control and information systems
2011	Duqu	Virus, Root Compromise, Windows-based Exploits	Industrial control systems data and information embezzlement
2012	Aramco	Virus	Service Disruption, Cyber espionage
2012	Flame	Worm, Windows-based Exploits	Cyber espionage
2014	Dragonfly Campaign	Worm, Trojans, Backdoors, Spear Phising	Cyber espionage
2016	Ukrainian Power Grid	User Compromise, Trojan, Worm	Service Disruption
2017	Dragonfly 2.0	Phising, Malicious email attachments, Trojan	Cyber espionage, Equipment Destruction, Unauthorized information disclosure
2018	Saipem Company	Virus	Service disruption

a vulnerability in the Server Message Block (SMB) in order to propagate itself to systems having the Siemens' SIMATIC Step7 SCADA control software, that is used to configure the Siemens S7-300 PLC [44]. Afterward, the worm propagated to the PLC and compromised the Profibus-based monitoring system. Stuxnet unveiled the real threat of cyber-warfare, as it is believed to be the first cyber-weapon that aims to exploit SCADA systems.

The security firm McAfee reported a number of coordinated cyber attacks against oil firms [45]. The attacks, code-named 'Night Dragon' are believed to originate from China and have been going on for over two years. The attackers penetrated the perimeter security controls through SQL injection attacks and compromised the DeMilitarized Zone (DMZ) and firewalls. In addition, a Remote Administration Tool (RAT) was installed that enabled the attackers to completely control and spy on the organization's systems. The attacks were not aiming for the SCADA systems, however, the infrastructure controlling those systems was compromised.

A new malware similar to Stuxnet, named Duqu was discovered in 2011 [46]. Duqu exploits a zero-day vulnerability in the Microsoft Word software in order to compromise the system. After the compromise, Duqu can secretly download and execute additional malware tools [47], in order to launch reconnaissance attacks against critical SCADA systems. The malware usually aims to compromise the control systems rather than the SCADA devices.

A series cyber attacks with the code name Dragonfly took place in 2014 [48], mainly targeting energy stakeholders. The targets of Dragonfly were petroleum pipelines, power generation plants, energy grid operators, and industrial hardware vendors. The attackers managed to compromise a number of equipment vendors and infected them with a trojan. The trojan was unintentionally installed by the operators, while they were installing software updates. The malware contained also a SCADA scanner module that searched for SCADA devices on TCP ports 102, 502 (Modbus port), and 44818. Additional attacks included spear phishing e-mails delivering malware and attacks that redirected visitors to fake websites hosting vulnerability exploit software.

In 2015, a series of cyber attacks against the Ukrainian grid caused power outages in the country [49]. The hacker group Sandworm was reported to have launched the attacks. The attackers sent a malicious Microsoft Excel document, which downloaded and installed a malware tool. The malware carried out Denial of Service attacks against SCADA controllers that resulted in the power outage across the country. In addition, the malware erased the infected systems' hard disks.

The Dragonfly 2.0 campaign launched a series of cyber attacks against a large number of energy companies in 2017 [50]. As the previous Dragonfly campaign, Dragonfly 2.0 utilized the same techniques and tools. In many cases, the hackers managed to successfully gain control of the company network, by compromising a software that the operators use to send commands to energy equipment, such as circuit breakers. During the first campaign, the attackers aimed to steal information about critical infrastructures, whereas in the second campaign they aimed to destroy the compromised equipment.

TABLE IV
SCADA PROTOCOL VULNERABILITIES

	Authentication Control	**Encryption Techniques**	**Integrity Check**	**Anti-replay Mechanisms**
BITBUS	✗	✗	✗	✗
Common Industrial Protocol	✗	✗	✗	✗
Distributed Network Protocol 3	✗	✓	✓	✗
Foundation Fieldbus H1	✗	✗	✗	✗
Foundadion HSE	✗	✗	✓	✗
HART	✗	✗	✓	✗
IEC 60870	✓	✗	✗	✗
IEC 61850	✓	✗	✗	✗
Modbus	✗	✗	✓	✗
PROFIBUS	✓	✓	✓	✗
PROFINET	✓	✓	✓	✗
SERCOS III	✓	✗	✓	✗
WorldFIP	✗	✗	✓	✗

B. Security Objectives and Vulnerabilities

Authors in [51] provide the security objectives, namely availability, authorization, authentication, confidentiality, integrity, and non-reputability. Availability refers to ensuring that the automation, control, safety, and communication systems are always available to the authorized users. The authorization manages the user access to the system. The authorization mechanisms determine the legitimacy of a user and restrict illegitimate users to control the system. The authentication objective is concerned with determining the user's identity and privileges inside the system. The confidentiality objective prevents information exposure to unauthorized users. The integrity objective refers to preventing modification of information by unauthorized users. The non-repudiability objective refers to the ability to provide irrefutable evidence of who performed certain actions.

Ensuring the aforementioned objectives is vital to the security of the SCADA systems. However, there are certain vulnerabilities that an adversary can exploit in order to compromise the systems. SCADA systems often utilize common computer protocols and functions such as file transfer over the network and remote access. Unencrypted data exchange can be compromised by an attacker in order to gain sensitive information. Additionally, system application and services require certain open network ports. An adversary can use those ports to gain access the SCADA system, collect information about it, and gain administrative privileges. Moreover, the adversary can upload malicious code that exploits a vulnerable application and gain unauthorized access. During the development of the first SCADA systems, security awareness had limited consideration as the SCADA systems were isolated from other systems. However, newer SCADA systems are able to communicate with other networks. Therefore, an attack against the communication networks can be escalated into attack against the whole SCADA system.

Several research works have analyzed and assessed the vulnerabilities of SCADA communication protocols [52], [53], [54], [55], [56]. Table IV summarizes several protocol shortcomings that make it vulnerable to cyber threats. The Authentication Control is used to authenticate the devices of the network, while the Encryption Techniques are used to encrypt the data before transmitting them over the communication channel. The Integrity Check ensures that the messages are received correctly without being modified. The Anti-replay Mechanisms prevent adversaries to inject malicious traffic in the network, that is similar to the normal traffic.

C. SCADA Security Challenges

The study in [7] proposes seven recommendations to the public and private sectors regarding the SCADA system security. Additionally, various technical and non-technical security challenges have been identified. In this work, we present the technical security challenges.

One of the main challenges is the lack of mature security tools tailored to the requirements of SCADA systems. Contrary to traditional computer systems, SCADA systems have different security requirements, as well as low computational capabilities. Moreover, the security mechanisms are not always considered in the specifications of a device or protocol, potentially due to high implementation cost or low computational capability of the device.

Ensuring security for a huge number of network devices that are often deployed in wide geographical areas is also challenging. In addition, physical access to these devices may be unrestricted. Thus, exploiting these devices can allow an adversary to compromise the whole network.

Since SCADA systems typically monitor and control critical infrastructures, they are targeted by technically skilled and well-organized attackers, called adaptive persistent adversaries. Common adversaries include criminal organizations (e.g., terrorists) and rival companies that have the required resources to create novel undisclosed attacks.

The use of legacy devices and protocols can introduce vulnerabilities that an adversary can exploit. As legacy SCADA systems were isolated from the Internet, security measures were not always required. In addition, the utilized proprietary protocols may include security breaches, therefore they cannot be always trusted. Another important security factor is the fact that the lifecycle of SCADA systems is much longer than the standard computer systems.

TABLE V
ATTACK TYPES IN SCADA SYSTEMS

Attack	Targeted/Untargeted	Violated Objectives
Denial of Service	Targeted	Availability
Eavesdropping	Targeted	Confidentiality, Authorization
Man-In-The-Middle	Targeted	Authentication, Confidentiality, Integrity
System break-in	Targeted	Authentication, Authorization
Virus	Untargeted	Availability, Integrity
Trojan	Untargeted	Confidentiality, Authentication
Worm	Untargeted	Confidentiality, Integrity, Authorization

Over the last years, technologies used in standard computer systems are being adopted by SCADA systems. For example, relays and mechanical devices have been replaced by microcontrollers and electronic devices, respectively, while operating systems have been integrated into SCADA systems. Consequently, the SCADA systems have inherited the vulnerabilities of standard computer systems. Moreover, as the software is becoming more complex, the probability of implementation errors increases.

D. Attack Types in SCADA Systems

A cyber-attack is considered as an intentional violation of one or more security objectives. Cyber-attacks can be classified into untargeted and targeted. Untargeted are designed to exploit any vulnerable system they discover, while targeted attacks aim to compromise a specific system. An overview of SCADA attack vectors is provided in [57]. The authors classify the attacks as physical attacks against SCADA hardware, attacks against SCADA software, and attacks against SCADA communications. Table V shows a list of some common attacks along with their impact on the security objectives of the system.

A similar classification is presented in [58]. The authors proposed a cyber-attack framework to extend the attack landscape for critical infrastructure, consisting of four attack classes, namely traditional IT-based attacks, protocol-specific attacks, configuration-based attacks, and process control attacks.

The aim of a Denial of Service (DoS) attacks is to ravage the availability and operation of the system. These attacks work by aggressively using all of the available resources of a device, so it cannot respond to the other legitimate requests. The author in [59], grouped various DoS attacks based on the OSI model. These attacks aim at electric power systems, but they can also be launched against SCADA systems. Specifically, there are DoS attacks against the SCADA services running in the Application Layer, such as Simple Mail Transfer Protocol (SMTP) and Session Initiation Protocol (SIP) flooding, resource exhaustion, and requests with large payloads. Similarly, the presentation layer attacks include malformed Secure Sockets Layer (SSL) requests and Domain Name System (DNS) queries. Regarding the session layer, common attacks include TCP sessions with long Time-to-Live (TTL) times and connection flooding. SYN flooding and Smurf are well known DoS attacks against the Transport Layer. With SYN flooding an adversary sends a massive number of SYN requests and the device responds and allocates resources to each one of them. The Smurf attack sends Internet Control Message Protocol (ICMP) packets to the network's broadcast address. Consequently, all the devices receive the ICMP packets and send the corresponding reply. If the rate of ICMP packets is too high, the network will be flooded with reply traffic. Border Gateway Protocol (BGP) hijacking and ICMP fragmentation are common Network Layer attacks. MAC flooding is a Data Link Layer attack, where an adversary sends multitude Ethernet frames, each one containing different source MAC address, in order to exhaust the memory of a switch, where the MAC addresses are stored. Finally, the physical layer attacks consist of wireless signal jamming and physical damage of the devices.

By eavesdropping, the attacker violates the confidentiality of the communication, by intercepting the communications. This attack mostly affects wireless communication systems, as the radio signals spread in a large area and anyone can receive the signal and recover the message. Wired communication systems are also vulnerable to this attack by tapping to the wires using specialized hardware. However, it is more difficult to carry out this attack in wired systems, as the adversary must have physical access to the premises. In order to mitigate this attack, the message should be encrypted using a secure encryption algorithm that enables only the legitimate receiver to decrypt it.

In a Man-In-The-Middle (MITM) attack, the attacker acts between the endpoints of the communication as a legitimate user. Additionally to the confidentiality violation, the attack can also tamper with the exchanged messages. The MITM attack exploits an inherent vulnerability in the Address Resolution Protocol (ARP). The ARP protocol does not provide an authentication mechanism, so any device connected to the network can impersonate a device, while the other devices believe that they communicate with the legitimate one. This attack can be mitigated by authenticating each message and utilizing certificates in the connection establishment. In addition, IDS can monitor the network to detect any unusual events or behavior deviations.

A virus attack manages to bypass access control and authentication mechanisms by exploiting a legitimate user. Virus attacks are often untargeted and they aim to execute malicious code in the compromised system. Trojans are untargeted attacks that violate the confidentiality and authentication objectives. Their aim is to mislead a user of its true intent and deploy malicious software. Finally, a worm is malicious software which is designed to automatically propagate itself by discovering and exploiting the vulnerabilities of a system, without the user's involvement. Worm infections are untargeted and usually violate the confidentiality and authorization objectives of the affected systems. Usually, worms have the ability to launch subsequent cyber-attacks from the infected hosts. These attacks can be mitigated by deploying proper antivirus software and updating it regularly. This software scans the system, looking for malware samples that match with a number of pre-configured signatures. In addition, the personnel handling these systems should receive proper training on how to avoid infecting the system with these kinds of malware.

The aforementioned mentioned attacks can affect both conventional computer and SCADA systems. Also, the attacks can propagate from conventional computers to SCADA systems and vise versa. Regarding the communication protocols, almost all of the protocols listed in the previous section can be affected by these attacks. Specifically, the DoS attack works both at the network and the application layer, meaning that the protocols that are based on these layers are vulnerable. The MITM attack can also affect all the protocols, as it works in the network layer, so an adversary can impersonate a controller and send to the field devices, resulting in possible equipment destruction. Viruses, trojans, and worms work at the application layer and usually aim conventional computers. However, certain high-level SCADA devices can be affected by these attacks.

E. Protocol-Specific Attacks

In this subsection, we present attacks that exploit vulnerabilities in higher layers.In order to discover these attacks, we performed an extensive literature search.

1) MODBUS: A taxonomy of attacks against the Modbus protocol is presented in [60] and [61]:

- *Slave Reconnaissance:* A Modbus message that requests the status information from the device is sent in order to discover the network devices.
- *Remote Restart:* The attacker repeatedly sends a Modbus message that restarts the device and executes the power-up test.
- *Slave Reconnaissance:* A Modbus message that requests the status information from the device is sent in order to discover the network devices.
- *Remote Restart:* The attacker repeatedly sends a Modbus message that restarts the device and executes the power-up test.
- *Diagnostic Register Reset:* The attacker sends a message that clears all the counters and the diagnostic register of the field device. The device configuration is modified, resulting in the disruption of the diagnostic operations.
- *Network Scanning:* The attacker sends legitimate messages to all network addresses in order to obtain information about the devices.
- *Broadcast Message Spoofing:* The attacker broadcasts fake messages to all slave devices. This attack cannot be detected easily as no response messages are sent to the master device from the slaves.
- *Irregular TCP Framing:* The attacker injects improperly framed messages or modifies the legitimate ones in order to cause connection termination between two devices.
- *Response Replay:* The legitimate traffic between master and slave devices is captured by the attacker and is replayed in order to disrupt the communication between these devices and/or insert a new fake device into the network.
- *Response Delay:* In this attack, the response messages are delayed that the master device receives obsolete data from the slave devices.
- *RST Flood:* The attacker injects a spoofed TCP packet with the RST flag set in order to close the TCP connection between two devices.
- *FIN Flood:* This attack involves injecting a spoofed TCP packet with the FIN flag set in order to terminate the TCP connection between two devices.

2) DNP3: As the DNP3 does not employ authentication and authorization mechanisms, all messages are assumed to be valid. Therefore, SCADA networks that rely to the DNP3 are susceptible to various attacks. The authors in [61] and [62] provide a taxonomy of attacks on the DNP3 protocol:

- *Reset Function Attack:* The attacker sends a message that causes the device to restart, making it unavailable for a period of time.
- *Transport Sequence Modification:* The attacker modifies the frame sequence field to inject spoofed messages in order to disrupt the communication.
- *Write Attack:* The attacker sends a message that writes data objects to a device and corrupts the data stored in the device memory.
- *Clear Objects Attack:* The attacker sends a message that clears the device memory, therefore erasing critical operation data.
- *Configuration Capture Attack:* The attacker sends a message with the fifth bit set in the second byte of the Internal Indications, which denotes that the configuration file of the device is corrupted. Consequently, the master device sends a new configuration file, which can be intercepted by the attacker.
- *Length Overflow Attack:* In this attack, an incorrect length field value is inserted that affects message processing. This can lead to data corruption and unexpected actions such as device crash.
- *Destination Address Tamper:* The attacker can tamper the destination address field in order to reroute requests and/or replies to other devices.
- *Unavailable Function Attack:* The attacker sends a message to the master device indicating that a slave is not functioning. Therefore, the master will assume that the device is unavailable and will stop sending requests.
- *Application Termination:* In this attack, a message that terminates the applications running in a device is sent. Consequently, the affected devices will not respond to the legitimate requests.
- *Fragmented Message Interruption:* The FIR and FIN flags indicate the first and final frames of a fragmented message, respectively. When the attacker sends a message with the FIR flag set, the previous incomplete fragments will be discarded.

An overview of SCADA attack vectors is provided in [57]. The authors classify the attacks as physical attacks against SCADA hardware, attacks against SCADA software, and attacks against SCADA communications.

TABLE VI
SCADA TESTBEDS

	Type	Protocol	Software Used	Attacks
[64]	Physical	IEC 61850, Modbus, DNP3	N/A	N/A
[65]	Simulated	Modbus	PowerWorld, RINSE, Custom protocol converter	DDoS
[68]	Simulated & Physical	N/A	Simulink, OMNET++	DDoS, MITM
[70]	Simulated & Physical	Modbus	OMNET++, Lego Mindstorms NXT	DDoS
[72]	Simulated & Physical	Modbus	N/A	No attacks performed
[73]	Simulated	Modbus	PowerWorld, Modbus RSim, OPNET	Command Injection, DoS
[76]	Simulated	Modbus	CORE Emulator	DoS, MITM
[78]	Simulated & Physical	DNP3, IEC 61850	Real Time Digital Simulator platform, DIgSILENT PowerFactory	Command Injection, DoS
[82]	Simulated	N/A	Emulab	DDoS

F. SCADA Security Testbeds

In this subsection, we present the SCADA testbeds that were developed in order to assist the research regarding SCADA security. We have performed an extensive literature search for surveys (e.g., [63]) and technical papers regarding SCADA testbeds. A summary of the proposed SCADA security testbeds is shown in Table VI. The Type column indicates whether the testbed is physical, simulated or both, while the Protocol column lists the implemented protocols. The software that was used to simulate the SCADA network and devices is listed in the Software column. Finally, the Attack column shows the attacks that were used for testbed evaluation.

The United States Department of Energy established the National SCADA testbed program in order to improve the security of SCADA systems used in the nation's critical energy infrastructures [64]. The program offers integrated expertise and resources of multiple national laboratories, including Idaho National Laboratory, Sandia National Laboratories, Argonne National Laboratory, Pacific Northwest National Laboratory, and Oak Ridge National Laboratory.

Authors in [65] developed a vulnerability assessment testbed for SCADA systems. The architecture consists of three simulated components: The Network Client provides a graphical view of the system states with the ability to control the component elements. The PowerWorld server [66] simulates the operation of the power grid, while the Rinse tool [67] provides a realistic simulation of a large network. A custom protocol converter software was developed to convert the PowerWorld protocol into the Modbus protocl. The authors carried out a Distributed DoS (DDoS) attack against the testbed, to study the effect of the attack.

Giani *et al.* [68] described the architecture of a SCADA testbed, that will help in designing and testing solutions to cyber attacks against SCADA systems. They envision three different implementations: a) A single simulation-based implementation using a simulation framework such as Simulink. b) A federated simulation-based implementation in which each component of the architecture is simulated separately using different technologies, such as Speedup for plant simulations, OMNET++ [69] for network simulation and DEVS for simulating software modules. c) The implementation using real commercial SCADA devices. Finally, they planned to carry out different attack scenarios, such as DoS attacks on sensors, and phishing attacks against the exchanged data.

Authors in [70] proposed a modular SCADA testbed based on the Modbus protocol. The OMNET++ simulator and Lego Mindstorms NXT [71] are used to simulate components such as RTUs, MTUs, and HMI. The communication between the aforementioned devices and components is realized through the TCP version of the Modbus protocol. The authors demonstrated the testbed by performing a DDoS attack against a simulated water plant.

Authors in [72] describe the Mississippi State University SCADA laboratory, which was built to facilitate the research in the security area of SCADA systems. The testbeds consist of commonly used software and hardware components across a wide range of industrial applications. The testbeds are divided into 2 categories based on their infrastructure. There are 5 testbeds that are based on the serial version of the Modbus protocol and 2 testbeds that are based on the TCP version.

Mallouhi *et al.* [73] presented a testbed designed to facilitate the evaluation of security approaches for SCADA systems. The architecture is composed of four main components and the Modbus protocol is used to support the communications requirements. The Process Control component provides the main monitoring control functions of the SCADA system, through the Modbus client. The PowerWorld tool is used to simulate the electrical grid component, which consists of transmission lines, transformers, and generators. The Modbus RSim [74] is used for simulating Modbus PLCs, that monitor the elements of the electrical grid. The simulated connection between the Process Control and the Modbus PLCs is realized through the OPNET Modeler (now rebranded as Riverbed Modeler) [75]. In the first attack scenario, it is assumed that the attacker has compromised the HMI, while the second scenario involves DoS attacks against the communication network.

A virtualized SCADA security testbed is proposed in [76]. The CORE emulator [77] is used as a basis for providing the SCADA communication infrastructure. The Modbus HMI, master and slave components were integrated as modules in the CORE emulator. For demonstration purposes, the authors built a water distribution system and evaluated the impact of the DoS and MITM attacks in the system's performance.

Authors in [78] introduce the PowerCyber testbed located at Iowa State University. The testbed consists of three simulated components, namely control, communication, and physical system. The control component consists of the control center, which provides monitoring and management of the SCADA system and the RTUs which are as an interface with the power system simulations. The communication component enables the connection between the RTUs and the control center, by utilizing the DNP3 and IEC 61850 protocols. The physical system component performs power system simulation using the Real Time Digital Simulator platform [79], for performing real-time power simulation, and the DIgSILENT PowerFactory software [80], for performing non-real-time simulation. Three attack scenarios were used for evaluating the testbed. The first scenario is a command injection attack from a compromised RTU, the second and third scenarios are DoS attack originating from the external and internal network, respectively.

DETERLab [81] is a large-scale emulation facility for cyber-physical systems, geared towards cyber-security experimentation. It is based on the Emulab and aims to facilitate the research and development program focused on the deployment of novel methodologies and technologies for experimental research in cyber-security.

The following observations are made: a) Most of the proposals leverage simulation techniques in order to simulate the whole SCADA system or several components of the system. b) The Modbus protocol is used in almost all proposals to provide communication infrastructure. c) A range of commercial and open source software was used for performing the simulations. d) The most commonly implemented attack type is the DoS, followed by the MITM. There are certain proposals that implemented a command injection attack.

G. Discussion

In this section, we listed certain high-impact security incidents in SCADA systems, discussed the SCADA security objectives, analyzed the attack types against those systems and presented several SCADA security testbeds. There are over 200 reported security incidents, mainly in the United States. We distinguished certain high-impact incidents, in order to show the importance of protecting SCADA systems against cyber-attacks. The impact of those attack ranges from light service disruption to more serious, such as critical data interception and equipment destruction. Several of those attacks also had a direct effect on public health and safety, while others were successfully mitigated without having irreversible consequences.

According to the study in [7], the priorities of the security objectives are different between standard computer and SCADA systems (Fig. 28). Ensuring the confidentiality of user information has the highest priority in the security of standard computer systems. On the other hand, ensuring system availability has the highest priority in SCADA systems. Any violation of those objectives is considered as a threat against the system. An adversary can exploit the system's vulnerabilities in order to compromise the system, by violating these objectives. The SCADA protocols are vulnerable by design to the external networks (i.e., the Internet). For example, both Modbus and DNP3 protocols do not support any access mechanisms and the communication process is unencrypted. Any user or device that has access to the network can act as a legitimate machine and intercept data or inject malicious traffic.

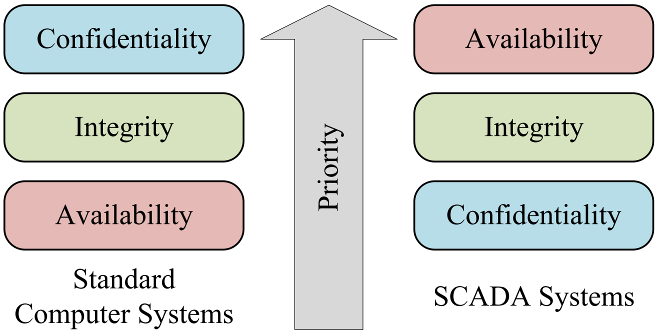


Fig. 28. Security Priority Comparison.

Similarly to conventional computer systems, SCADA systems are vulnerable to attacks aiming at the lower OSI layers (i.e., transport, network, and data link layers). However, the higher layer attacks vary, based on the SCADA communication protocol. The attacks can violate one or more of the security objectives and can be targeted or untargeted. Most of the attacks violate the availability and confidentiality objectives of the SCADA systems. There are several studies that analyze and categorize the attacks based on their type, protocol, and layer.

Lastly, this section includes a review of SCADA security testbeds that have been developed in order to assist security researchers. The testbeds are mainly simulated, or a combination of physical and simulated components. Most testbeds are based on the Modbus protocol as it is the most acclaimed protocol, due to its open specifications and high availability of hardware equipment. The software used for the simulation consists of open source and commercial tools that can simulate SCADA devices and large industrial networks. Finally, the testbeds were evaluated by testing them against common cyber-attacks, such as DDoS, MITM and command injection.

IV. SCADA Security Proposals

This section provides a thorough review of works that aim to secure SCADA systems. Firstly, we review the works that are not protocol-specific and can be realized with any SCADA system. Afterward, we review works that specifically target the Modbus, DNP3 and PROFINET protocols, respectively.

We adopted a review approach similar to the one presented in [83]. The approach (that is based on [84]) suggests exploring the most established literature sources, article databases and proceedings and carrying out backward and forward analyses to determine earlier relevant documents. We extensively searched the databases of well-known publishers in the areas of network and computer security, computer science, and industrial systems, including the Institute of Electrical and Electronics Engineers (IEEE), Elsevier, Springer, Association for Computing Machinery (ACM), and Wiley. The searching terms include the following keywords: "SCADA security", "SCADA intrusion detection", "SCADA cyberattacks", and "SCADA threats". The search returned 50.000 research papers that were published within the last decade. These papers were filtered by examining their titles and abstracts. The filtering resulted in about 120 research papers, which were manually

TABLE VII
SCADA ATTACK DETECTION SOLUTIONS

Reference	Approach	Methodology	Reported Challenges	Evaluation Results
[85]	Traffic Classification	Anomaly-based intrusion detection system based on the Error Back-Propagation and the Levenberg-Marquardt algorithms Performance evaluation based on recorded datasets, consisted of both normal and malicious traffic	Signature-based IDSs generate high number of false negatives IDSs should be continuously update in order to detect new attacks	100% detection rate 0% false positives
[88]	System Variable Inspection	System's state modeled after critical component measurement values Detection based on the system's proximity to critical state Evaluation in a real testbed considering accuracy and average detection time	Similar approaches cannot discriminate between accidental faults and cyber attacks	99% true positives 1% false negatives
[89]	Attack Detection	Proposal of an IDS that examines all layers of the SCADA system Performance evaluation in a real photovoltaic system connected to the grid	Most security countermeasures examine incoming traffic from external networks, without considering the traffic from the internal network	100% detection rate
[90]	Traffic Classification	Combination of two techniques for identifying the SCADA system's state Experimental evaluation in a simulated environment, considering the accuracy and computational complexity	Feeding the entire captured datasets to the IDS requires high amount of storage and computational power, in order to classify the traffic	The proposed approach has significant accuracy in detecting inconsistent states.
[91]	System Variable Inspection	Proposal of a graph-based detection scheme based on the relations of the variables among the SCADA system Performance evaluation using a simulated boiler of a real power plant	Most IDSs are inefficient to detect attacks that deeply target a critical control component The IDSs cannot detect an attack repeatedly performs slight modifications to compromised control components	100% for few affected variables 65% for many affected variables
[92]	Traffic Classification	Leverage of network telemetry data in order to classify SCADA network traffic Utilization of multiple classification algorithms Performance evaluation using simulated PLCs	Attack detection based on network telemetry can detect attacks, that are unable to be detected by considering only the application and network data	94.3% overall accuracy 5.70% false positives 0% false negatives
[98]	Traffic Classification	Combination of IWP-CSO for feature optimization and HNA-NN for classification Experimental validation in simulated network	Deployment of network based IDS in SCADA networks is challenging The high amount of dataset features produces high false alarm rate for unkown attacks	93.1% accuracy
[100]	Traffic Classification	Proposal of a multi-level approach The first level utilizes a Bloom filter, while the second one utilizes the k-nearest neighbor classifier	Most IDS cannot analyze SCADA traffic Real-world datasets are required in order train accurate classification models The analysis of network traffic is not always sufficient	97% accuracy 98% precision

analyzed in order to identify the ones relevant to the aim of this work. Consequently, 39 research papers were selected to be reviewed in-depth in this section.

A. SCADA Attack Detection Solutions

Table VII provides a summary of the security tactics that aim to secure a SCADA system. The Approach column describes the approach of the proposal as follows: The Attack Detection approaches aim to detect an ongoing attack and raise alerts. The Traffic Classification approaches process network flows and classify it as normal or malicious behavior. Traffic Encryption approaches leverage cryptographic algorithms to encrypt the data. The Methodology column of the table describes the specific methods or algorithms that were used, while the Testbed column provides information about the implementation of the testbed that validates the performance of the solution. The Reported Challenges column notes the various challenges that the authors encountered. Finally, the Evaluation Results column lists the overall accuracy of the proposed solution.

Linda *et al.* [85] presented an anomaly-based intrusion detection system based on the combination of two neural network algorithms, namely the Levenberg-Marquardt [86] and the Error Back-Propagation [87] algorithms. A window based feature extraction approach was adopted in order to extract certain key features from the packet header. The proposed detection system consists of the dataset construction and the process of training the neural network. During the dataset construction, both normal and malicious traffic are used. The training set is fed to the combination of the Levenberg-Marquardt and the Error Back-Propagation algorithms. The performance of the proposed approach was tested using recorded network traffic datasets, consisting of normal traffic and attacks generated from tools such as Nmap, Nessus, and Metasploit. The results show that the proposed approach achieved a perfect detection rate with no false positives.

Authors in [88] proposed an attack detection approach based on Critical State Analysis. The system's state is modeled after the values of certain critical components. By the continuous monitoring of the system, it can be predicted whether the system is heading to a critical state. Several tests were carried out in real testbeds in order to validate the of the proposed approach in terms of accuracy and average detection time. The results indicate an accuracy rate of 99% and less than 1% false positive rate.

Yang *et al.* [89] proposed a hierarchical multi-attribute IDS tailored to SCADA systems. The proposed IDS consists of the following components: a) The access-control whitelist, which examines the addresses in the ethernet, network and transport layers. If a corresponding source and destination pair is not in the whitelist, the IDS takes a predefined action, such as raising an alarm. b) The protocol-based whitelist, which only permits the traffic that complies with specific protocol specifications. c) The behavior-based rules that define normal behavior by performing deep packet inspection. The behavior rules are based on the correlation of relevant measured values, the time and frequency related constraints, the packet length, and the permitted function codes. If a packet fails to be validated by any of the aforementioned components, it is considered malicious. The experimental validation was performed in a real grid-connected photovoltaic system, by carrying out MITM attacks. The experimental results show that the proposed IDS successfully detected all the attacks with minimal latency.

An unsupervised anomaly detection approach was proposed in [90]. The proposed approach is a combination of two novel techniques: the identification between consistent and inconsistent data states, and the instantiation of rules regarding the detection of state proximity. The consistency of sensor measurements and actuator control data indicates the normal state of the system's operation, while any inconsistency will indicate malicious activity. The SCADA system specifications define the consistent data. The separation between consistent and inconsistent states is performed based on two assumptions. Firstly, the amount of consistent data is higher than the amount of the inconsistent ones. Secondly, the inconsistent data features are statistically different. After the state identification, detection rule extraction is performed. The extracted rules are able to fully represent the system states. The authors performed MITM attacks in a simulated water distribution system. The performance of the proposed approach was evaluated in terms of accuracy and computational complexity.

Wang *et al.* [91] proposed a method for detecting injection attacks based on the relations between the variables of the system. The proposed method consists of three steps. In the Component Analysis step, the internal relations between variables are analyzed, while in the Detection Model Generation step, a graph-based detection model for efficient detection is designed. Finally, in the Origins Inference step, the inference model detects the intrusions and indicates the possible origins. A simulated power plant boiler was used to evaluate the proposed approach. The values of the boiler were being recorded every second for 2000 seconds, while random variables were selected and injected with arbitrary data, within its valid range. The results show that the proposed approach successfully detected all the injection attacks, in cases that the affected variables were few. However, the detection accuracy dropped significantly in the case of injection attacks affecting many variables.

Ponomarev and Atkison [92] proposed an IDS that utilizes network telemetry to detect cyber attacks. The following network telemetry features were selected: response time, client-side and server-side dropped packets, elapsed time between dropped packet retransmission. In order to achieve high accuracy many classification algorithms were utilized, such as REPTree [93], Naive Bayes [94], Simple Logistic [95], Ripple-Down Rule [96], and J48 [97]. The evaluation testbed consists of simulated PLC units that generate both benign and malicious traffic. The results show that the proposed IDS achieves 94.3% overall accuracy, 5.70% false positives and no false negatives.

Authors in [98] presented two algorithms to detect intrusions in SCADA networks. The first algorithm, called Intrusion Weighted Particle based on the Cuckoo Search Optimization [99] (IWP-CSO), is used for extracting and optimizing the features obtained from the dataset. The second algorithm, called Hierarchical Neuron Architecture based Neural Network (HNA-NN), is used to perform the classification based on the optimized features. The performance evaluation was carried out in a simulated environment and considered different datasets. The combination of the proposed algorithms achieves an accuracy rate of 93.1%.

Khan *et al.* [100] proposed a multi-level approach for anomaly detection for SCADA systems. A Bloom filter constitutes the first level, where the packets are analyzed. If the signature of a packet does not match a set of pre-installed signatures, then the packet is consider malicious and it is dropped. The packets that have been considered as benign by the fist level will be forwarded to the second level. In the second level, the packets will be analyzed by a classifier k-nearest neighbors classifier. Similarly, the packets that will be classified as abnormal will be dropped. The authors carried out performance evaluation experiments using a real gas pipeline system dataset. The evaluation results indicate 97% accuracy and 98% precision.

B. Modbus

Table VIII summarizes the Modbus protocol security proposals, and shares the same format with VII. Cheung *et al.* [101] designed an anomaly detection-based Modbus IDS, that involves analyzing of TCP headers, pattern recognition, and data monitoring, combined with custom Snort rules. The construction of proper detection models is challenging, as it may lead to high false alarms. However, the communication patterns in SCADA networks present more static behavior than the common ones, so it is feasible to define the expected communication patterns. The authors conclude that a model-based intrusion detection is a promising approach for securing SCADA systems.

Authors in [102] describe a modified version of the Modbus protocol that utilizes anti-replay techniques and authentication mechanisms. The proposed module consists of four components. The Stream Builder which extracts the payloads from the packet stream and sends them to the other components, the Encryption/Decryption Unit which uses the RSA algorithm to encrypt and decrypt the payloads, the SHA-2 Validator that validates the messages, and the ADU Builder/Reader that constructs the Modbus ADUs. The proposed module was tested in an experimental power plant testbed, in order to evaluate the performance in terms of latency and overhead.

Goldenberg and Wool [103] modeled the Modbus communication traffic using Deterministic Finite Automation (DFA). The proposal is based on the highly periodic traffic pattern. The static communication pattern of SCADA networks enables the modeling of each communication channel as a DFA. The DFA consists of certain states and transition functions. A threshold value is used to detect the presence of unknown transitions between states in the DFA model. The results indicate that most of the unknown transitions were indeed attacks or false alarms.

A set of SNORT rules for both the seiral and TCP/IP versions of Modbus were proposed in [104]. The proposed rules consist of the name, the applicable protocol (TCP or Serial), and the rule text.

Authors in [105] develop a security solution for the Modbus protocol, by deploying security functions in the messaging stack prior to transmission. The Modbus PDU bytes are encrypted using AES [106], RSA [107], or SHA-2 [108] algorithms, while the secret key is exchanged between the master and the slave using a separate secure channel. The efficiency of the proposal is verified by attacking a Modbus testbed using a variety of authentication, integrity, non-repudiation and confidentiality attacks.

Erez and Wool [109] designed an anomaly detection system based on the Modbus protocol registers. An automated process for register classification was developed, based on the following observations: the sensor register values featured a stationary distribution, the counter register values featured monotonic non-decreasing behavior, and the constant register values featured zero variance. The classification algorithm is a single-window decision tree, which evaluates whether the examined traffic matches to one of the aforementioned observations. In the learning phase, different behavioral models were developed for each of the observations, by training the system using legitimate traffic. In the enforcement phase, any deviation from the corresponding behavioral model is considered an anomaly. The evaluation results indicate that the false alert rates are 1.62%, 0%, and 0.88%, for the sensor, counter and constant registers respectively.

In [110], the authors designed an industrial firewall, based on the Modbus TCP protocol, that combines security policies with deep packet inspection methods. The firewall is realized in a Linux platform by using the iptables tool. The industrial control network is divided into different security zones, each zone featuring different security policies. The data are captured and processed in real time, in order to determine whether they comply with the specified policies. An environment that simulates a PLC that drives an electric motor, was used to validate the reliability of the proposed firewall. The experiments indicate that the SYN/ACK flood attacks [111], that were used against the protected network, were successfully intercepted.

Deng *et al.* [112] used Support Vector Machine (SVM) [113] to identify abnormal traffic. The Modbus TCP data are preprocessed to remove unnecessary information so that only the function codes and coils remain in each sequence. In the proposed solution, the frequency of function codes and the number of coils suffice for the classification of the traffic. The experiments were carried out using different SVM kernel function, having accuracy results of 76.05% for the linear kernel function, 89.61% for the polynomial kernel functions and 96.55% for the radial basis kernel function.

Li *et al.* [114] utilized Decision Trees and Neural Networks to classify datasets composed of normal and malicious traffic. The datasets were collected from a simulated factory environment based on Modbus protocol. The test environment was operating for a few days while four kinds of attacks were carried out, namely reconnaissance attacks, command and response injection attacks, and DoS attacks. The generated dataset consists of 64692 instances of which 59842 were normal, while the rest were malicious. The J48 decision tree algorithm was used for classification, having 99, 83% accuracy. Two neural networks with 1 and 2 hidden layers were constructed. The accuracy results were 97.41% and 97.46% respectively.

Yusheng *et al.* [115] proposed an innovative two-part algorithm for intrusion detection. The rule extraction part consists of three modules. The deep protocol parser analyzes both the TCP/IP layers and Modbus application layer, in order to extract the key fields of the packets. The key fields are the IP addresses, ports, sequence numbers, acknowledgment numbers, payload length for the TCP/IP layers and transaction identifiers, protocol identifiers, unit identifiers, function codes and reference numbers for the Modbus application layer. The normal rule set is generated by analyzing the relations within the protocol packet, the relations between the devices, and analyzing the periodicity of the packets. The abnormal rules are generated by extracting and analyzing the features and patterns of the attack behavior (e.g., DoS attacks). The deep inspection part of the proposed algorithm performs real-time deep packet inspection in order to identify which set of rules the inspected packet belongs to. The performance of the proposed algorithm was evaluated in a simulated environment. The results indicate that the proposed algorithm was able to successfully detect all the attacks, namely DoS, MITM, and Relay attacks.

Authors in [116] proposed an intrusion detection method for the Modbus TCP protocol based on honeypots. The Conpot tool was used to simulate a Modbus device in order to capture the traffic sequences. Agglomerative hierarchical clustering [117] was applied to the captured sequences based on a similarity factor. For each cluster, the sequence with the maximum average similarity was selected as the representative sequence of the cluster. These representative sequences were compared with existing attack sequences and based on their similarity, the whole cluster is classified as normal or abnormal. The authors evaluated both the effect of the similarity factor on the identification of five attack types and the accuracy of the proposed solution. The results indicate that the accuracy of the IDS was 92% with 0% false positives.

Dong and Peng [118] proposed an SVM algorithm to classify attacks on a Modbus network. The Wireshark tool [119] was used for capturing and parsing of data packets from a real Modbus device. The captured data was processed into sequences of function codes and register address combinations. The combination of function codes and register addresses is used to calculate the frequency of the sequence of pattern subsequences and then map the frequency to the same dimension eigenvector. The conversion of the combination of different lengths to the same length vector is used to describe the communication features of several packets in the Modbus TCP/IP communication process. The experimental results indicate that the classification accuracy is 94.13%, which shows that the proposed SVM algorithm has a certain advantage in the training of small samples.

The authors in [120] propose an IDS based on the Bro IDS to detect any abnormal behavior of a system that utilizes the Modbus communication protocol. A simulated testbed was utilized in order to evaluate the performance of the Bro IDS implementation. The evaluation results indicate that the proposed IDS implementation successfully detected the attacks that were carried out, namely the MITM and sensor calibration attacks.

C. DNP 3

Table IX summarizes the security proposals that aim to ensure the security of the DNP3 protocol. The table shares the same format with VII. Authors in [121] present the DNPSec framework, which aims to enable confidentiality, authenticity, and integrity in the DNP3 protocol. The main advantage of the proposed framework is that it does not require any modification to the applications or devices, as it only changes the data format of the DNP3 Data Link Layer. DNPSec encrypts the frame and inserts a header, followed by a key sequence number at the start of the DNP3 frame, and an authentication data field at the end. The header is used for addressing and indicating the start of a new session, which requires the slave devices to fetch a new session key from the database. The session keys are generated by the master device and inserted into the database. The key sequence number contains a counter value, which is increased each time the master device sends a message. If the counter reaches the limit, the master terminates the session and starts a new one. The authentication data field is used for the integrity check of the DNP3 frame.

Mander *et al.* [122] implemented a set of security rules for data transmission between DNP3 devices. The proposed security rules focus to the DNP3 function code, object type, qualifier field. If a frame does not comply with those rules it is discarded.

Bai *et al.* [123] proposed an rule-based anomaly detection framework, consisting of two operating modes. In the training mode, the normal rule set is built from the collected data of possible normal behavior. In addition to the TCP/IP headers, the DNP3 payloads are also parsed and analyzed. In the online mode, the traffic is classified based on its deviation from the initial normal rule set. The xMasterSlave simulation software was used to set up the DNP3 testbed, emulating a real environment. A series of attacks were carried in order to evaluate the performance of the framework. The results show that the framework features 0.15% rate for false positives and 0.09% rate for false negatives.

Li *et al.* [124] analyzed the security shortcomings of the DNP3 protocol and proposed a Snort detection rule template for abnormal traffic. The template defines the format of the rule's header which consists of seven parameters and the rule's body consisting of ten parameters. A rule against DoS attacks was generated as a practical example.

Amoah *et al.* [125] developed a security mechanism for the broadcast communication mode of the DNP3 protocol. The existing DNP3 Secure Authentication (DNP3-SA), which is intended for the unicast communication mode, utilizes the challenge-response approach. Nevertheless, this scheme is unsuitable meaning for the broadcast communication mode, as the master station must exchange and store a number of challenges and response messages with each device. This will introduce delays and increase communication overhead, which renders the DNP3-SA impractical for broadcast communication. The proposed scheme utilizes the cryptographic primitives (i.e., AES-128, AES-GMAC, SHA-1-HMAC, and SHA-256-HMAC) specified in existing DNP3-SA, in order to effectively secure DNP3 broadcast communication against injection, relay and spoofing attacks. Finally, the authors evaluated the performance of the proposed scheme in terms of computational and storage overhead.

Nivethan and Papa [126] presented an extension of Linux-based firewalls for securing DNP3-based infrastructures. The proposed scheme uses the iptables tool [127] in order to inspect the payload of a DNP3 message and identify suspicious DNP3 commands. The authors evaluated the firewall by deploying a rogue DNP3 device in a real smart-grid testbed, in order to generate malicious messages. The firewall was able to detect and block all the malicious messages.

Lin *et al.* [128] presented a semantic analysis framework for detecting and mitigating control command injection attacks. The Bro IDS and the DNP3 analyzer were utilized to validate the network packets and detect attacks at the protocol level. Based on the extracted semantics, the effect of the commands are evaluated by the IDS prior to their execution. The authors simulated a small-scale power system and injected malicious control commands in the network. The experimental results indicate 0.78% rate for false positives, 0.01% rate for false negatives and a response latency of about 200ms.

Authors in [129] designed an authentication and encryption protocol for the DNP3 broadcast communication mode. The proposed DNP3 Broadcast Authentication and Encryption (DNP3-BAE) protocol consists of two sub-protocols. The Identity Authentication and Key Agreement, which provides periodic verification of the device's identity and security status, while the Key-update and Broadcast Message Authentication facilitates the key-exchange and encryption of

TABLE VIII
MODBUS SECURITY SOLUTIONS

Reference	Approach	Methodology	Reported Challenges	Evaluation Results
[101]	Attack Detection	Model-based detection, exploiting the static communication pattern of SCADA networks Developed Snort rules for detecting violations Experimental validation in real network	Difficulty in constructing proper detection models Many false alarms in case of inaccurate models	Model-based intrusion detection is a promising approach for monitoring SCADA networks
[102]	Traffic Encryption	Proposal of a secure module that encrypts/decrypts and validates the payloads, before constructing the ADUs Performance evaluation in experimental power plant testbed	Traditional security measures do not address attacks that target the SCADA communication protocols	No evaluation test were performed in terms of accuracy
[103]	Attack Detection	Deterministic Finite Automation-based approach Unsupervised training for the model Validation using real power grid data	The shortcomings of Modbus in terms of security allows attackers to easily inject malicious Modbus messages in the network The proprietary nature and potential sensitivity of SCADA operations complicate the acquisition of real SCADA data	Successful detection of real anomalies Very low false-positive rates
[104]	Attack Detection	Set of Snort rules for both TCP and Serial Modbus	No challenges were reported	No evaluation tests were performed
[105]	Traffic Encryption	Encryption using AES, RSA, SHA-2 algorithms Keys distributed through a secure channel Validation in simulated environment	The Modbus protocol does not incorporate any security features	98% in unicast communication 95% in broadcast communication
[109]	Attack Detection	Anomaly detection based on the deviation from the standard behavior model	The validity of the data values is a critical aspect of the SCADA system security	False rates: 0.86% Overall 1.62% for sensor registers 0% for counter registers 0.88% for constant registers
[110]	Attack Detection	Iptables-based firewall that performs real-time deep packet inspection Validation in simulated environment	Current firewall technologies are mainly focused on the protection of traditional computer networks	The proposed firewall successfully intercepted the performed attacks
[112]	Traffic Classification	Support Vector Machine algorithms Classification based on Modbus function codes and number of coil	The Modbus protocol does not incorporate any security features	76.05% linear kernel function 89.61% polynomial kernel functions 96.55% radial basis kernel function
[114]	Traffic Classification	Decision Tree & Neural Network algorithms Dataset collected from a simulated realistic testbed	The common traffic datasets such as KDD99 are obsolete and unsuitable for SCADA traffic	99.83% decision tree 97.41% 1-layer neural network 97.46% 2-layers neural network
[115]	Attack Detection	2-part IDS algorithm that involves normal and abnormal rule generation based on the datasets and deep real-time packet inspection for detecting attacks Simulated real environment	Pattern matching approaches have high computation load and low accuracy The Modbus protocol has certain vulnerabilities by design	100% detection rate 0.045% false positives
[116]	Traffic Classification	Utilization of a honeypot system to capture attack sequences Clustering similar sequences Extraction of representative sequence for each cluster Classification of each clustered based on the similarity of the representative sequence with existing sequences Simulation of a Modbus smart meter using Conpot	The reported attacks on Modbus often present unexpected patterns, which complicates the accuracy of the IDS	92% detection rate 0% false positives
[118]	Traffic Classification	Support Vector Machine classification algorithm Model based on function codes and register addresses combinations Data captured from a thermal power unit using Wireshark	The relative small amount of abnormal SCADA data imposes imbalance of positive and negative samples to the training process	94.13% detection rate
[120]	Traffic Classification	Implementation of Bro IDS that monitors a simulated testbed	No challenges were reported	The proposed IDS implementation can successfully detect the two attacks that were carried out

the communications, using the existing DNP3-SA encryption primitives. The SPAN tool was used to simulate the protocol function and verify the security of the protocol. The verification results show that the proposed solution can effectively protect sensitive data and accurately authenticate the entities of the network.

TABLE IX
DNP3 SECURITY SOLUTIONS

Reference	Approach	Methodology	Reported Challenges	Evaluation Results
[121]	Traffic Encryption	Modification of the Data Link Layer by adding two fields at the start and one at the end of the frame respectively	Ensurance of the confidentiality, integrity and authenticity with minimal modifications to the system	No evaluation tests were performed
[122]	Traffic Classification	Implementation of security rules to filter out traffic that do not comply to those rules	Protection against malicious traffic, originating from corporate networks	No evaluation tests were performed
[123]	Traffic Classification	Rule based anomaly detection Normal rule-set was built from analyzing TCP/IP headers and DNP3 payloads A realistic environment testbed was simulated using the xMasterSlave tool	TLS and IPSec protection techniques only secure the layers below the application layer, without considering the security of DNP3 application layer	0.15% false positives 0.09% false negatives
[124]	Attack Detection	A rule design template for Snort is proposed An example rule for detecting a potential denial of service attack was presented	DNP3 protocol was initially designed for isolated networks, however with its increasing integration with the Internet certain design shortcomings arise	No evaluation tests were performed
[125]	Traffic Encryption	Proposal of a DNP3-SA for broadcast communication Utilization of the existing DNP3-SA cryptographic primitives Broadcast messages are verified through a hash chain	The existing DNP3 Secure authentication mechanism is limited to unicast communication mode only	Performance evaluation in terms of computational and storage overhead
[126]	Firewall	Extension of iptables to filter common attacks on the DNP3 protocol Evaluation on real smart-grid testbed	The use of TCP/IP for communication exposes the SCADA systems to Internet threats	The approach successfully detected and blocked the malicious messages
[128]	Attack Detection	Bro IDS and DNP3 analyzer for validation of packets and extraction of control commands semantics Power flow analysis of the semantics to evaluate the execution consequences Simulation of a small-scale power system	Control-related attacks modify certain packet fields, which are encoded in a legitimate packet format, making the attack detection challenging	0.78% false positives 0.01% false negatives 200ms latency
[129]	Traffic Encryption	Proposal of an authentication and encryption protocol for the DNP3 broadcast communication mode Utilization of DNP3-SA encryption primitives Simulation and verification of the proposed scheme using the SPAN tool	The existing DNP3 Secure authentication mechanism is limited to unicast communication mode only	The verification results show that the proposed solution can effectively protect sensitive data and accurately authenticate the entities of the network

D. PROFINET

Table X provides a summary of the reviewed security solutions regarding the PROFINET protocol. The table shares the same format with Table VII.

Paul *et al.* [130] performed vulnerability and attack analysis of the PROFINET protocol. The results of the analysis were used for developing an IDS tailored to the security requirements of the PROFINET protocol. The proposed network IDS is based on N-gram anomaly detection and utilizes deep packet inspection in order to identify protocol messages. The resulting protocol messages are split into sequences of n events, called n-grams. Machine learning approaches are used for training and distinguishing between normal network traffic and anomalies.

Authors in [131] modified the Snort packet decoding engine to enable processing of PROFINET real-time data. The experimental results show that the modified Snort can effectively detect intrusions in real-time.

Pfrang and Meier [132] presented two attack techniques that can compromise a PROFINET device. The first attack is based on switch port stealing, while the second exploits the PROFINET's DCP command to perform a reconfiguration attack. The authors proposed an attack detection scheme, by broadcasting alerts in case of modification of the switch and PROFINET device configuration, respectively for each attack. To perform validation of the proposed scheme they utilized real PROFINET components and virtual machines and switches to build the testbed. They considered 14 different attack scenarios, utilizing the aforementioned attacks and different network topologies. The experimental results show that 6 of the 14 attack scenarios were successfully detected.

Authors in [133] proposed an anomaly detection scheme for PROFINET networks. The captured data are processed using the sliding window algorithm to extract a subset of traffic-related features, while an artificial neural network is used to classify the traffic based on those features. The security scheme was applied to three real PROFINET networks of different sizes. The authors performed experiments in order to find the optimal extracted features as well as the optimal number of artificial neural network parameters. The reported overall accuracy of the proposed scheme is over than 90%.

E. Other SCADA Communication Protocols

Table XI provides a summary of the reviewed solutions regarding other SCADA communication protocols. The format of this table is the same as the previous ones, with an

TABLE X
PROFINET Security Solutions

Reference	Approach	Methodology	Reported Challenges	Evaluation Results
[130]	Traffic Classification	Deep packet inspection to identify PROFINET message Messages are split into streams in order to apply machine learning techniques	Firewalls are inadequate in protecting against attacks that are initiated from inside the network SCADA systems require implementation of multi-stage security	No evaluation tests were performed
[131]	Attack Detection	Modification of the Snort decode engine to process PROFINET real-time data	Snort is inadequate for PROFINET real-time data	No evaluation tests were performed
[132]	Attack Detection	Real PROFINET components and virtual machines and switches were utilized to build the testbed	The PROFINET protocol lacks inherent security mechanisms, making it extremely vulnerable to attacks	6 of the 14 attack scenarios were successfully detected
[133]	Traffic Classification	Sliding Windows Algorithm for feature extraction Artificial Neural Networks for traffic classification based on the extracted features Performance was validated in three different sized PROFINET networks	The are limited research works that consider the real-time PROFINET protocol	Over 90%

additional column that denotes the utilized SCADA communication protocol.

The authors in [134] proposed a set of Snort rules in order to detect attacks in substations utilizing the IEC 61850 communication protocol. The network traffic of a series of simulated attacks was captured and analyzed in order to extract the detection rules. The evaluation results show that the proposed system is capable of detecting malicious attacks.

Yang *et al.* [135] proposed a rule-based IDS using a deep packet inspection method that includes signature-based and model-based detection approaches tailored to the IEC 60870 communication protocol. In addition, they implemented the proposed rule-based IDS using Snort. The experimental results indicate that the proposed rule-based IDS can effectively identify malicious traffic.

The authors in [136] present an anomaly-detection model for IEC 61850 through normal-behavior profiling of the exchanged packets. Specifically, the authors used a SVM algorithm to create the normal-behavior models and installed these models in the anomaly detection engine. For the performance evaluation, they carried out experiments using packets collected from a real IEC 61850 substation. The performance results feature accuracy values of 98.98% and 98.56% for MMS and GOOSE messages, respectively.

An IDS for IEC 61850 substations is presented in [137]. The proposed IDS approach provides anomaly-based and parameter-based detection. The main idea of parameter-based detection is to monitor significant operation parameters of the substation. A cyber-physical testbed was developed in order to validate the proposed IDS, while the experimental results were recorded in a log file. The results indicate that the proposed IDS can effectively detect cyber attacks.

Wong *et al.* [138] consider the security of the EtherNet/IP communication protocol, by expanding Suricata's parser in order to decode EtherNet/IP packets. Moreover, they conducted performance evaluations in terms of packet drop rate and CPU usage.

The authors in [139] expanded the Snort tool to process EtherCAT frames. In addition, they developed an initial set of rules in order to evaluate their Snort expansion. However, the authors have not included any evaluation results.

V. SCADA Survivability and Resilience

In the previous section, a number of SCADA defense mechanisms against cyber threats was presented and discussed. The reported results highlight the high performance level of these mechanisms. Nevertheless, there are cases where a defense mechanism cannot detect and mitigate every threat. In such cases, attacks that reduce the availability of the system, such as DoS and virus attacks, can disable critical components of the infrastructure. In addition, the aforementioned defense mechanisms cannot protect the infrastructure from physical threats such as natural disasters, or physical attacks.

The network topology has a crucial impact on the survivability and resilience of the SCADA system. The SCADA communication protocols, that were presented in Section II utilize a number of topologies, each one offering different advantages and disadvantages, that enable the connection between the controller and the field devices. The bus, tree, and daisy-chain topologies are the simplest and the most cost-effective, as they connect the devices in series. However, in case of a failing node due to an attack or a wiring fault, only the devices deployed up to this point will have a connection with the control center. The point-to-point and star topologies offer more resilience against failures, compared to the bus, line, and daisy-chain. The peer-to-peer and ring offer the most resilience in case of failures, as there are multiple paths between the controller and the devices. Finally, the more recent and advanced SCADA communication protocols (such as the ones that are based on Ethernet technologies) implement acknowledgment mechanisms that guarantee the correct exchange of data.

To this end, the survivability concept was developed, which requires a certain functions of a SCADA system are operational even if parts of the infrastructure are compromised or destroyed. This section provides a review of proposals that

TABLE XI
OTHER SCADA PROTOCOLS SECURITY SOLUTIONS

Reference	Protocol	Approach	Methodology	Reported Challenges	Evaluation Results
[134]	IEC 61850	Attack Detection	Development of Snort rules for IEC 61850	Attack detection in IEC 61850 systems is challenging, due to the limited processing capabilities of the devices	The proposed system is capable of detecting the malicious attacks
[135]	IEC 60870	Attack Detection	Development of a Snort-based IDS that uses deep packet inspection	IEC 60870 transmits messages in clear text without any encryption mechanism.	The proposed system is capable of detecting the malicious traffic
[136]	IEC 61850	Traffic Classification	Utilized SVM to create behavior models of a substation	Signature-based and rule-based anomaly detection approaches cannot detect novel attacks	98.98% accuracy for MMS messages 98.56% accuracy for GOOSE messages
[137]	IEC 61850	Attack Detection	Proposed an IDS that utilizes anomaly-based and parameter-based detection approaches	No challenges were reported	The results indicate that the proposed IDS can effectively detect cyber attacks.
[138]	EtherNet/IP	Attack Detection	Expansion of Suricata in order to process EtherNet/IP packets	Surricata cannot analyze EtherNet/IP packets	Less than 2% drop rate 20–120% CPU usage
[139]	EtherCAT	Attack Detection	Expansion of Snort in order to process EtherCAT frames	Snort stops the analysis after decoding the Ethernet frame There are no Snort rules for the EtherCAT protocol	No evaluation results were presented

aim to evaluate and/or ensure the survivability and resilience of SCADA systems.

Authors in [140] present a model for survivability of Smart Grid under vulnerabilities and severe emergencies. They utilize concepts from graph theory to analyze the vulnerabilities and their impact on the performance of the network. Using the presented model, they examine the survivability of an IEEE-118 bus system under random and targeted cyber attacks.

Queiroz *et al.* [141] propose a probabilistic model that predicts the survivability of SCADA systems The proposed model utilizes network traffic to create a Bayesian network based on the data exchanged among services. The performance of the proposed model was evaluated through a demonstration scenario involving a SCADA network under cyber attacks.

Authors in [142] propose an extensible and flexible framework for SCADA survivability, based on interdependency modeling. The framework aims at vulnerability reduction by analyzing both structural and functional vulnerabilities. In addition, they model an IEEE-30 bus of Smart Grid and SCADA networks under random node failures and physical attacks targeting many nodes. Using this model, the level of robustness is investigated by measuring the functionality of the system as a function of the node failures.

Kirsch *et al.* [143] present a robust SCADA system capable of surviving a partial compromise based on intrusion-tolerant state machine replication. They also discuss SCADA systems survivability requirements and provide an overview of novel techniques that integrate intrusion-tolerant replication mechanism to SCADA systems. Finally, they evaluated the system performance in terms of latency, accuracy, and scalability.

In [144], the authors present a survivable intrusion-tolerant replication model, that ensures the reliability across diverse system components and the resilience of the system over its lifetime. The model was evaluated through experiments involving both physical and virtualized environments.

Lopez *et al.* [145] introduce a MultiPath TCP scheme for SCADA systems based on the Modbus protocol. By utilizing multiple subflows over the network interfaces, SCADA systems are protected against network failures. A series of simulations involving scenarios of link failures and DDoS attacks was carried out in order to evaluate the performance of the proposed scheme.

Authors in [146] present a resilient architecture for critical infrastructures based on autonomic computing and Moving Target Defense techniques. Within the Moving Target Defense concept, the communication infrastructure is constantly shifting and changing in order to increase complexity for the attackers, limit the exposure of vulnerabilities, and enhance the resilience of the system. A smart grid testdbed was developed in order to evaluate its resilience against cyber attacks, such as flooding, DDoS, and jamming attacks.

Babay *et al.* [147] developed a novel architecture called Spire, which distributes replicas of the SCADA control center across multiple locations in order to enhance the resilience against cyber attacks. In order to evaluate the efficiency of the proposed solution, the authors deployed Spire in a wide area consisting of two control centers and two data centers.

The authors in [148] proposed a framework for modeling and assessing the resilience of critical infrastructure. Specifically, they built a Bayesian network model in order to asses the risk associated with the disruption of complex electrical networks. In addition, a real case study was selected to validate the proposed framework.

A comprehensive solution for ensuring the SCADA survivability is presented in [149]. The proposed solution is based on virtualization technologies in order to build a resilient communication infrastructure. The experimental results demonstrate the effectiveness of the proposed solution.

The authors in [150] propose a robust extension of the Multipath-TCP protocol. The proposed extension uses a novel stream hopping mechanism that hides open port numbers by periodically renewing the sub-flows. The experimental results indicate that the proposed protocol can effectively mitigate DoS attacks with low communication overhead.

Rehak *et al.* [151] present a CIERA methodology designed for Critical Infrastructure Elements Resilience Assessment. The method primarily relies on the complex assessment of the robustness, recoverability, and adaptability of elements in technically oriented sectors with respect to disruptive events of naturogenic, technogenic and anthropogenic origin. It takes into account the functional, structural and performance parameters of the elements being assessed while facilitating the identification of the element's weak points.

This section discussed the impact of network topology in the survivability and resilience of SCADA systems and reviewed research works that aim to evaluate and/or enhance the survivability and resilience. The assessment of a systems' survivability and resilience is the first step towards enhancing them. To this end, multiple assessment schemes were proposed that are able to model the system survivability and resilience as well as its behavior against potential cyber and physical threats. Regarding the survivability and resilience enhancement, many research works replicate various critical components of the communications infrastructure, while others provide redundant data flows and continuous infrastructure shifting.

VI. SCADA Trends & Advancements

This section provides the trends and advancements in the SCADA systems, compelled by the ever-growing requirements of the industrial applications, and empowered the advancements in processing, networking, and storage resources.

A. Novel SCADA Communication Protocols

The emerging Industry 4.0 is changing the way industrial and automation applications operate [152]. The protocols mentioned in this section are based on a master-slave configuration. However, the amount and type of data that is exchanged between industrial and automation components render this configuration inadequate. Consequently, novel distributed protocols are being designed in order to satisfy the increasing application requirements.

Authors in [153] developed a framework for designing distributed communication protocols, that can satisfy the strict real-time requirements of automation applications. The architecture consists of two layers: the interface layer, which provides operation in time-slots and the coordination layer that assigns a device to each time slot. They proposed an Ethernet implementation, but the framework can be applied to other shared-medium environments such as WIFI and WirelessHART.

Skodzik *et al.* [154] proposed HaRTKad, which is a Peer-to-Peer approach based on the Kad network. The Kad network is variant of the Kademlia [155] decentralized P2P protocol. Kad was extended by the Time Division Multiple Access mechanism in order to support strict real-time applications. The proposed prototype enables the realization of time constraint applications, ensuring high reliability, flexibility, and scalability.

Sági and Varga [156] presented the architecture of a distributed SCADA system that enables efficient real-time monitoring and control procedures in industrial environments. It relies on a distributed real-time database storage system that facilitates data distribution with configurable bandwidth based on the application demands.

B. Internet of Things

The Internet of Things (IoT) is an emerging concept driven by the advancements in the wireless communication technologies [157]. An IoT system is a collection of collaborating smart devices utilized in various consumer applications. The Industrial Internet of Things (IIoT) [158] is considered an evolution of SCADA systems that focuses on industrial applications such as power generation and distribution, transportation, and manufacturing. For example, a critical application of IIoT is the predictive maintenance of industrial equipment. Predictive maintenance can lead to decreased downtime, reduced maintenance costs, and increased productivity. A taxonomy of IoT protocols, schemes and mechanisms are presented in [159], while in [160] the authors provide a review of IoT protocols that are applicable to the Smart Grid concept.

The inherent security challenges of IoT communications along with the criticality of industrial applications urges researchers to devise novel security schemes for addressing the industrial security requirements [161]. An analytic framework for modeling cyber attacks against IoT infrastructures is introduced in [162]. The authors in [163] survey the state of IoT security and discuss the challenges, countermeasures and future directions.

C. Virtualization Techniques

Virtualization is another popular concept, which enables the abstraction and sharing of physical resources among different parties. Virtualization effectively reduces the overall cost of equipment, facilitates its configuration and provides flexibility and scalability. Network Function Virtualization (NFV) [164] is an emerging paradigm, which offers new ways of designing, deploying, and managing network services. The abstraction between virtual and physical devices, that is enabled by virtualization technologies, extends the life of older software and hardware [165]. Therefore, virtualization is a compelling concept for modernizing obsolete SCADA systems that cannot be replaced due to operational reasons or high costs. Furthermore, virtualization can be used to design and develop SCADA security testbeds in a cost-effective manner.

Cruz *et al.* [166] proposed a framework for building scalable SCADA testbeds based on virtualization technologies. Moreover, a case study demonstrating security attacks is presented.

Authors in [167], proposed a novel and modular approach for virtualizing replicates of complex SCADA systems. The SCADA system is segmented into smaller components, which are virtualized independently. This approach reduces the size and cost of SCADA testbeds, facilitating the cybersecurity research.

Cahn *et al.* [168] designed and deployed a SCADA network architecture which provides a reliable, secure, auto-configured network through the use of SDN technologies.

D. SDN Visibility

Software Defined Networking (SDN) [169] is an emerging concept that separates the control and data plane, and simplifies the programmability of the network. The SDN controller has a global view of the network state, which enables the development of countermeasures against security threats with very low impact on the communication requirements. SDN provides high-level network abstraction and an Application Programming Interfaces (API) for monitoring and managing the communication infrastructure. The network programmability enables the development of network security applications that effectively monitor the network in order to detect malicious traffic.

By leveraging the advantages of SDN, the SDN visibility security measure can be deployed to protect critical infrastructures. Using SDN visibility, the network configuration (such as the IPs of SCADA field devices or HMIs) can be reconfigured in the presence of cyber attacks, without disrupting the operation of the SCADA system. A review of security works utilizing the SDN visibility is presented below:

Mehdi *et al.* [170] showed the feasibility of utilizing SDN in order to accurately detect malicious activity inside the network. For evaluation purposes, they implemented the Threshold Random Walk with Credit Based Limiting [171], Rate-Limiting [172], Maximum Entropy Detector [173], and NETAD [174] algorithms in the OpenFlow [175] controller.

Xing *et al.* [176] presented SnortFlow, which is an intrusion detection system based on OpenFlow. The system leverages Snort's detection capabilities and the network reconfiguration features of OpenFlow.

Authors in [177] proposed a source address validation mechanism based on SDN. A protective perimeter is formed by a number of OpenFlow devices. Any packet, that originates outside the perimeter, is forwarded to the controller. The source of the packet is validated based on a set of generated rules.

Giotis *et al.* [178] proposed a method based on OpenFlow and sFlow, that can effectively detect and mitigate traffic anomalies. The packet sampling capabilities of sFlow are combined with an entropy-based detection algorithm.

Authors in [179] proposed a system for Distributed and Collaborative per-flow Monitoring (DCM). A monitoring tool is installed in SDN-enabled switches and forwards the flow information of a new packet to the controller. The controller uses Bloom filters to decide how the packet should be handled.

Authors in [180] designed an eavesdropping countermeasure for securing communication flows between SCADA components. Using SDN, the communication routes between SCADA devices are modified in certain intervals.

The authors in [181] present a deep learning anomaly detection approach in SDN environments. The OpenFlow switches forward their network statistics to a centralized controller. The controller sends the statistics to the intrusion detection module for analysis. A Deep Neural Network is used to analyze and detect flow anomalies.

A novel SDN-enabled security architecture for the smart grid is proposed in [182]. It is based on a specialized SDN controller that forwards AES-128 encrypted metering data.

Machii *et al.* [183] extended the IEC62443's Zones and Conduits security measure, by proposing a dynamic zoning methodology based on SDN.

The authors in [184] leverage the SDN concept to virtualize a data diode. Data diodes provide a physical mechanism for enforcing strict unidirectional between two networks [185]. They are often built using fiber optic transceivers by removing the transmitting and receiving components from each side, respectively. Therefore, it is physically impossible to compromise these devices and intercept network traffic.

E. Big Data Analytics

The interconnection of SCADA systems using high speed wired or wireless networks allows the exchange of large amounts of data in a very short time. This enables the leverage of Big Data analytics [186], [187], which can effectively assist in the detection and mitigation of cyber-attacks. There are existing proposals that leverage the Big Data analytics in order to secure critical infrastructures.

Authors in [188] developed a real-time IDS that performs traffic classification using Big Data analytics. In order to train the detection model they extracted certain features from DARPA [189], KDD99 [190], and NSL-KDD [191] datasets. The Apache Spark machine learning library [192] was used to perform classification using algorithms such as Naive Bayes, Support Vector Machine, Random Forests, and REPTree.

Vimalkumar and Radhika [193] also used the Apache Spark to design solution for detecting cyber attacks. They built a custom dataset that consists of data from Phasor Measurement Units. For traffic classification, they used the Deep Neural Network, Support Vector Machine, Naive Bayes, and Random Forest algorithms.

Finally, Natesan *et al.* [194] proposed an IDS that is based on the Apache Hadoop framework [195]. They used the KDD99 dataset and the Naive Bayes algorithm for the classification of the traffic.

F. SCADA Cyber Hygiene

The severe increase in the frequency of cyber attacks against critical infrastructures has raised concerns about security at every level of an organization or company. The organizations and companies must be better prepared to respond and recover from novel cyber-attacks, as adversaries are constantly developing and experimenting with different types of malware. An exemplary countermeasure is presented in [196]. The authors developed an anonymous incident communication channel, that enables Smart Grid operators to cooperatively exchange cyber attack details and patterns. These new variants of malware allow hackers to launch multiple types of attacks against individuals and organizations. Furthermore, certain security incidents were reported (e.g., [48], [50]) where the cyber attacks against a company originated from the compromise of another one.

In order to effectively secure critical infrastructures, efficient cyber hygiene strategies should be adopted. Cyber hygiene involves establishing certain routine measures in order to minimize the risks from cyber attacks. The adoption of good cyber hygiene practices reduces the risk that a vulnerable organization will be exploited in order to launch attacks and compromise related organizations.

The insider threat is a significant security concern for organizations managing critical infrastructures [197]. This highlights implications regarding the high awareness levels among employees about the insider threat. Therefore, the organizations should provide proper training regarding the behavioral indicators of insider threats and confidential reporting processes.

The National Institute of Standards and Technology (NIST) [198] has released a series of guidelines for the Smart Grid, which can be generalized for SCADA systems. To begin with, the cybersecurity countermeasures should be deployed at multiple locations to resist many attack approaches. Such measures are the enforcement of security policies within the organization and the employment of technical tools that implement the security mechanisms and services. Secondly, all security approaches suffer from inherent vulnerabilities. By deploying layered defenses these vulnerabilities can be diminished. Additionally, the trust relationships between systems and organizations have to be evaluated, established, and maintained. Moreover, roles and responsibilities have to be specified for the trusted partners. The use of cryptographic mechanisms, such as security keys and certificates, should be enforced. Moreover, intrusion detection systems should be deployed. Those systems are responsible for detecting, analyzing, responding and reporting any intrusions and anomalous events in a very short time. Finally, it should be mandatory that all the staff attends to a comprehensive program that includes training, practical experience and awareness. Also, the system administrators should be certified by recognized authorities.

The European Network and Information Security Agency (ENISA) published a list of seven recommendations that focus on improving the security of SCADA systems [199]. Those recommendations include 1) the creation of Pan-European and National SCADA security strategies, that will serve as references for stakeholders, 2) the creation of a good practices guide for SCADA security, 3) the creation of security plan templates, that will guide the operators in classifying their systems and prioritizing the most critical ones, 4)the fostering of security risk awareness and training, 5) the creation of a common security framework, that will help stakeholders to detect potential threats and evaluate security countermeasures in a controlled and isolated environment, 6) the creation of emergency response team of security experts, that will provide the necessary services to handle and recover from security violations, and 7) the fostering of security research by leveraging existing research programmes.

G. Lessons Learned, Open Research Problems and Challenges

This subsection summarizes the lessons learned that derive from the review, analysis, and discussion of the security concerns regarding SCADA systems.

The architecture of a SCADA system consists of multiple components, such as HMIs, MTUs, RTUs, and field devices. The communications among these components is enabled by industrial communication protocols. The legacy protocols have low requirements in terms of throughput and bandwidth, while the monitoring and control operations take place locally. The newer protocols have increased throughput and bandwidth requirements, and they are connected to the Internet in order to enable remote monitoring and control.

SCADA systems monitor and control the process of critical infrastructures such as power telecommunications, transportation, and manufacturing plants. It is apparent that cyber threats against critical SCADA systems are on the rise. There are reports on numerous incidents worldwide against SCADA systems, damaging the infrastructure and threatening public health.

The underlying communication protocols of SCADA systems are threatened from several cyber attacks that aim to violate the availability, confidentiality, authorization, and integrity of the system. To this end, the security aspect of SCADA systems is receiving significant attention. Multiple research works aim to design and develop SCADA security testbeds (physical or and protection mechanisms. Modbus and DNP3 are the most widely used communication protocols, however, they have no (Modbus) or weak (DNP3) security mechanisms. Consequently, there are many research works aiming to secure these protocols. Common protection approaches utilized throughout the reviewed works include attack detection schemes, traffic encryption algorithms, traffic classification techniques, and leverage of firewall tools.

Most of the reviewed solutions regarding the Modbus protocol, attempt to classify network traffic using Neural Networks, Decision Trees, and Support Vector Machines. The attack detection techniques are mostly based on a pre-configured set of detection rules or models. The AES, RSA, and SHA-2 are the most prominent encryption algorithms. Finally, all the proposal feature high detection rates and very low false positives. Regarding the DNP3 protocol, most of the proposals are based on attack detection approaches, leveraging rule-based and anomaly detection techniques. Traffic encryption solutions use the well-known AES, RSA and SHA-2 algorithms, which are already included in the protocol. Similar to the Modbus protocol, the proposed solutions feature high detection rate and low false positives.

Finally, the PROFINET solutions are limited and there are not many details regarding their implementation and accuracy results. As the protocol does not have publicly available specifications, the proposals aim to provide security by using

TABLE XII
APPROACHES TO SCADA SECURITY CHALLENGES

Challenge	**Approach**
Lack of mature security tools for SCADA systems	Effective security tools for SCADA systems have emerged over the last years
Ensuring security for huge number of devices, as well as protection against persistent adversaries	Leverage of Big Data analytics
Legacy devices and protocols introduce security vulnerabilities	Adoption of appropriate cyber hygiene strategies
Inherited vulnerabilities from standard computer systems	Enhance the security of the SCADA system by leveraging technologies such as NFV and SDN.

schemes that do not modify the protocol's specifications. Most of the proposals use rule-based attack detection technique, while there is one proposal that uses Artificial Neural Networks for traffic classification.

The attack detection techniques that are based on a pre-configured set of detection rules or models can be implemented in hardware with limited processing resources and can achieve detection with very low latency. However, these techniques are unable to detect previously unknown attacks and may result in high false positive rates. To this end, techniques such as Neural Networks and Support Vector Machines were proposed, as they can effectively detect novel attacks. However, they require hardware with more processing resources depending on the system that they monitor.

Nevertheless, the proposed security mechanisms cannot detect and mitigate all cyber threats. Moreover, they are incapable of protecting the SCADA system against physical attacks. Consequently, the SCADA survivability and resilience concept has emerged, that provides methods for risk assessment and model the behavior of the system in the presence of threats.

The communication requirements of SCADA systems are increasing, as new and advanced applications are emerging. In order to address the ever-increasing requirements, new SCADA communication protocols are being developed. The emerging Software Defined Networking concept and virtualization technologies enable the development of novel cyber security mechanisms by leveraging the network's flexibility and programmability. The advances in the computing and storage capabilities enable the use of Big Data analytics in large traffic datasets.

In light of the aforementioned remarks, the SCADA security challenges that were presented in Section III-C have been addressed as follows (also summarized in Table XII).

The lack of mature security tools for SCADA systems is one of the main security challenges. To this end, numerous research works have emerged over the recent years, that propose effective security tools for SCADA systems. Particularly, in this work, we have reviewed a) eight research works that aim to detect attacks against SCADA systems using various communication protocols, b) thirteen works that aim to secure SCADA systems utilizing the Modbus communication protocol, c) eight works focusing on the security of SCADA systems using the DNP3 protocol, d) four works focusing on the Profinet protocol, and e) six research works that aim to secure the IEC 61850 and 60870, EtherCAT, and EtherNet/IP protocols.

Ensuring the security for huge number of devices is challenging. In addition, these devices are attractive targets for adaptive persistent adversaries. The ever-increasing computing, networking, and storage capabilities enable the utilization of Big Data analytics. There are several research works (e.g., [188], [193], [194]) that leverage Big Data analytics in order to create novel anomaly detection approaches. These approaches can monitor and analyze the network traffic from a huge number of devices, and effectively detect potential anomalies. Furthermore, as they do not rely on signature or rule detection, they can detect novel attacks.

The use of legacy devices and protocols introduces several security vulnerabilities to the SCADA system. However, the replacement of these devices and protocols is not always possible, often due to the high equipment cost or the incompatibility of other protocols and/or equipment. These security issues can be mitigated by adopting appropriate cyber hygiene strategies, that can effectively reduce the risk, as well as the impact of a potential compromise.

As the technologies of standard computer systems are being adopted in SCADA systems, their vulnerabilities are also inherited. Nevertheless, these novel technologies can enable the development of novel and efficient countermeasures. The NFV and SDN concepts can be leveraged to develop secure, resilient, and auto-configured SCADA networks. For example, a SCADA network will change its configuration in case of an attack, without disrupting the operation of the monitoring and control processes.

VII. CONCLUSION

SCADA systems are crucial to industrial applications such as power generation and distribution, telecommunications infrastructures, transportation, and manufacturing industries. As SCADA systems are being interconnected to the Internet, they are exposed to security threats that can disrupt their normal operation. As a consequence, researchers are focusing on increasing the security and reliable operation of those systems.

In this work, we described the general SCADA architecture and provided a detailed overview of the well-known SCADA communication protocols. Certain SCADA security incidents were reported, in order to project the paramount importance of SCADA security violations and their impact on public health and safety. Afterwards, we discussed the security objectives, the threats, and the attacks that affect the SCADA systems. It can be observed that SCADA systems have the same security objectives and are affected by the same threats and attacks as the common computer systems.

Moreover, we performed a thorough review of SCADA security proposals and discussed the state of security. Most of the security proposals follow similar approaches, consisting of model or rule based attack detection, classifying traffic using SVM, Neural Networks, and Decision Trees, and traffic encryption. The overall evaluation results claim high accuracy and very low false positives.

Lastly, we presented the SCADA trends and future advancements. These include the design of novel SCADA protocols in order to address the requirements of the Industry 4.0 applications. Additionally, the use of virtualization technologies can further reduce the deployment cost, facilitate the configuration and maintenance, and provide high scalability and reliability. Furthermore, the advancements in communication and processing technologies enable the incorporation of Big Data analytics, as a measure against cyber-attacks. Finally, the adoption of good cyber hygiene strategies is crucial to efficiently securing critical infrastructures.

References

[1] K. Stouffer, J. Falco, and K. Scarfone, *Guide to Industrial Control Systems (ICS) Security*, document NIST SP-800, NIST, Gaithersburg, MA, USA, 2011.

[2] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in SCADA networks," *Comput. Security*, vol. 25, no. 7, pp. 498–506, 2006.

[3] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "SCADA security in the light of cyber-warfare," *Comput. Security*, vol. 31, no. 4, pp. 418–436, 2012.

[4] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 5–20, 1st Quart., 2012.

[5] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Comput. Netw.*, vol. 57, no. 5, pp. 1344–1371, 2013.

[6] R. Leszczyna and E. Egozcue, "ENISA study: Challenges in securing industrial control systems," in *Securing Critical Infrastructures and Critical Control Systems: Approaches for Threat Protection*. Hershey, PA, USA: IGI Global, 2013, pp. 105–143.

[7] R. Leszczyna, E. Egozcue, L. Tarrafeta, V. F. Villar, R. Estremera, and J. Alonso, "Protecting industrial control systems-recommendations for Europe and member states," Eur. Union Agency Netw. Inf. Security, Heraklion, Greece, Rep., 2011. [Online]. Available: https://www.enisa.europa.eu/publications/protecting-industrial-control-systems.-recommendations-for-europe-and-member-states/at_download/fullReport

[8] J. Gao *et al.*, "SCADA communication and security issues," *Security Commun. Netw.*, vol. 7, no. 1, pp. 175–194, 2014.

[9] S. McLaughlin *et al.*, "The cybersecurity landscape in industrial control systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1039–1057, May 2016.

[10] S. Nazir, S. Patel, and D. Patel, "Assessing and augmenting SCADA cyber security: A survey of techniques," *Comput. Security*, vol. 70, pp. 436–454, Sep. 2017.

[11] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Design Test*, vol. 34, no. 4, pp. 7–17, Aug. 2017.

[12] S. Ghosh and S. Sampalli, "A survey of security in SCADA networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019.

[13] I. Verhappen and A. Pereira, *Foundation Fieldbus*. Research Triangle Park, NC, USA: ISA, 2008.

[14] K. Bender, *PROFIBUS: The Fieldbus for Industrial Automation*. New York, NY, USA: Prentice-Hall, 1993.

[15] J. Azevedo, N. Cravoisy, and N. World. *The WorldFIP Protocol*. Accessed: Apr. 27, 2020. [Online]. Available: http://www.dei.isep.ipp.pt/∼emt/infind/fiptut.pdf

[16] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 116–126, Sep. 1997.

[17] *IEEE Standard for Electric Power Systems Communications—Distributed Network Protocol (DNP3)*, IEEE Power Energy Soc., Piscataway, NJ, USA, Oct. 2012.

[18] *EtherCAT Slave Controller*, BECKHOFF New Autom. Technol., Verl, Germany, Oct. 2014.

[19] S. J. Vincent, *Foundation Fieldbus High Speed Ethernet Control System*, Fieldbus Inc., Austin, TX, USA, 2001.

[20] A. Robertinit, "PROFINET: The future of the Ethernet-based automation," in *Proc. IPLNET Workshop*, Sep. 2003, pp. 9–10.

[21] *SERCOS III: Proposal for a Publicly Available Specification for Real-Time Ethernet*, document IEC 65C/341/NP, Int. Electrotech. Commission, Geneva, Switzerland, 2004.

[22] M. Wlas, M. Gackowski, and W. Kolbusz, "The Ethernet powerlink protocol for smart grids elements integration," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2011, pp. 2070–2075.

[23] S. H. Abbas and S. H. Hong, "A top-down approach to add hot-pluggable asynchronous devices to RAPIENET infrastructure," in *Proc. 9th Int. Symp. Commun. Inf. Technol.*, 2009, pp. 128–133.

[24] *Modbus Application Protocol Specification*, Schneider Autom. Inc., Bengaluru, India, Apr. 2012.

[25] *Unitronics, Communication With the Vision PLC*. Accessed: Apr. 27, 2020. [Online]. Available: https://unitronicsplc.com/Download/SoftwareUtilities/Unitronics PCOM Protocol.pdf

[26] L. Rosa, M. Freitas, S. Mazo, E. Monteiro, T. Cruz, and P. Simões, "A comprehensive security analysis of a SCADA protocol: From OSINT to mitigation," *IEEE Access*, vol. 7, pp. 42156–42168, 2019.

[27] V. Schiffer, D. Vangompel, and R. Voss, *The Common Industrial Protocol (CIP) and the Family of CIP Networks*, ODVA, Milwaukee, WI, USA, 2006.

[28] K. Etschberger, R. Hofmann, J. Stolberg, C. Schlegel, and S. Weiher, *Controller Area Network: Basics, Protocols, Chips and Applications*. IXXAT Autom., Ravensburg, Germany, 2001.

[29] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proc. ACM 1st Annu. Conf. Res. Inf. Technol.*, 2012, pp. 51–56.

[30] S. A. Baker, S. Waterman, and G. Ivanov, *In the Crossfire: Critical Infrastructure in the Age of Cyber War*. McAfee Inc., Santa Clara, CA, USA, 2009.

[31] *RISI—The Repository of Industrial Security Incidents*. Accessed: Apr. 27, 2020. [Online]. Available: http://www.risidata.com/

[32] T. Smith. (Oct. 2001). *Hacker Jailed for Revenge Sewage Attacks*. [Online]. Available: https://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/

[33] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security Privacy*, vol. 99, no. 4, pp. 33–39, Jul./Aug. 2003.

[34] K. Poulsen. (Aug. 2003). *Slammer Worm Crashed Ohio Nuke Plant Net*. [Online]. Available: https://www.theregister.co.uk/2003/08/20/slammer_worm_crashed_ohio_nuke/

[35] E. Levy, "The making of a spam zombie army. Dissecting the Sobig worms," *IEEE Security Privacy*, vol. 99, no. 4, pp. 58–59, Jul./Aug. 2003.

[36] (Aug. 2003). *Computer Virus Brings Down Train Signals*. [Online]. Available: https://www.informationweek.com/computer-virus-brings-down-train-signals/d/d-id/1020446

[37] D. Goodin. (Nov. 2007). *Electrical Supe Charged With Damaging California Canal System*. [Online]. Available: https://www.theregister.co.uk/2007/11/30/canal_system_hack/

[38] S. Gorman. (Apr. 2009). *Electricity Grid in U.S. Penetrated by Spies*. [Online]. Available: https://www.wsj.com/articles/SB123914805204099085

[39] C. Glenn, D. Sterbentz, and A. Wright, "Cyber threat and vulnerability analysis of the U.S. electric sector," Idaho Nat. Lab., Idaho Falls, ID, USA, Rep., 2016. [Online]. Available: https://www.osti.gov/servlets/purl/1337873

[40] D. Goodin. (Jul. 2009). *FEDS: Hospital Hacker's 'Massive' DDoS Averted*. [Online]. Available: https://www.theregister.co.uk/2009/07/01/hospital_hacker_arrested/

[41] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.

[42] G. Keizer. (Sep. 2010). *Is Stuxnet the 'Best' Malware Ever?* [Online]. Available: https://www.computerworld.com/article/2515757/malware-vulnerabilities/is-stuxnet-the-best-malware-ever.html

[43] Y. Melman. *Computer Virus in IRAN Actually Targeted Larger Nuclear Facility*. Accessed: Apr. 27, 2020. [Online]. Available: https://www.haaretz.com/1.5118389

[44] D. Kushner. (Mar. 2014). *StuxNet Attackers Used 4 Windows Zero-Day Exploits*. [Online]. Available: https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/

[45] J. Leyden. (Feb. 2011). *Chinese Cyberspies Target Energy Giants*. [Online]. Available: https://www.theregister.co.uk/2011/02/10/night_dragon_cyberespionage/

[46] K. Zetter. (Oct. 2011). *Son of Stuxnet Found in the Wild on Systems in Europe*. [Online]. Available: https://www.wired.com/2011/10/son-of-stuxnet-in-the-wild/

[47] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, "Duqu: A Stuxnet-like malware found in the wild," CrySyS Lab., Budapest, Hungary, Rep. 14, 2011.

[48] D. Starkey. (Sep. 2017). *Hacker Group Dragonfly Takes Aim at U.S. Power Grid*. [Online]. Available: https://www.geek.com/tech/hacker-group-dragonfly-takes-aim-at-us-power-grid-1715157/

[49] A. Greenberg. (Jun. 2017). *How an Entire Nation Became Russia's Test Lab for Cyberwar*. [Online]. Available: https://www.wired.com/story/russian-hackers-attack-ukraine/

[50] M. Kumar. (Sep. 2017). *Dragonfly 2.0: Hacking Group Infiltrated European and U.S. Power Facilities*. [Online]. Available: https://thehackernews.com/2017/09/dragonfly-energy-hacking.html

[51] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proc. IEEE*, vol. 93, no. 6, pp. 1152–1177, Jun. 2005.

[52] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for SCADA systems," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1836–1846, Nov. 2008.

[53] Q. Wanying, W. Weimin, Z. Surong, and Z. Yan, "The study of security issues for the industrial control systems communication protocols," in *Proc. Joint Int. Mech. Electron. Inf. Technol. Conf.*, 2015, pp. 693–698.

[54] K. Coffey *et al.*, "Vulnerability assessment of cyber security for SCADA systems," in *Guide to Vulnerability Analysis for Computer Networks and Systems*. Cham, Switzerland: Springer, 2018, pp. 59–80.

[55] Q. S. Qassim, N. Jamil, M. Daud, A. Patel, and N. Ja'affar, "A review of security assessment methodologies in industrial control systems," *Inf. Comput. Security*, vol. 27, no. 1, pp. 47–61, 2019.

[56] F. Wang, X.-Z. Zheng, N. Li, and X. Shen, "Systemic vulnerability assessment of urban water distribution networks considering failure scenario uncertainty," *Int. J. Crit. Infrastruct. Protect.*, vol. 26, Sep. 2019, Art. no. 100299.

[57] E. Irmak and İ. Erkek, "An overview of cyber-attack vectors on SCADA systems," in *Proc. IEEE 6th Int. Symp. Digit. Forensic Security (ISDFS)*, 2018, pp. 1–5.

[58] N. R. Rodofile, K. Radke, and E. Foo, "Extending the cyber-attack landscape for SCADA-based critical infrastructure," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 14–35, Jun. 2019.

[59] R. Leszczyna, "A systematic approach to cybersecurity management," in *Cybersecurity in the Electricity Sector*. Cham, Switzerland: Springer, 2019, pp. 87–125.

[60] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack taxonomies for the Modbus protocols," *Int. J. Crit. Infrastruct. Protect.*, vol. 1, pp. 37–44, Dec. 2008.

[61] Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of attacks on industrial control protocols," in *Proc. IEEE Int. Conf. Protocol Eng. (ICPE) Int. Conf. New Technol. Distrib. Syst. (NTDS)*, 2015, pp. 1–6.

[62] S. East, J. Butts, M. Papa, and S. Shenoi, "A taxonomy of attacks on the DNP3 protocol," in *Proc. Int. Conf. Crit. Infrastruct. Protect.*, 2009, pp. 67–81.

[63] R. Leszczyna, "Cybersecurity assessment," in *Cybersecurity in the Electricity Sector*. Cham, Switzerland: Springer, 2019, pp. 149–179.

[64] K. Barnes and B. Johnson, "National SCADA test bed substation automation evaluation report," Idaho Nat. Lab., Idaho Falls, ID, USA, Rep. INL/EXT-09-15321, 2009.

[65] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye, and D. Nicol, "SCADA cyber security testbed development," in *Proc. IEEE 38th North Amer. Power Symp. (NAPS)*, 2006, pp. 483–488.

[66] *PowerWorld—The Visual Approach to Electric Power Systems*. Accessed: Apr. 27, 2020. [Online]. Available: http://www.powerworld.com/

[67] M. Liljenstam, J. Liu, D. Nicol, Y. Yuan, G. Yan, and C. Grier, "RINSE: The real-time immersive network simulation environment for network security exercises," in *Proc. 19th Workshop Principles Adv. Distrib. Simulat.*, 2005, pp. 119–128.

[68] A. Giani, G. Karsai, T. Roosta, A. Shah, B. Sinopoli, and J. Wiley, "A testbed for secure and robust SCADA systems," *ACM SIGBED Rev.*, vol. 5, no. 2, p. 4, 2008.

[69] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simulat. Tools Techn. Commun. Netw. Syst. Workshops*, 2008, p. 60.

[70] C. Queiroz, A. Mahmood, J. Hu, Z. Tari, and X. Yu, "Building a SCADA security testbed," in *Proc. IEEE 3rd Int. Conf. Netw. Syst. Security*, 2009, pp. 357–364.

[71] Y. Kim, "Control systems lab using a LEGO Mindstorms NXT motor system," *IEEE Trans. Edu.*, vol. 54, no. 3, pp. 452–461, Aug. 2011.

[72] T. Morris, R. Vaughn, and Y. S. Dandass, "A testbed for SCADA control system cybersecurity research and pedagogy," in *Proc. ACM 7th Annu. Workshop Cyber Security Inf. Intell. Res.*, 2011, p. 27.

[73] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of SCADA control systems (TASSCS)," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT)*, 2011, pp. 1–7.

[74] *Modbus PLC Simulator*. Accessed: Apr. 27, 2020. [Online]. Available: http://www.plcsimulator.org/

[75] *SteelCentral Riverbed Modeler | Riverbed | GB*. Accessed: Apr. 27, 2020. [Online]. Available: https://www.riverbed.com/gb/products/steelcentral/steelcentral-riverbed-modeler.html

[76] A. Almalawi, Z. Tari, I. Khalil, and A. Fahad, "SCADAVT—A framework for SCADA security testbed based on virtualization technology," in *Proc. IEEE 38th Conf. Local Comput. Netw. (LCN)*, 2013, pp. 639–646.

[77] J. Ahrenholz, "Comparison of CORE network emulation platforms," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2010, pp. 166–171.

[78] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 847–855, Jun. 2013.

[79] R. Kuffel, J. Giesbrecht, T. Maguire, R. Wierckx, and P. McLaren, "RTDS—A fully digital power system simulator operating in real time," in *Proc. IEEE Int. Conf. Energy Manag. Power Del. (EMPD)*, vol. 2, 1995, pp. 498–503.

[80] *Powerfactory—DIgSILENT*. Accessed: Apr. 27, 2020. [Online]. Available: http://www.digsilent.de/en/powerfactory.html

[81] J. Wroclawski *et al.*, "DeterLab and the deter project," in *The GENI Book*. Cham, Switzerland: Springer, 2016, pp. 35–62.

[82] J. Mirkovic and T. Benzel, "Teaching cybersecurity with DeterLab," *IEEE Security Privacy*, vol. 10, no. 1, pp. 73–76, Jan./Feb. 2012.

[83] R. Leszczyna, "A review of standards with cybersecurity requirements for smart grid," *Comput. Security*, vol. 77, pp. 262–276, Aug. 2018.

[84] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quart.*, vol. 26, no. 2, pp. 13–23, 2002.

[85] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2009, pp. 1827–1834.

[86] B. M. Wilamowski and H. Yu, "Improved computation for Levenberg–Marquardt training," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 930–937, Jun. 2010.

[87] A. Van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm." *Neural Netw.*, vol. 5, no. 3, pp. 465–471, 1992.

[88] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 179–186, May 2011.

[89] Y. Yang *et al.*, "Multiattribute SCADA-specific intrusion detection system for power networks," *IEEE Trans. Power Del.*, vol. 29, no. 3, pp. 1092–1102, Jun. 2014.

[90] A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems," *Comput. Security*, vol. 46, pp. 94–110, Oct. 2014.

[91] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu, "SRID: State relation based intrusion detection for false data injection attacks in SCADA," in *Proc. Eur. Symp. Res. Comput. Security*, 2014, pp. 401–418.

[92] S. Ponomarev and T. Atkison, "Industrial control system network intrusion detection by telemetry analysis," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 2, pp. 252–260, Mar./Apr. 2016.

[93] S. Kalmegh, "Analysis of WEKA data mining algorithm REPTree, simple cart and RandomTree for classification of Indian news," *Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 1–9, 2015.

[94] I. Rish *et al.*, "An empirical study of the naive Bayes classifier," in *Proc. Workshop Empirical Methods Artif. Intell. (IJCAI)*, vol. 3, 2001, pp. 41–46.

[95] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *J. Educ. Res.*, vol. 96, no. 1, pp. 3–14, 2002.

[96] B. R. Gaines and P. Compton, "Induction of ripple-down rules applied to modeling large databases," *J. Intell. Inf. Syst.*, vol. 5, no. 3, pp. 211–228, 1995.

[97] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Decision tree analysis on J48 algorithm for data mining," *Proc. Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 6, pp. 1114–1119, 2013.

[98] S. Shitharth and P. Winston, "An enhanced optimization based algorithm for intrusion detection in SCADA network," *Comput. Security*, vol. 70, pp. 16–26, Sep. 2017.

[99] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, 2013.

[100] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, vol. 7, pp. 89507–89521, 2019.

[101] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. SCADA Security Sci. Symp.*, vol. 46, 2007, pp. 1–12.

[102] I. N. Fovino, A. Carcano, M. Masera, and A. Trombetta, "Design and implementation of a secure Modbus protocol," in *Proc. Int. Conf. Crit. Infrastruct. Protect.*, 2009, pp. 83–96.

[103] N. Goldenberg and A. Wool, "Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems," *Int. J. Crit. Infrastruct. Protect.*, vol. 6, no. 2, pp. 63–75, 2013.

[104] T. H. Morris, B. A. Jones, R. B. Vaughn, and Y. S. Dandass, "Deterministic intrusion detection rules for Modbus protocols," in *Proc. IEEE 46th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2013, pp. 1773–1781.

[105] A. Shahzad *et al.*, "Real time Modbus transmissions and cryptography security designs and enhancements of protocol sensitive information," *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.

[106] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. New York, NY, USA: Springer, 2013.

[107] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[108] *Secure Hash Algorithm 2*, Nat. Inst. Stand. Technol., Gaithersburg, MA, USA, 2002.

[109] N. Erez and A. Wool, "Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems," *Int. J. Crit. Infrastruct. Protect.*, vol. 10, pp. 59–70, Sep. 2015.

[110] W. Shang, Q. Qiao, M. Wan, and P. Zeng, "Design and implementation of industrial firewall for Modbus/TCP," *J. Comput.*, vol. 11, no. 5, pp. 432–438, 2016.

[111] D. S. Rana, N. Garg, and S. K. Chamoli, "A study and detection of TCP SYN flood attacks with IP spoofing and its mitigations," *Int. J. Comput. Technol. Appl.*, vol. 3, no. 4, pp. 1476–1480, 2012.

[112] L. Deng, Y. Peng, C. Liu, X. Xin, and Y. Xie, "Intrusion detection method based on support vector machine access of Modbus TCP protocol," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2016, pp. 380–383.

[113] S. Suthaharan, "Support vector machine," in *Machine Learning Models and Algorithms for Big Data Classification*. Boston, MA, USA: Springer, 2016, pp. 207–235.

[114] S.-C. Li, Y. Huang, B.-C. Tai, and C.-T. Lin, "Using data mining methods to detect simulated intrusions on a Modbus network," in *Proc. IEEE 7th Int. Symp. Cloud Service Comput. (SC2)*, 2017, pp. 143–148.

[115] W. Yusheng *et al.*, "Intrusion detection of industrial control system based on Modbus TCP protocol," in *Proc. IEEE 13th Int. Symp. Auton. Decentralized Syst. (ISADS)*, 2017, pp. 156–162.

[116] P.-H. Wang, I.-E. Liao, K.-F. Kao, and J.-Y. Huang, "An intrusion detection method based on log sequence clustering of honeypot for Modbus TCP protocol," in *Proc. IEEE Int. Conf. Appl. Syst. Invent. (ICASI)*, 2018, pp. 255–258.

[117] L. Hubert, "Approximate evaluation techniques for the single-link and complete-link hierarchical clustering procedures," *J. Amer. Stat. Assoc.*, vol. 69, no. 347, pp. 698–704, 1974.

[118] H. Dong and D. Peng, "Research on abnormal detection of modbus TCP/IP protocol based on one-class SVM," in *Proc. IEEE 33rd Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, 2018, pp. 398–403.

[119] Q.-L. Luo, K.-F. Xu, W.-Y. Zang, and J.-G. Liu, "Network protocol parser and verification method based on wireshark," *Comput. Eng. Design*, vol. 32, no. 3, pp. 770–773, 2011.

[120] Z. Hill, J. Hale, M. Papa, and P. Hawrylak, "Using bro with a simulation model to detect cyber-physical attacks in a nuclear reactor," in *Proc. 2nd Int. Conf. Data Intell. Security (ICDIS)*, 2019, pp. 22–27.

[121] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera, "DNPSec: Distributed network protocol version 3 (DNP3) security framework," in *Advances in Computer, Information, and Systems Sciences, and Engineering*. Dordrecht, The Netherlands: Springer, 2007, pp. 227–234.

[122] T. Mander, F. Nabhani, L. Wang, and R. Cheung, "Data object based security for DNP3 over TCP/IP for increased utility commercial aspects security," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2007, pp. 1–8.

[123] J. Bai, S. Hariri, and Y. Al-Nashif, "A network protection framework for DNP3 over TCP/IP protocol," in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, 2014, pp. 9–15.

[124] H. Li, G. Liu, W. Jiang, and Y. Dai, "Designing snort rules to detect abnormal DNP3 network data," in *Proc. IEEE Int. Conf. Control Autom. Inf. Sci. (ICCAIS)*, 2015, pp. 343–348.

[125] R. Amoah, S. Camtepe, and E. Foo, "Securing DNP3 broadcast communications in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 12, no. 4, pp. 1474–1485, Aug. 2016.

[126] J. Nivethan and M. Papa, "A Linux-based firewall for the DNP3 protocol," in *Proc. IEEE Symp. IEEE Technol. Homeland Security (HST)*, 2016, pp. 1–5.

[127] G. Yang and S.-Y. Chen, "Research on Linux firewall based on netfilter/iptables," *Comput. Eng. Design*, vol. 17, p. 22, Jan. 2007.

[128] H. Lin, A. Slagell, Z. T. Kalbarczyk, P. W. Sauer, and R. K. Iyer, "Runtime semantic security analysis to detect and mitigate control-related attacks in power grids," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 163–178, Jan. 2018.

[129] Y. Lu and T. Feng, "Research on trusted DNP3-BAE protocol based on hash chain," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 108, 2018.

[130] A. Paul, F. Schuster, and H. König, "Towards the protection of industrial control systems–conclusions of a vulnerability analysis of PROFINET IO," in *Proc. Int. Conf. Detect. Intrusions Malware Vulnerability Assess.*, 2013, pp. 160–176.

[131] Z. Feng, S. Qin, X. Huo, P. Pei, Y. Liang, and L. Wang, "Snort improvement on PROFINET RT for industrial control system intrusion detection," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, 2016, pp. 942–946.

[132] S. Pfrang and D. Meier, "Detecting and preventing replay attacks in industrial automation networks operated with PROFINET IO," *J. Comput. Virol. Hacking Techn.*, vol. 14, pp. 1–16, Feb. 2018.

[133] G. S. Sestito *et al.*, "A method for anomalies detection in real-time Ethernet data traffic applied to PROFINET," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2171–2180, May 2018.

[134] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh, and J.-C. Tan, "An intrusion detection system for IEC61850 automated substations," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2376–2383, Oct. 2010.

[135] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. Wang, "Intrusion detection system for IEC 60870-5-104 based SCADA networks," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2013, pp. 1–5.

[136] H. Yoo and T. Shon, "Novel approach for detecting network anomalies for substation automation based on IEC 61850," *Multimedia Tools Appl.*, vol. 74, no. 1, pp. 303–318, 2015.

[137] Y. Yang, K. McLaughlin, L. Gao, S. Sezer, Y. Yuan, and Y. Gong, "Intrusion detection system for IEC 61850 based smart substations," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, 2016, pp. 1–5.

[138] K. Wong, C. Dillabaugh, N. Seddigh, and B. Nandy, "Enhancing suricata intrusion detection system for cyber security in SCADA networks," in *Proc. IEEE 30th Can. Conf. Elect. Comput. Eng. (CCECE)*, 2017, pp. 1–5.

[139] A. Granat, H. HÖFKEN, and M. Schuba, "Intrusion detection of the ICS protocol EtherCAT," in *Proc. DEStech Trans. Comput. Sci. Eng.*, 2017, pp. 113–117.

[140] P. Chopade and M. Bikdash, "Modeling for survivability of smart power grid when subject to severe emergencies and vulnerability," in *Proc. IEEE Southeastcon*, 2012, pp. 1–6.

[141] C. Queiroz, A. Mahmood, and Z. Tari, "A probabilistic model to predict the survivability of SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 1975–1985, Nov. 2013.

[142] P. Chopade, M. Bikdash, and I. Kateeb, "Interdependency modeling for survivability of smart grid and SCADA network under severe emergencies, vulnerability and WMD attacks," in *Proc. IEEE Southeastcon*, 2013, pp. 1–7.

[143] J. Kirsch, S. Goose, Y. Amir, D. Wei, and P. Skare, "Survivable SCADA via intrusion-tolerant replication," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 60–70, Jan. 2014.

[144] M. Platania, D. Obenshain, T. Tantillo, R. Sharma, and Y. Amir, "Towards a practical survivable intrusion tolerant replication system," in *Proc. IEEE 33rd Int. Symp. Rel. Distrib. Syst.*, 2014, pp. 242–252.

[145] I. Lopez, M. Aguado, C. Pinedo, and E. Jacob, "SCADA systems in the railway domain: Enhancing reliability through redundant MultipathTCP," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 2305–2310.

[146] J. Pacheco, C. Tunc, and S. Hariri, "Design and evaluation of resilient infrastructures systems for smart cities," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, 2016, pp. 1–6.

[147] A. Babay, T. Tantillo, T. Aron, M. Platania, and Y. Amir, "Network-attack-resilient intrusion-tolerant SCADA for the power grid," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2018, pp. 255–266.

[148] N. U. I. Hossain, R. Jaradat, S. Hosseini, M. Marufuzzaman, and R. K. Buchanan, "A framework for modeling and assessing system resilience using a Bayesian network: A case study of an interdependent electrical infrastructure system," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 62–83, Jun. 2019.

[149] H. Sándor, B. Genge, Z. Szántó, L. Márton, and P. Haller, "Cyber attack detection and mitigation: Software defined survivable industrial control systems," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 152–168, Jun. 2019.

[150] K. Demir, F. Nayyer, and N. Suri, "MPTCP-H: A DDoS attack resilient transport protocol to secure wide area measurement systems," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 84–101, Jun. 2019.

[151] D. Rehak, P. Senovsky, M. Hromada, and T. Lovecek, "Complex approach to assessing resilience of critical infrastructure elements," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 125–138, Jun. 2019.

[152] M. Hermann, T. Pentek, and B. Otto, "Design principles for Industrie 4.0 scenarios," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 3928–3937.

[153] K. W. Schmidt and E. G. Schmidt, "Distributed real-time protocols for industrial control systems: Framework and examples," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1856–1866, Oct. 2012.

[154] J. Skodzik, P. Danielis, V. Altmann, and D. Timmermann, "HaRTKad: A hard real-time Kademlia approach," in *Proc. IEEE 11th Consumer Commun. Netw. Conf. (CCNC)*, Jan 2014, pp. 309–314.

[155] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proc. Int. Workshop Peer-to-Peer Syst.*, 2002, pp. 53–65.

[156] M. Sági and E. Varga, "Dependable peer-to-peer SCADA architecture," *Acta Polytechnica Hungarica*, vol. 14, no. 6, pp. 231–243, 2017.

[157] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[158] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial Internet of Things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018.

[159] A. Triantafyllou, P. Sarigiannidis, and T. D. Lagkas, "Network protocols, schemes, and mechanisms for Internet of Things (IoT): Features, open challenges, and trends," *Wireless Commun. Mobile Comput.*, vol. 2018, Sep. 2018, Art. no. 5349894.

[160] S. K. Goudos, P. Sarigiannidis, P. I. Dallas, and S. Kyriazakos, "Communication protocols for the IoT-based smart grid," in *IoT for Smart Grids*. Cham, Switzerland: Springer, 2019, pp. 55–83.

[161] K. Kimani, V. Oduol, and K. Langat, "Cyber security challenges for IoT-based smart grid networks," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 36–49, Jun. 2019.

[162] P. Sarigiannidis, E. Karapistoli, and A. A. Economides, "Modeling the Internet of Things under attack: A G-network approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1964–1977, Dec. 2017.

[163] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) security: Current status, challenges and prospective measures," in *Proc. IEEE 10th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, 2015, pp. 336–341.

[164] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[165] J. Reeser, T. Jankowski, and G. M. Kemper, "Maintaining HMI and SCADA systems through computer virtualization," *IEEE Trans. Ind. Appl.*, vol. 51, no. 3, pp. 2558–2564, May/Jun. 2014.

[166] T. Cruz, R. Queiroz, P. Simões, and E. Monteiro, "Security implications of SCADA ICS virtualization: Survey and future trends," in *Proc. 15th Eur. Conf. Cyber Warfare Security (ECCWS)*, 2016, pp. 74–83.

[167] T. Alves, R. Das, A. Werth, and T. Morris, "Virtualization of SCADA testbeds for cybersecurity research: A modular approach," *Comput. Security*, vol. 77, pp. 531–546, Aug. 2018.

[168] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substation automation to future smart grids," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, 2013, pp. 558–563.

[169] O. N. Fundation, "Software-defined networking: The new norm for networks," ONF, Menlo Park, CA, USA, White Paper, pp. 2–6, 2012.

[170] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2011, pp. 161–180.

[171] S. E. Schechter, J. Jung, and A. W. Berger, "Fast detection of scanning worm infections," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2004, pp. 59–81.

[172] J. Twycross and M. M. Williamson, "Implementing and testing a virus throttle," in *Proc. USENIX Security Symp.*, vol. 285, 2003, p. 294.

[173] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, 2005, pp. 32–32.

[174] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in *Proc. ACM Symp. Appl. Comput.*, 2003, pp. 346–350.

[175] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[176] T. Xing, D. Huang, L. Xu, C.-J. Chung, and P. Khatkar, "SnortFlow: A OpenFlow-based intrusion prevention system in cloud environment," in *Proc. IEEE 2nd GENI Res. Educ. Exp. Workshop (GREE)*, 2013, pp. 89–92.

[177] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," in *Proc. IEEE 19th Int. Conf. Netw. Protocols (ICNP)*, 2011, pp. 7–12.

[178] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.

[179] Y. Yu, C. Qian, and X. Li, "Distributed and collaborative traffic monitoring in software defined networks," in *Proc. ACM 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 85–90.

[180] E. G. da Silva, L. A. D. Knob, J. A. Wickboldt, L. P. Gaspary, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2015, pp. 165–173.

[181] T. A. Tang, L. Mhamdi, D. C. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. IEEE Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, 2016, pp. 258–263.

[182] A. Irfan, N. Taj, and S. A. Mahmud, "A novel secure SDN/LTE based architecture for smart grid security," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Depend. Auton. Secure Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM)*, 2015, pp. 762–769.

[183] W. Machii *et al.*, "Dynamic zoning based on situational activitie for ICS security," in *Proc. IEEE 10th Asian Control Conf. (ASCC)*, 2015, pp. 1–5.

[184] M. B. de Freitas, L. Rosa, T. Cruz, and P. Simões, "SDN-enabled virtual data diode," in *Computer Security*. Cham, Switzerland: Springer, 2018, pp. 102–118.

[185] J.-H. Yun, Y. Chang, K.-H. Kim, and W. Kim, "Security validation for data diode with reverse channel," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Security*, 2016, pp. 271–282.

[186] S. J. Walker, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. London, U.K.: John Murray, 2014.

[187] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[188] M. M. Rathore, A. Ahmad, and A. Paul, "Real time intrusion detection system for ultra-high-speed big data environments," *J. Supercomput.*, vol. 72, no. 9, pp. 3489–3510, 2016.

[189] *DARPA Intrusion Detection Data Sets*. Accessed: Apr. 27, 2020. [Online]. Available: https://www.ll.mit.edu/rd/datasets

[190] *KDD Cup 1999 Data*. Accessed: Apr. 27, 2020. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[191] *NSL-KDD Dataset*. Accessed: Apr. 27, 2020. [Online]. Available: http://www.unb.ca/cic/datasets/nsl.html

[192] *Apache Spark*. Accessed: Apr. 27, 2020. [Online]. Available: http://spark.apache.org/

[193] K. Vimalkumar and N. Radhika, "A big data framework for intrusion detection in smart grids using Apache spark," in *Proc. IEEE Int. Conf. Adv. Comput. Commun. Informat. (ICACCI)*, 2017, pp. 198–204.

[194] P. Natesan, R. R. Rajalaxmi, G. Gowrison, and P. Balasubramanie, "Hadoop based parallel binary bat algorithm for network intrusion detection," *Int. J. Parallel Program.*, vol. 45, no. 5, pp. 1194–1213, 2017.

[195] *Apache Hadoop*. Accessed: Apr. 27, 2020. [Online]. Available: http://hadoop.apache.org/

[196] A. Triantafyllou, P. Sarigiannidis, A. Sarigiannidis, E. Rios, and E. Iturbe, "Towards an anonymous incident communication channel for electric smart grids," in *Proc. ACM 22nd Pan Hellenic Conf. Informat.*, 2018, pp. 34–39.

[197] A. J. C. Bell, M. B. Rogers, and J. M. Pearce, "The insider threat: Behavioral indicators and factors influencing likelihood of intervention," *Int. J. Crit. Infrastruct. Protect.*, vol. 24, pp. 166–176, Mar. 2019.

[198] *Guidelines for Smart Grid Cyber Security (Vol. 1 to 3)*, document NIST IR-7628, NIST, Gaithersburg, MA, USA, Aug. 2010.

[199] *Protecting Industrial Control Systems, Recommendations for Europe and Member States*, Eur. Netw. Inf. Agency, Heraklion, Greece, 2011.

Dimitrios Pliatsios (Member, IEEE) received the undergraduation degree from the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece, in 2016. He is currently pursuing the Ph.D. degree with the Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece. His research interests are in the areas of 5th generation mobile networks, network virtualization technologies, and computer & network security.

Panagiotis Sarigiannidis (Member, IEEE) received the B.Sc. and Ph.D. degrees in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. He has been an Assistant Professor with the Department of Informatics and Telecommunications Department, University of Western Macedonia, Kozani, Greece, since 2016. He has published over 120 papers in international journals, conferences and book chapters. He has been involved in several national, EU and international projects. He is currently the Project Coordinator of the H2020 Project SPEAR: Secure and Private Smart Grid (DS-SC7-2017) and the Operational Program MARS: Smart Farming With Drones (Competitiveness, Entrepreneurship, and Innovation), while he serves as a Principal Investigator with the Erasmus+ KA2 ARRANGE-ICT: Partnership for Addressing Megatrends in ICT (Cooperation for Innovation and the Exchange of Good Practices). His research interests include telecommunication networks, Internet of Things, and telecommunications and network security.

Thomas Lagkas (Senior Member, IEEE) received the Ph.D. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 2006. He is an Assistant Professor with the Computer Science Department, International Hellenic University. He was a Lecturer and then a Senior Lecturer with the CITY College, University of Sheffield International Faculty, from 2012 to 2019. He also served as the Research Director for the Computer Science Department, CITY College and a Leader of the ICT Track of the South-East European Research Centre. His research interests are in the areas of IoT communications and distributed architectures, wireless communication networks, QoS in medium access control, mobile multimedia communications, power saving/fairness ensure for resource allocation in wireless sensor-cooperative broadband networks as well as in hybrid fiber-wireless networks, e-health data monitoring, 5G systems, flying ad hoc networks, communication security, and computer-based educational technologies with more than 70 publications at a number of widely recognized international scientific journals and conferences. He also participates in the Editorial Boards of the following journals: IEEE Internet of Things, *Computer Networks*, *Telecommunication Systems*, and the *EURASIP Journal on Wireless Communications and Networking*. He is a fellow of the Higher Education Academy, U.K.

Antonios G. Sarigiannidis (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree from the Department of Informatics, Aristotle University, Thessaloniki, Greece, in 2007, 2009, and 2016, respectively. He has authored or coauthored more than 10 journal and conference papers. His research interests include medium access protocols, dynamic bandwidth allocation and traffic engineering in optical, wireless and hybrid optical-wireless networks, visualization, analytics, and security analysis. He received the CCNA Routing and Switching certification from Cisco in 2018. He is currently with Sidroco Holdings Ltd., Limassol, Cyprus.