EXPERIMENTING WITH DIGITAL SIGNATURES OVER A DNP3 PROTOCOL IN A MULTITENANT CLOUD-BASED SCADA ARCHITECTURE

MARIUS MARIAN, (Member, IEEE), ADELIN CUSMAN, FLORIN STÎNGĂ, DRAGOS IONICĂ, AND DAN POPEȘCU, (Member, IEEE)

1Department of Computer and Information Technology, University of Craiova, 200440 Craiova, Romania
2Cert Digital, Centrul de Calcul S.A., 210132 Tg. Jiu, Romania
3Department of Automation and Electronics, University of Craiova, 200440 Craiova, Romania

ABSTRACT The ideas presented in this paper are summarized as follows. The first idea entails improving the security of supervisory control and data acquisition (SCADA) architectures by means of asymmetric cryptography and digital signatures and measuring the performance overhead. This allows achieving some obvious subsequent goals such as data-origin authentication, and the traceability and implicit non-repudiation of commands given to intelligent field and direct control equipment. The possibility to include digital signatures with a minimum impact on a standard and a reliable data communication protocol, such as Distributed Network Protocol version 3 (DNP3), also known to have a mature, industrially validated, open-source implementation, has been tested and the results are presented. A second idea concerns designing and developing a multitenant cloud-based architecture for a SCADA environment. This hypothesis focuses on certain SCADA operators that manage multiple industrial control systems (ICS) and intend to consolidate process data in a centralized manner.

INDEX TERMS Digital signatures, cybersecurity, SCADA, industrial control system, cloud-based services.

I. INTRODUCTION

This paper presents an experimental cloud-based platform for simulating and validating the use and performance of digital signatures within a typical industrial control system (ICS)/supervisory control and data acquisition (SCADA) architecture. The assumption that we made was that the remote terminal stations will be implemented by means of intelligent electronic devices (IED). This assumption is based on the relative low costs of such devices and the emerging trends in Industry 4.0. Our interest in developing such a platform is given by the current security challenges addressing both ICS and SCADA [1]–[8].

Our research addressed two directions. First, we designed and set up a cloud-based multitenant ICS/SCADA experimental cybersecurity platform. In operation, the “gap air” concept was proven a myth [8], and ICS/SCADA environments are no longer completely insulated from the Internet and information technologies. Therefore, except for extremely dedicated applications, future ICS/SCADA environments must be designed from scratch as a space where the operational technologies (OT) will have to coexist with information technology (IT) and the Internet. Moreover, in practice, a growing trend of using relatively inexpensive IED for the implementation of ICS/SCADA components has been noted. We can speculate that the future will eventually see a convergence of OT and IT, and the ICS/SCADA owners will be drawn into adopting cloud-based solutions for several financial and management reasons (e.g., reduction of design, development, operating, and maintenance costs).

The second and novel contribution of this research is that we worked on some security topics that may attract the interest of ICS/SCADA practitioners: the need for strong data-origin and end-to-end authentication (between the communicating ICS/SCADA components), chain of command traceability and non-repudiation. These aspects forced us to search for alternative ways in which a mature and already existing technology (i.e., public-key certificates and digital signatures) can be incorporated in common SCADA environments and data communication protocols.

The associate editor coordinating the review of this manuscript and approving it for publication was Ana Lucila Sandoval Orozco.
TABLE 1. List of acronyms and abbreviations.

| Acronym | Description |
|---------|-------------|
| ADC     | Analog to Digital Converter |
| API     | Application Programming Interface |
| CRoB    | DNP3 Control Relay Output Block |
| DAC     | Digital to Analog Converter |
| DBMS    | Database Management System |
| DCS     | Distributed Control System |
| DNP3    | Distributed Network Protocol version 3 |
| GPIO    | General Purpose Input/Output |
| GWAC    | GridWise Architecture Council |
| HMI     | Human-Machine Interface / Interaction |
| HSM     | Hardware Security Module |
| I2C     | Inter-Integrated Circuit |
| ICS     | Industrial Control System |
| IDS     | Intrusion Detection Systems |
| ICCP    | Inter-Control Center Communications Protocol |
| IEC     | International Electrotechnical Commission |
| IED     | Intelligent Electronic Device |
| IoT     | Internet of Things |
| IIoT    | Industrial Internet of Things |
| IP      | Internet Protocol |
| IPsec   | Internet Protocol security |
| IT      | Information Technology |
| JSON    | JavaScript Object Notation |
| KPI     | Key Performance Indicator |
| LAN     | Local Area Network |
| LCM     | Local Communication Module |
| LPDU    | DNP3 Link Protocol Data Unit |
| MACsec  | Media Access Control security |
| MTS     | Master Terminal Station |
| MTU     | Master Terminal Unit |
| NTP     | Network Time Protocol |
| OT      | Operational Technology |
| PKCS    | Public-Key Cryptography Standards |
| PLC     | Programmable Logic Controller |
| RAID    | Redundant Array of Independent Disks |
| RFC     | Remote Communication Framework |
| REST    | Representational State Transfer |
| RTS     | Remote Terminal Station |
| RTU     | Remote Terminal Unit |
| SaaS    | Software as a Service |
| SCADA   | Supervisory Control and Data Acquisition |
| SGAM    | Smart Grid Architecture Model |
| SGIRM   | Smart Grid Interoperability Reference Model |
| SICADD  | In Romanian, Sistem Centralizat de Achizitie de Date (Centralized Data Acquisition System) |
| SPF     | Single Point of Failure |
| SQL     | Structured Query Language |
| SPI     | Serial Peripheral Interface |
| SSL     | Secure Sockets Layer |
| TCP     | Transmission Control Protocol |
| TLS     | Transport Layer Security |
| TPM     | Trusted Platform Module |
| UART    | Universal Asynchronous Receiver-Transmitter |
| VPN     | Virtual Private Network |
| WAN     | Wide Area Network |
| X.509v3 | Standard format for public-key certificates |
| XML     | eXtensible Markup Language |

For this industrial type of research, DNP3 came first among the alternatives (e.g., Modbus, ICCP, IEC 61850, etc.), based on its multiple characteristics: it is a standard and is widely used in public utilities infrastructures (e.g., by power, oil, gas and water companies), is known to be robust and efficient, and is also amenable to developers for starting their work from a reliable, mature, and well-known open-source implementation (e.g., OpenDNP3 provided by Automatak).

This paper aims to show that intelligent electronic devices increasingly used as ICS/SCADA middleware and field components have enough computation power to deliver the results expected by the asymmetric cryptography in the generation, respectively, in the verification and validation phases of the electronic signatures. At the same time, both the master terminal station (MTS)/master terminal unit (MTU), and the remote terminal station (RTS)/remote terminal unit (RTU) can support and deliver the security features introduced earlier, thus achieving a secure cloud-based SCADA communication and operation. The DNP3 protocol will be used to accommodate digital signatures by establishing an operational configuration and agreement between the communicating components (MTU – RTU/outstation) A prototype of the proposed cloud-based SCADA architecture was tested with this extension of DNP3.

II. RELATED WORK

We have examined standards, journal or conference papers, and technical reports that address the specific topics of this paper. We have mainly investigated the related work with a focus on practical implementations, validated data and results that could support our applied research for an answer on how to incorporate end-to-end and data-origin authentication into an operational ICS/SCADA environment. In what follows, we will detail the findings.

ICS/SCADA architectures have traditionally been closed environments that were dedicated for achieving specific functionalities with an acceptable and a priori known number of devices and operators involved. The technical focus was on availability, robustness, integrity and reliability above all. SCADA operators were reluctant in opening the communication network toward the Internet. For various, mainly non-technical reasons (such as reducing operation and maintenance costs, remote monitoring and management, etc.), the network perimeter was voluntarily breached by the ICS/SCADA owner, and thus, the internal field and direct control components were suddenly accessible and available from the organization network that in turn was connected to the Internet. The effect was that curious eyes and skilled actors (but not always legitimate) were increasingly able to acquire the process data and even intervene in the controlled processes. Once this happened, an avalanche of security solutions was proposed, designed and implemented, mainly in order to patch and insulate the original perimeter breach.

One obvious observation is that this trend of allowing access to the internal SCADA network and components became a fact and cannot be reversed. ICS/SCADA systems will likely no longer be designed to be insulated from the Internet. Several reports and studies have proven that the air gap has definitively become a myth [8].
To date, the technical literature has documented both a timeline and a taxonomy of security attacks against industrial control systems (SCADA included) ranging from the infamous Stuxnet attack on Siemens Step7-controlled Iranian nuclear centrifuges [1], to the German steel mill cyberattack [2], to the 2015 and 2016 power blackouts in Ukraine [3], [4], to the TRISIS attack against Schneider Triconex safety instrumentation systems for Saudi oil and gas facilities [5], or to the CRASHOVERRIDE malware attack on power grids [6]. The list may continue given the presence of well-organized and well-trained advanced persistent threats (APT) groups, detailed in [4] and [7].

Based on a rather simplifying classification, the architectural blocks of interest for any attack are the hardware (equipment), the software and the communication network. Attackers may have a multitude of options in deciding the attack vectors and the vulnerabilities to target [9]. The effects of the vulnerabilities in the hardware, software, network or their operational configuration manifest in the form of security threats. An interesting overview of the major threats to SCADA systems is provided by Babu et al. [10] in the following order: zero day vulnerabilities, lack of task prioritization, database injections, and last, communication protocol issues.

Among the generic and well-known threats such as loss of availability, integrity and confidentiality, Ghosh and Sampalli [11] mention one point of interest for our research which is the lack of authentication in the DNP3 protocol (yielding masquerading attacks). Gao et al. [12] also report the lack of end-to-end authentication in DNP3 and the susceptibility of the protocol in front of replay attacks. They provide hints and propose support and solutions (using neural networks for training an intrusion detection system) against injection and denial of service attacks. However, the replay attacks in SCADA are determined to remain particularly hard to detect.

One related and frequently encountered vector of attack is deception; it is accomplished in many ways in cyberspace, including for example, identity deception (one entity is made to believe in the false identity claimed by another), trojan horse deception (when a malware is planted into an innocent software), or denial of service (when apparently innocent requests are forwarded toward a service with the intent of flooding its processing capabilities). An interesting deception attack scenario applied to a water management company is presented in [13]. The authors proved that stealth deception attacks can bypass the SCADA security mechanisms and remain undetected. In [14], Mo et al. present an interesting approach for detecting integrity attacks against SCADA by using autocorrelation (and analyzing the implicit authentication signal). The countermeasure proposed by [14] is validated on a simplified model of a power system (i.e., a micro-grid).

Radvanovsky and Brodsky [15] in discussing the future of control systems, mention the security issues associated with the lack of authentication in DNP3 and bring forward the updated security requirements of IEC standards 61850 and 62351 that emphasize the need for secure authentication for MTUs and RTUs in any SCADA environment. The remaining problems to be solved in practice, by those that choose IEC 62351 is to determine, likely on their own, how to implement those security requirements, how to handle the needed components of a dedicated public-key infrastructure (i.e., certification authorities, key servers, revocation servers/lists, etc.), how to manage the cryptographic keys on various SCADA components, and how, what and for how long to log the associated data/information. A step forward end-to-end authentication is made by Ruland and Sassmannshausen in [16]; in their paper, they describe a possible way to achieve the IEC 62351 requirement for smart grid security by means of XML signatures and X.509 certificates. The implementation details are not completed, and the impact on the communication layer is not quantified.

Moreira et al. [17] discuss the application, performance and impact of common security protocols on the overall operation of a real-time SCADA environment with a focus on the security of substation communication. For this, a comparison of the IPsec (Internet Protocol security) and MACsec (Medium Access Control security) protocols is detailed, and the authors underline the necessity to implement and test the MACsec protocol performance in an IED-based testbed environment. Similarly, interesting ideas were derived from the research of Ghaleb et al. [18] but with a focus on the security of the MTS/human-machine interface (HMI) and RTS communication analyzing the behavior and the impact of three types of attack: replay, man-in-the-middle and stealth command modification. Unfortunately, the communication protocol under scrutiny is not DNP3.

Data integrity as a security requirement is a way to understand the trust on the exchanged data in an ICS/SCADA environment [19]–[21]. The literature review showed us that a considerable body of research was involved in this direction. First, the way in which data are made trustworthy is analyzed and discussed in Zafara et al. [31], and in Qian et al. [32]. The comparison of cryptographic algorithms and the unavoidable management of cryptographic keys that are used within SCADA infrastructures are discussed, and various solutions are devised as seen in Choi et al. [25], Lim [26], Premnath et al. [27], Robles et al. [28], Rezai et al. [29], [30]. These works all imply that ICS/SCADA environments of the future will need and will therefore have to find solutions to tackle the burden of cryptography and various cryptosystems on the overall operational performance of the system.

For our research purposes, the last topic we have investigated throughout the technical corpus, was the migration of ICS/SCADA systems into the cloud paradigm. We believe the idea can be traced back to its first mention in the paper of Hopkinson et al. [23] concerning the quality of service associated with the communication networks of utility providers. Several attempts, proposals, and risk analyses have been made in the last decade about the possibility to adopt the cloud paradigm [24], or to perform an integration or a full
migration into the cloud [33]–[39], involving and using many concepts from Internet of Things (IoT), Industrial Internet of Things (IIoT) and Industry 4.0 Sajid et al. [34] classify and rank the best practices and recommendations for cloud adoption. Among the recommendations made, they discuss network security best practices such as network segregation, traffic analysis and intrusion detection, logging of all activities and components within the SCADA, continuous and prompt updating the software and configurations, plus different types of security verifications, audits and monitoring (addressing memory dumps, file integrity, malicious code or components, etc.).

Nguyen et al. [35] were influential and motivating since many arguments were delivered for integrating cloud services into SCADA. The paper presents a possible cloud-based SCADA for micro-grids dedicated to research purposes. It discussed some of the problems of integrating SCADA and cloud services, such as: the interoperability model (for micro-grids there are several proposals including GWAC – GridWise Architecture Council, SGAM – Smart-Grid Architecture Model, and SGIRM – IEEE Smart-Grid Interoperability Model for Energy and IT Operations), a suitable choice of communication protocols, smart and diligent decisions concerning the security policy of the ensemble, and the common information model (for the data exchanged and processed). The authors advise that future work is needed, and supplementary measures must be provided for achieving security and reliability of the postintegration system.

The relevant findings in the referenced literature are further presented in Table 2. The main topics of our ICS/SCADA research were compared against these referenced research papers in order to determine the potential element of novelty of our approach. We focused the analysis on four important points (as seen in Table 2): first, the need for strong authentication of entities involved in ICS/SCADA data exchanges. Second, the use of cloud-based architectures or solutions for industrial cyber-physical systems. Third, the subsequent approach for a multitenant cloud-based SCADA solution. Finally, the analysis of security and categories of attacks affecting ICS/SCADA communication protocols (with a focal point on the DNP3 protocol).

### III. GENERAL ARCHITECTURAL DETAILS

SCADA architectures have evolved over time from monolithic to distributed, and finally to interconnected. The monolithic SCADA generation was designed immediately around the process (one computer to one process), in a centralized manner with little or no connectivity. The second generation improves in its networking capabilities; however, the data communication protocols used are closed, with little or no internetworking. The first and second generations of SCADA architectures are tagged as legacy, closed and feeding on the air gap myth thus exposing many of the currently investigated security threats. The third generation is aware of the security threats; its components can be internetworked, and in general, it does observe some open architecture and the constraints and advantages of an open interconnected system. The findings presented in this section are based on the previous work of the authors [39], [40]. In [39] we investigated the cloud-based paradigm for the MTU design, and the security characteristics offered by the DNP3 protocol. In [40] we analyzed several storage systems (relational and non-relational) that could support the multitenant approach for a cloud-based SCADA architecture and proposed an experimental RTU architecture.

### A. SCADA COMPONENTS

We started from the fact that various computing resources and technologies are available for current SCADA components. We have adopted a simplified approach to the level stack depicted in a typical distributed control system (DCS), for each SCADA component. In other words, the MTU (called SICADD in Fig. 1, where SICADD in Romanian stands for “Centralized Data Acquisition System”) will consolidate the two upper levels for production control and scheduling, while the RTU will map onto the direct control and plant supervisory levels. In Fig. 2, Fig. 3, Fig. 4, and Fig. 5 are depicted the architecture and functional components for the remote and master terminal stations. Their software implementation will largely follow these architectural guidelines. Our research efforts were targeted toward the following objectives:

- Designing the SICADD-MTU operational platform that will ensure the requirements regarding system scalability, performance and security, and the design of notification, monitoring and reporting components;
- Designing the RTU experimental platform.

The centralized data acquisition system (SICADD-MTU) application programming interface (API) module will be implemented as a web service implementing the representation state transfer (REST) architectural style. In one of the possible operational approaches, the MTU will generate periodic requests to the enrolled RTUs. At its turn, the RTU will generate and transfer data packages toward the MTU API component. The web interface displayed by the SICADD portal will offer functions for:

- User management;
- Role management (i.e., security profiles);
- Management of access permissions;
- Management of hosted SCADA companies;
| Reference                                                                 | Summary of relevant findings / application domain                                                                 | ICS/SCADA topics investigated:                                                                 | 1 | 2 | 3 | 4 |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---|---|---|---|
| Karnouskos et al. [1], Lee et al. [2], Liang [3], Messaoud et al. [4],   | ICS/SCADA architectures are highly vulnerable to advanced persistent threats through a multitude of attack vectors. | 1. Strong authentication mechanisms                                                              | - | - | - | ✓ |
| Dragos [5], Slowik [6], [7], CyberX [8], Irmak & Erkek [9]               | Need to protect legacy ICS/SCADA. Need for strong data protection and security mechanism (e.g., user authentication and traceability) including the secure-by-design approach in all life-cycle phases. | 2. Cloud-based architecture                                                                     | - | - | - | ✓ |
| Babu et al. [10]                                                         | Review and analysis of security vulnerabilities affecting SCADA systems only, highlighting the current security risks targeting the communication protocols. | 3. Multitenant approach                                                                          | - | - | - | ✓ |
| Ghosh and Sampali [11]                                                   | Survey of existing security standards dedicated to ICS/SCADA based on vulnerabilities and prevention of attacks.    | 4. Communication protocol’ security issues/attack vectors                                        | - | - | - | ✓ |
| Gao et al. [12]                                                          | Use of neural networks in the process of training intrusion detection systems to protect and secure ICS/SCADA systems against typical data/command-injection and denial-of-service attacks. |                                                                                                  | - | - | - | ✓ |
| Amin et al. [13]                                                         | Detail a security threat assessment against a water SCADA system with a focus on stealthy deception attacks using downstream sensor measurements. |                                                                                                  | - | - | - | ✓ |
| Mo et al. [14]                                                           | Enumerate the conditions enabling the feasibility of replay attacks against cyber-physical systems. Simulations, results and a methodology for using a noisy control authentication signal are detailed. |                                                                                                  | - | - | - | ✓ |
| Radovanovsky & Brosky [15]                                               | A comprehensive handbook on SCADA and control systems security.                                                 |                                                                                                  | ✓ | - | - | ✓ |
| Ruland & Sassmannshausen [16]                                            | Investigates the need for end-to-end authentication in smart grid control systems.                             |                                                                                                  | ✓ | - | - | ✓ |
| Moreira et al. [17]                                                      | Survey of cybersecurity schemes and solutions for substation automation systems with a focus on IEEE 802.1AE standard (MACsec). |                                                                                                  | - | - | - | ✓ |
| Ghaleb et al. [18]                                                       | Provide a network security analysis for the PLC communication with the engineering stations that are in charge with their setup and configuration. |                                                                                                  | - | - | - | ✓ |
| Asghar et al. [19]                                                       | Survey of cybersecurity threats for ICS/SCADA with a focus on technologies (such as IoT and big analytics) and associated challenges. Investigate the use of risk assessment methodologies for improving the security of ICS/SCADA. |                                                                                                  | - | - | - | ✓ |
| Lezzi et al. [20]                                                        | A reference framework for cybersecurity issues in Industry 4.0 is proposed with a focus in several areas: systems vulnerabilities, cybersecurity threats, risks, and countermeasures. |                                                                                                  | - | - | - | ✓ |
| Bhamare et al. [21]                                                      | A survey of challenges to securing ICS/SCADA. Applicability of machine learning to ICS/SCADA cybersecurity. Cloud-migration issues and security requirements analysis. |                                                                                                  | - | ✓ | - | ✓ |
| Martins et al. [22]                                                      | Details of a smart metering infrastructure with a focus on communication protocols, compatibility of smart grids, data models, resilience and security. |                                                                                                  | - | - | - | ✓ |
| Reference | Description | Security | IoT | Cloud Computing | Cybersecurity |
|-----------|-------------|----------|-----|-----------------|--------------|
| Hopkinson et al. [23] | One of the first papers considering the quality-of-service in utility communication networks. | - | ✓ | - | - |
| Karnouskos et al. [24] | Details an architectural framework for cloud-based industrial cyberphysical systems. Relevant elements for virtualization of components and for the migration of ICS/SCADA into cloud. | - | ✓ | - | - |
| Choi et al. [25] | Propose a key management scheme for secure SCADA communication with a focus on message broadcasting, on the spreading of the computation load across the high-power nodes (such as MTU) and reducing the load on the RTU nodes. | ✓ | - | - | ✓ |
| Lim [26] | Proposes an ID-based key management architecture for securing communication in SCADA systems (fast session key establishment, secret key recovery, etc.) A pairing algorithm is introduced for the key management in typical SCADA components, to enable protection against accidents or malicious attacks on keys. | ✓ | - | - | ✓ |
| Premath et al. [27] | Provide results and insight in an experiment using NTRU-based encryption and authentication schemes suitable for SCADA systems. | ✓ | - | - | ✓ |
| Robles et al. [28] | Provide insight and results through a SCADA experiment by comparing symmetric and asymmetric encryption schemes as used in the communication between typical MTU and RTU components. | ✓ | - | - | ✓ |
| Rezai et al. [29] | Propose a hybrid key management architecture for SCADA network security. | ✓ | - | - | - |
| Rezai et al. [30] | Review of key management schemes and their performance in SCADA networks. | ✓ | - | - | ✓ |
| Zafara et al. [31] | Investigate the generic topic of secure data provenance in different communication scenarios (wireless sensor networks, IoT, cloud computing, etc.) | ✓ | ✓ | - | ✓ |
| Qian et al. [32] | Propose a trusted-ID referenced key scheme for SCADA in iron and steel plants. The scheme is particularly efficient against man-in-the-middle and replay attacks by using elliptic-curve cryptography. | ✓ | ✓ | - | ✓ |
| Lee et al. [33] | Propose a security testbed for various power control systems in which IoT technologies are applied. The focus was on the operational availability of the power control systems when security technologies and mechanisms are applied. Several cyberattacks were simulated and recognized by the deployed IDS. | - | - | - | ✓ |
| Sajid et al. [34] | Survey of security challenges in integrating cyber-physical systems with Internet of Things (utilizing cloud computing services). Highlights a summary of best practices and recommendations for such an integration. | - | ✓ | - | ✓ |
| Nguyen et al. [35] | Investigate the requirements for the interoperability of micro-grid platforms. For this, they propose a private hybrid cloud-based SCADA architecture. A platform-as-a-service communication model is used. | - | ✓ | - | ✓ |
| Kajati et al. [36] | An evaluation of different cloud providers and data communication efficiency in the context of the Reference Architecture Model for Industry 4.0 (RAMI). Provide network-related recommendations and a classification on levels of RAMI cyber-physical systems. | - | ✓ | - | - |
| Ferrag et al. [37] | Investigate the security solutions (requirements and vulnerabilities) for fog-based smart grid SCADA systems. Application of machine learning techniques for IDS for protecting against data injection attacks. | ✓ | ✓ | - | ✓ |
| Talaat et al. [38] | Investigate the use of a hybrid or a private cloud for the security, sorting and storage of data retrieved from power smart grid systems. | - | ✓ | - | ✓ |
| Marian et al. [39] | Investigate and propose the design of a multitenant cloud-based paradigm for a SCADA MTU. Extensions of DNP3 protocol to include end-to-end authentication (through digital signatures) are also proposed. | ✓ | ✓ | ✓ | ✓ |

**TABLE 2. (Continued.) Summary of referenced papers.**
TABLE 2. (Continued.) Summary of referenced papers.

| Reference          | Description                                                                                                                                                                                                 | Validation |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Marian et al. [40] | Analysis of storage systems appropriate for a multitenant cloud based MTU architecture. Proposes a SCADA RTU experimental platform to handle digital signatures.                                                  | ✓          |
| Khan et al. [42]   | Investigate the challenges associated to a secure cloud-migration approach for legacy ICS/SCADA systems.                                                                                                       | -          |
| Bošjančič Rakas et al. [43] | A review and classification of available intrusion detection systems for SCADA. Provides a structured evaluation methodology and also details for development of IDS testbeds.                                           | ✓          |
| Coates et al. [44] | Propose a trust system architecture to enhance the SCADA networks security with a minimal impact on TCP/IP communications.                                                                                  | ✓          |

B. DATA COMMUNICATION PROTOCOL

As stated, our experimental platform uses DNP3; this is a three-layer protocol that can be placed on top of a typical TCP/IP network. Each of the three layers of DNP3 applies a series of functions and transformations on the actual payload, preparing the encapsulated data for the next layer in the hierarchy. At the application layer, the transport functions will automatically split the data into smaller payloads such that the size and format of the data packages is respected. The same repeats at the next layers in the sequence.

To accommodate digital signatures, DNP3 will be extended through a user-defined object at the application layer. The final frame that will reach the next layer in the hierarchy will contain the actual value of the digital signature, and the metadata that will help interpreting this specific payload. The metadata will be stored in the header of the layer; in this case, it is called link protocol data unit (LPDU). The LPDU will be modified such that it includes the digital signature block of data. For this, assuming a pair of public and private keys on 2048 bits, the last 256 bytes (at least 16 User Data blocks) of
the LPDU will be reserved to store the actual payload of the signature.

At SICADD-MTU, the entire LPDU message will be stored in binary format, as it was received, along with the SICADD-MTU timestamp, and an electronic signature automatically generated by the system. Once stored, the LPDU message will be decoded and saved as discrete values in the database management systems (DBMS), from where it will be taken over by the processing and transformation functions. The primary DBMS structure that will save the discrete values, will follow the same persistence scheme as the RTU device.

C. SECURITY MECHANISMS DEPLOYED
This section presents the technical mechanisms that will be deployed in order to achieve the security objectives specific to any ICS/SCADA: availability, integrity and confidentiality.

1) CONFIDENTIALITY
Analyzing the functional architecture of the SICADD-MTU system, at least two vulnerable points can be observed; these are, in fact, the only entry points in the system, and their existence is necessary for the following:
- Access to the user interface of the system, displayed through the SICADD-MTU Web portal;
- The communication channel required to manage messages from and to the RTUs.

On the RTU side, the system will be configured not to expose public interfaces; all communication is done through a secure channel that is configured and initialized prior to the beginning of the proper communication session. This secure channel will be protected through a virtual private network (VPN) IPsec site-to-site connection, which will use a qualified digital certificate for privacy and authentication. IPsec together with a public-key certificate offers the safest and most scalable way to implement a VPN. The standard format for public-key certificates is X.509v3. IPsec authentication can be provided through pre-distributed keys (another easy-to-deploy scenario) or digital certificates (this scenario requires a trusted certification authority – CA – server on both sides).

Similarly, protection for users accessing the SICADD-MTU Web component during portal-level operating sessions will be achieved through a secure SSL/TLS channel.

2) INTEGRITY AND NON-REPUDIATION
An electronic (or digital) signature will guarantee both the integrity and the origin of the signatory of an electronic information.

For critical SCADA operations/commands, in order to have a legal value similar to the handwritten signature, the digital certificate (containing the public key of the operator) must be signed by a certification authority accredited according to European Regulation 910/2014. Such a signature is called a qualified digital signature.

Given the advantages and features set out above, we will also experiment the use of a qualified electronic signature both in the authentication process and to ensure the integrity of messages for SICADD-MTU Web access control and the critical SICADD-RTU communications.

DNP3 Secure Authentication provides authentication (i.e., message integrity only) at the application layer. It does not offer confidentiality or non-repudiation; it exclusively addresses security threats related to spoofing (interception and forgery), modification and replay (capturing authentication credentials and replaying them). It addresses authentication only for cryptographic key exchanges, not for other type of data (e.g., payload). The DNP3 Secure Authentication mechanism can be used in both directions, i.e., MTU to RTU or RTU to MTU and is based on a challenge and response handshake.

3) DIGITAL SIGNATURES OF DNP3 DATA
A digital signature is a mathematical construct agreed by the participants to a public-key cryptosystem that will enable a particular recipient to verify that a digital message was not tampered with (i.e., its integrity is preserved) and to trust the authenticity of its source (i.e., the message origin is indeed the one that is claimed). In asymmetric cryptography it is assumed that every participant has a pair of cryptographic keys (in practice, these are two values/numbers that are mathematically related such that data encrypted with one of the keys can only be decrypted by means of the other key).

![FIGURE 6. A typical digital signature scenario.](image-url)

The typical digital signature scenario and the message exchange between two participants in a public-key cryptosystem, are illustrated in Fig. 6. The sender (or the signer) is encrypting the hash value (or the digest) of the message to be signed, with her private key and thus, generates the digital signature of the message. The receiver verifies the digital signature by means of the sender’s public key (that may be available in a trustful manner via a digital certificate). This public-key certificate is issued by a trusted third party (termed as a certification authority – CA, a central component in a public-key infrastructure – PKI). The role of a CA is to enable confidence (i.e., trust) of all participants in the identity of the public-key owner. Such a digital (or public-key) certificate is a container for many pieces of information among
which we mention the public key (its actual numerical value), the identity of the public-key owner, the validity period of the certificate, the identity of the certification authority (the one that issues the certificate), and the digital signature of the certificate.

![Figure 7. DNP3-oriented message construction and signature generation.](image)

Our approach to encapsulate digital signatures within the DNP3 protocol, is illustrated in Fig. 7. To generate a digital signature, a digest value for the DNP3 message must be obtained from a hash algorithm. Obviously, both the MTU and RTU must agree upon the use of the same hash algorithm. For this, in the enrollment phase, every new RTU added to the SCADA architecture needs to negotiate (based on its computing resources) and communicate to the MTU an initial setup configuration for its cryptographic profile (and capabilities). The way in which this setup configuration is passed by the RTU to the SICADD-MTU is via a JSON (JavaScript Object Notation) object. This configuration will contain the signature message format, among other technicalities. An example of such configuration for the signature format and the acceptable hash algorithm is given in Fig. 8.

![Figure 8. Digital signature format for the RTU set-up configuration.](image)

Fig. 7 illustrates the construction of the DNP3 message and how the associated digital signature is generated by the RTU. DNP3 transactions appear as an exchange of user requests (from MTU to RTU) and user responses (from RTU to MTU). Every MTU is polling its enrolled RTUs for data acquired from the controlled processes. Each user request or user response may contain a subset of possible data types (i.e., binary input, analog input, counter input, control output, and analog output). These data types are organized as arrays. In our depicted scenario, we have used the analog input array that represents the input quantities that the RTU measured or calculated. The elements in each data array are indexed by values from 0 to [N-1], where N represents the number of blocks for the data type used (in our case, our RTU is transmitting the analog input data type).

![Figure 9. DNP3 frame structure.](image)

In DNP3, the Link layer is in charge of maintaining an active and reliable physical link between the MTU and the RTU. Furthermore, the DNP3 Link layer is manipulating the exchange of user requests and user responses by means of DNP3 frames (i.e., packets of data). When the volume of data is abundant, more than one frame is necessary. A DNP3 frame (see Fig. 9) has a maximum length of 292 bytes, and it contains a header and a data section. The header section is fixed and is 10 bytes long, containing information about the DNP3 frame (i.e., synchronization and control information, count of user bytes in the frame, the destination and source addresses, and a cyclic redundancy check). The data section depends on the amount of data to be transmitted. This data section is therefore divided into user data blocks. One user data block may vary from 1 up to a maximum of 16 bytes. For each user data block (i.e., every 16 bytes) a cyclic redundancy check is computed.

Returning to Fig. 7, one can observe that three analog input elements are stored in a single DNP3 message (24 bytes) that is subsequently hashed with the configured digest algorithm (i.e., SHA256). For the illustrated 24-byte DNP3 message, a 32-bytes digest value is obtained. This digest value is then encrypted with the RTU private key, and thus a 32-byte digital signature is generated. In the resulting DNP3 message, the initial 24 bytes (the actual payload for the analog inputs) are concatenated with the 32-bytes digital signature (encoded in BASE64). Once this signed DNP3 message arrives at the destination (i.e., at the SICADD-MTU), the encapsulated DNP3 payload data and its digital signature will be retrieved by the MTU. In Fig. 7, LINECOUNT represents the number of DNP3 frames necessary to transport the signature from the RTU to the MTU (in our case, LINECOUNT is 2 because the maximum payload of a DNP3 frame is 250 bytes). The signature will be extracted by concatenating the LINECOUNT DNP3 OctetStrings, and then performing a BASE64 decoding, as seen in Fig. 7.

### IV. SICADD-MTU EXPERIMENTAL PROTOTYPE

#### A. OPERATIONAL ARCHITECTURE

Considering the real-time constraints and the critical nature of certain operations that are performed in industrial processes,
it is necessary to plan, design, develop, test and eventually validate an operational architecture that will ensure both a large volume of writing requests, and a high availability of the system.

In support of these requirements, MTU-SICADD will take advantage of a cluster-oriented approach, as well as its individual subsystems in charge with processing the incoming data; these subsystems need to be designed and implemented to deliver the expected volume of writing requests for such a system, as seen in Fig. 10. All the components will be paired; this means that there is a functional cluster built from several machines that play the same role. The purpose of the cluster is thus to provide horizontal scalability, which means that we can theoretically add as many nodes as there are needed, to a certain instance of implementation, in order to support the volume of requests for a specific functionality, under optimal operational parameters. The communication within our SCADA architecture will make use of public networks/Internet, as illustrated in Fig. 10.

B. DATABASE ARCHITECTURE AND STRUCTURE
Considering our previous experience with database management systems, we have analyzed some of the most used database management systems, in order to identify a database management system that will provide native support for the functionalities required for the MTU experimental platform, along with a high level of performance. The main characteristics identified in our analysis that can be influenced by the database management system are as follows:

- Write speed (INSERT/UPDATE) of the records;
- Possibility and tools available for clustering, HA (high availability), Fail-Over and Load-Balancing;
- Monitoring features;
- Trigger support;
- Programmatic, programmed and periodic execution functionalities;
- Connectivity and support for JAVA and .NET programming/development languages.

Databases are the central component of a computer application, whose main purpose is to store data in a logical way, providing structure and mechanisms for data manipulation. There are several database management systems that can be used to manipulate and save large data (the so-called Big Data domain) available with either commercial or non-commercial (free) licenses such as Microsoft SQL Server, Oracle Database, MySQL, IBM DB, and so on. Many providers work on modern techniques of spatial-temporal databases, object-relational, parallel databases, etc.

An increasing share of production systems serving Software as a Service (SaaS) [41] platforms in recent years, is represented by so-called NoSQL or nonrelational database systems such as MongoDB, Couch DB or Couchbase. These types of databases integrate very well and offer high performance when used in conjunction with JavaScript/Ajax web technologies since information storage is done without processing, based on JSON documents.

Therefore, we have integrated in the SICADD-MTU system a NoSQL server that will be used as a buffer for receiving DNP3 packets just before their decoding and consequent storing in the relational database server (i.e., MySQL or MariaDB). These two last tasks will be made by means of separate threads of execution, which are independent of the main thread whose only function will be to store the received messages in the NoSQL server.

At the SICADD-MTU level, it is mandatory that any operation is monitored and audited, with the possibility of precise identification of the data involved and of the time at which it took place. For this purpose, the database management system will implement a dedicated structure for the permanent storage of this category of information.

The structure of the database, as defined in Fig. 11, will therefore allow us to perform and achieve:

- Operations of centralization and processing of the values acquired from the controlled industrial processes;
- Graphical representation of networks, sub-networks up to sensor level;
- Notification operations;
- Search and indexing operations;
- System auditing with the registration of the affected registration.
Within the investigated cloud-based SCADA architecture, there is a component that facilitates the access of the MTU operators to the information reported by the RTUs into the system, as well as to the operating commands that are specific for each RTU. This component is represented by the SICADD-MTU Web portal (see Fig. 5). The portal is operational for the moment within a private cloud and is accessible via Internet or in the case of a production environment, via a VPN connection. The associated database structure provides persistent support for the following elements (see Fig. 12):

- **PermissionPolicyUser** associated with portal user accounts;
- **DT_User** will contain details about portal users;
- **DT_Company** will list those company-type entities that will ensure the multitenant functionality of the portal (i.e., the simultaneous hosting of several entities, on the same platform);
- **DT_Location** specifies whether a company can have one or more locations where it has its RTU devices installed;
- **DT_Rtu** contains the definitions of RTU devices associated with a company’s location;
- **DT_AutoLine** specifies that a company can define a functional scheme that includes several RTUs, the portal thus offering the possibility of graphical representation of a production line and rapid tracking and diagnosis of its operation;
- **DT_Relay** specifies the execution relay entities associated with an RTU; an execution relay can be controlled via the portal;
- **DT_Sensor** indicates the sensor type entities, associated with an RTU, whose value can be read by the portal;
- **DT_SensorHistory** contains audit records for each sensor;
- **DT_RelayHistory** contains audit records for each relay unit.

In addition to the specific structures presented above, the portal also implements special models through which it provides audit support at the lowest level of information, and also at the field level in a table (see Fig. 11). This allows the administrator or an user with appropriate permissions to see when information has been changed and by whom (these are extremely important issues for traceability and liability of security objectives).

### C. THE MULTITENANT APPROACH

Under the SaaS paradigm [41], a “multitenant” architecture assumes that a single software instance operating on a single hardware infrastructure will serve multiple customers. Each client will have access to the same interface and the same set of functionalities; although they will operate on the same database, its structure (of the database) will allow and ensure access only to the client’s own information.

Within the SICADD-MTU Web portal, the administrator will define virtual SCADA MTU environments for each SCADA operator that will use the portal. The administrator will also define a user with the role of COMPANY_ADMIN who will have permissions to various critical functionalities, such as creating users associated with his company. Depending on the credentials of each user (of a specific company), that user will be able to subsequently register, manage, query or view the RTU structure of their own company.
The implementation of persistent security structures offers a great deal of granularity in establishing security profiles and access permissions:

- Establishing navigation permissions relative to the portal menu;
- Access permissions at the database entity level (read, write, create);
- Permissions to access information about entities (at the field level in tables - read, write);
- Conditional permissions, depending on certain values or functions applied to other information.

Defining security profiles also offers the possibility to implement a security scheme based on permission groups, allowing the administrator to assign a role to a user, depending on the operations and tasks he holds (or over which he has access privileges).

D. CLUSTERING, FAIL-OVER, REPLICATE, AND LOAD BALANCING MECHANISMS

While a typical, dedicated server is a powerful computer connected to the Internet through high-speed connections and hosted in a remote data center, a high-availability cluster is an advanced system that is equipped with redundant power supplies, a completely redundant communication network, storage disk drives structured in a RAID architecture, and with many other features, ensuring the highest level of availability and the best performance and reliability without any point of failure (SPF – Single Point of Failure).

In addition to availability, also important is the ability to withstand and recover after an incident or disaster. This requires implementing enough security mechanisms capable of ensuring both integrity and retrieval of information for those extreme cases/scenarios where a system may be compromised. Compromising a system can have many causes, ranging from hardware failures to actions such as security incidents with a human source (e.g., hacking). To eliminate these vulnerabilities, the following concepts are involved:

- Security backup assumes exporting data in a format from which they can be extracted and restored later.
- Fail-over – represents the mechanism by which one server will take over the function served by another server when the second one becomes inoperative or temporarily unable to serve the target functionality;
- Load-balancing – represents the mechanism by which a uniform distribution of service requests is provided, between servers with the same functionality in order to uniformize the load of the servers within the cluster.

V. RTU EXPERIMENTAL PROTOTYPE

From a functional point of view, an RTU must contain a modular structure that provides both scalability and flexibility in terms of data acquisition and/or control channels.

A. OPERATIONAL ARCHITECTURE

Such a structure is illustrated in Fig. 2. A brief presentation of the modules is necessary.

We have designed and developed the following modules:

- Wide Area Network (WAN) represents the public network to which the RTU is connected; in practice, it will be a network protected by a firewall. This network will be used by the RTU to establish the connection with the SICADD-MTU;
- Secure Communication Protocol: represents the module that will implement the secure connection protocol, through which the RTU will communicate with the SICADD-MTU;
- RTU Manager allows the management of the specific RTU commands (e.g., getStatus, reset, getTime, setTime, etc.);
- DATA REPORTING: this module will query the internal RTU database system, where the values acquired by the RTU are temporarily stored. This module will have access only to data reading operations and will not be able to alter/write the contents of the databases (i.e., DBMS module);
- DBMS represents a dedicated database management system in which the acquired values from the field equipment will be stored;
- RTU KERNEL: represents the main RTU module that will ensure the synchronization of the data acquisition operations with their storage in the DBMS module;
- Digital Signature Module (DSM): the module that will implement the specific digital signature functionalities, functionalities that will be accessed both by KERNEL and by the REPORTING module. This module will also provide support to the COMMUNICATION module for security and non-repudiation in RTU-MTU transactions;
- CONFIGURATION: this module will provide read and update functionalities for RTU configuration elements (input/output configuration, communication interfaces, authorization and authentication elements);
- WEB SERVER: this module will allow the administration of the RTU device. Through this module, the configuration may be consulted and updated, but it can also provide at least a raw form (text mode) for consulting the acquired field data;
- Local Communication Module (LCM): offers functionalities for direct data acquisition and control through analog/digital input and output;
- Digital I/O Module: manages the conversion of acquired digital values into values that can be stored in the DBMS or sent to external control modules via configured input/output;
- Analog I/O Module: manages the conversion of analog values into digital values or digital conversion into analog values for storage in DBMS, or for signaling external command and control systems through configured input/output;
- Remote Communication Framework (RCF): implements different communication interfaces for data acquisition or commands to external programmable logic controllers (PLC)/IED devices. Examples of protocols that can be implemented here are as follows: I2C, SPI, RS232, RS485, MODBUS, TCP, etc.
- Local Area Network (LAN): represents the local network interface through which the administration of the device can be performed or through which the RTU can communicate with other PLCs or IEDs;
- RS232: represents the serial communication interface with PLC/IED devices;
- MODBUS: represents the industrial communication interface with PLC/IED;
- General Purpose Input Output (GPIO): predefined number of access pins that can be configured as digital/analog inputs/outputs.

The experimental RTU will be able to achieve the following functionalities.

1) DIRECT DATA ACQUISITION

Direct data acquisition will be provided by the “Digital I/O Module” and “Analog I/O Module”. These modules will have direct access to the GPIO hardware interface and will receive commands from the LCM module, through which they will be asked to read or write a GPIO pin. The RTU will be able to perform:

- Direct acquisition of analog data via digital to analog converters (DAC) connected to GPIO pins configured as analog inputs;
- Direct data acquisition in digital format by reading GPIO pins configured as digital inputs;
- Analog control processes via analog to digital converters (ADC) connected to GPIO pins configured as analog outputs;
- Digital process control via GPIO pins configured as digital outputs.

2) INTERCONNECTION WITH PLC/IED

Digital communication with other data acquisition or control devices can be achieved through dedicated hardware interfaces. Depending on the degree of software implementation of the RCF module, RTU can collect data from other PLC or IED devices via I2C, SPI, UART or Ethernet. The RCF module will provide the necessary read/write functions of the hardware interfaces, functions that will be called by the KERNEL module for storage in the DBMS or for transmitting control commands to external PLC/IED devices.

3) COMMAND A PROCESS

The RTU device will be able to transmit the following types of commands:

- Local control, via the LCM module:
  - digital control through the write operation and configuration of a GPIO pin as digital output;
  - analog control through the write operation and configuration of a GPIO pin as analog output;
- Remote control via the RCF module. This type of control involves the digital transmission of control messages to devices connected to the RTU via LAN, I2C, SPI, RS232, RS485, ModBus interfaces.

4) SECURE CONNECTION WITH MTU

The “Secure Communication Protocol” module will be responsible for implementing the secure communication protocol with MTU. This module will implement an extended DNP3 protocol that will use the electronic signature for security.

5) RTU SPECIFIC COMMANDS

The “RTU MANAGER” module will be responsible for processing RTU commands. This module will consist of commands such as:

- reset: reset (restart) RTU device;
- get_Status: read RTU status (OPERATIONAL, BUSY, MALFUNCTION)
  - OPERATIONAL: The RTU is working properly,
  - BUSY: RTU processes other commands and cannot perform the operational test,
  - MALFUNCTION: the operational test detected malfunctions;
• `getErrorLog`: read audit errors in order to determine the cause for which the RTU is inoperative/partially inoperative;
• `clearErrorLog`: clears stored error stacks;
• `getTime`: reads the current time of the RTU device;
• `setTime`: sets the RTU time for synchronization with SICADD.

6) RECEIVING AND PROCESSING MESSAGES AT THE SOURCE
Since the sensor network connected to the RTU is a heterogeneous structure in relation to the format and protocols used, the RTU module will have to apply a series of transformations to the inputs in order to obtain a data packet to be stored in the internal DBMS. As presented in the previous report, entry into the RTU can be done by:
• LAN – via a TCP port;
• RS232, RS485, ModBus – through a serial interface;
• GPIO – through a direct connection to input/output pins exposed by the RTU.

The values or input messages will be processed in the first phase by the components specific to each interface RCF, respectively LCM. These components will send data packets to the RTU KERNEL in a format recognized by it.

To simplify intermodule communication and the integration of RTU KERNEL components at the code level, the RTU will outline several specific interfaces that the interconnected components will need to implement as shown in Fig. 15 and Fig. 16.

The kernel component will implement an `INetPoint` table with registered sensors. The data can be read both on order and based on the events generated by each network point. Each network point can generate a `DataReceived` event that the kernel can intercept by subscribing to this event. Since all actions performed by the kernel are asynchronous, reading the data, even in the case of a read operation performed on the point, will also be performed by subscribing to the `DataReceived` event, as shown in Fig. 17:

B. RTU DIGITAL SIGNATURE MODULE
For the RTU prototype, electronic signature validation is implemented for DNP3 control relay output blocks (CROB) data. In Listing 1 and Listing 2 are provided the pseudocodes for the MTU CROB command delivery (the MTU is sending a digitally signed command to the RTU), and the RTU CROB command execution (the RTU is verifying the digital signature prior to the execution of the received command from the MTU).

VI. RESULTS AND DISCUSSION
Applied research into the adoption of SCADA as a secure cloud service is important for several benefits it could provide:
• Enhancement of its capabilities to acquire, consolidate, and analyze large volumes of data;
• Data can be interpreted, reported and presented to people and business intelligence components, in a multitude of forms, depending on the needs (reports, graphs, functional or view diagrams, etc.);
• Capability to collect data from remotely localized field equipment, if the equipment is interfaced with an intelligent electronic device;
• Remote monitoring, management and control;
• Flexibility and scalability;
• Intensive use in industrial sectors (e.g., production, extraction, telecommunications, transport, energy, etc.)

To validate some of the security objectives stated in the introduction (strong data-origin authentication and end-to-end authentication, traceability of command chain,
non-repudiation, etc.), a development and testing platform has been designed, developed, installed, configured and operated. Furthermore, software was produced by the authors in order to support secure DNP3-based data communication between RTU and SICADD-MTU.

For deploying this SCADA architecture in a real scenario, we have acquired inspiration from using smart meters [22] (e.g., traffic light control, water management) over a city/district scenario. The design and prototyping of the software tools and applications as specified in the analysis stage include installing and configuring an RTU platform for development and testing purposes. The hardware platform used is composed of several components:

- Raspberry Pi 3 B+;
- Raspberry Pi Zero W;
- Pimoroni Enviro pHat;
- Pimoroni Automation pHat;
- A private cloud implementation based on MySQL and Microsoft Internet Information Services.

The private cloud used for development has the role of providing shared resources for development and testing, relieving local stations of common tasks such as hosting a database management system, or generating the project build for the testing team. Given that at the SICADD-MTU level, we will make use of the electronic signature, a network hardware security module (HSM) was installed in the private development cloud, which will allow the SICADD-MTU server application to generate the electronic signatures required to secure the DNP3 protocol. For the current stage of the research project, two servers and one HSM were installed and configured.

1) RTU SOFTWARE IMPLEMENTATION

The RTU software platform is implemented as a service (daemon application) that will run on the experimental RTU

---

**Listing 1 MTU CROB Command Delivery**

| Pseudocode procedure for MTU CROB Command Delivery |
|---|
| **Input:** status // 1 = ON/ 0 = OFF |
| **Output:** success (= TRUE) / fail (= FALSE) |
| 1 Begin |
| 2 headers_list: = [] |
| 3 If status = 1 THEN |
| 4 crob_command: = CreateCROB(LATCH_ON) |
| 5 Else |
| 6 crob_command: = CreateCROB(LATCH_OFF) |
| 7 End if |
| 8 crob_header: = CreateControlRelayOutputBlockHeader (crob_command) |
| 9 signature_flag_header: = CreateAnalogOutputDouble64Header(‘’X509’’) |
| 10 signature: = CreateX509Signature(crob_command) |
| 11 signature_headers: = CreateAnalogOutputDouble64Headers(signature) |
| 12 push signature_flag_header to headers_list |
| 13 push crob_header to headers_list |
| 14 push signature_headers to headers_list |
| 15 DirectOperate(header_list) |
| 16 End |

---

**Listing 2 RTU CROB Command Execution**

| Pseudocode procedure for RTU CROB Command Execution |
|---|
| **Input:** LPDU // AnalogOutputDouble64_0, ControlRelayOutputBlock, AnalogOutputDouble64_1, AnalogOutputDouble64_2, ... |
| **Output:** success (= TRUE) / fail (= FALSE) |
| 1 BEGIN |
| 2 signed: = false |
| 3 signature: = [ ] |
| 4 delayed_crob_command: = null |
| 5 COMMANDSET: = DecodeLPDU(LPDU) |
| 6 FOR every dp3_command in COMMANDSET |
| 7 IF dp3_command is AnalogOutputDouble64 THEN |
| 8 IF ConvertToString(dp3_command) is ‘’X509’’ THEN |
| 9 signed: = true |
| 10 ELSE |
| 11 IF signed is true THEN |
| 12 data: = ConvertToByteArray(cmd) |
| 13 push data to signature[ ] |
| 14 ELSE |
| 15 execute dnp3_command |
| 16 ENDIF |
| 17 ELSE IF dp3_command is ControlRelayOutputBlock THEN |
| 18 IF signed THEN |
| 19 delayed_crob_command: = CreateDelaydCrob(dnp3_command) |
| 20 ELSE |
| 21 execute CROB(dnp3_command) |
| 22 ENDIF |
| 23 ENDFOR |
| 24 IF delayed_crob_command is not null THEN |
| 25 IF SignatureIsValid(delayed_crob_command, signature) THEN |
| 26 execute delayed_crob_command |
| 27 ELSE |
| 28 reject execution of delayed_crob_command |
| 29 ENDIF |
| 30 ENDIF |
| 31 ENDIF |
| 32 ENDIF |

---
prototype (using a Raspberry Pi platform), and whose main role is to implement either:

- The role of outstation within a communication channel, based on the DNP3 protocol, or
- The RTU-specific commands and control tools.

The software application is written in C++ programming language. The compilation for the ARM-based Raspberry Pi platform was made using the GNU g++ compiler version 8.3.0 (available under the Raspbian 8.3.0-6+rc1 operating systems). For the software development process, Microsoft Visual Studio uses a “remote” compilation feature on a Raspberry Pi 3 platform. The Raspberry Pi 3 development platform configuration includes:

- Raspbian GNU Linux 10 operating system (Buster) - armv71;
- OpenSSL cryptographic software library 1.1.1d (available at https://www.openssl.org/);
- Wiringpi software library;
- Pimoroni software library;
- Pimoroni Enviro Phat add-on hardware board;
- Longmai mToken CryptoID USB cryptographic device.

To implement the DNP3 data communication protocol-specific structures, the software application uses the programming interface and the OpenDNP3 software library provided by Automatak (available at https://dnp3.github.io).

Following a re-evaluation of the implementation of the electronic signature transmission, it was decided that a method that does not affect the DNP3 protocol would be much more useful in order to maintain compatibility with existing MTU servers. This will allow the use of an RTU device built on the specifications of this project to be used in already functional SCADA systems.

The method chosen in the development of the RTU prototype, uses for the transmission of the electronic signature the DNP3 data packets of OctetString type. Since for each index in this class we have a size limitation (255 characters) and we cannot transmit all the information related to the signature on a single position, the signature data will be divided into several positions and will be reconstructed at the MTU level afterward. The typical tested time length for creating a digital signature on the RTU is under 115 ms, on average, while significantly less time is needed for validating one (under 3 ms).

2) MTU SOFTWARE IMPLEMENTATION

This application is used currently as a testing tool for the RTU prototype. Within the application, a DNP3 channel and a master module that will ensure communication with the outstation component of the RTU are implemented.

In addition to the role of testing the RTU, this application implements electronic signature generation for commands sent to the RTU and validates the electronic signature generated by the RTU.

The features of this application reflect the structure of the RTU prototype:

- Monitor the current status of the RTU:
  - Viewing the values of analog/digital inputs;
- RTU command and control;
- Display electronic signature information:
  - Identity of the signer;
  - Validity;
  - Data structure containing the electronic signature;
  - The result of the signature verification/validation process;
  - Errors resulting from the verification/validation process of the electronic signature.

The application is developed in C# programming language using the Microsoft Visual Studio 2017 platform and uses the programming interface and the DNP3 software library implemented by Automatak.

For the transmission of the electronic signature from the MTU to RTU, in this implementation the DNP3 data structures of AnalogOutputDouble64 type were used. Since the data type associated with this structure is a double, we therefore have the possibility of transmitting groups of 8 bytes in which we can encode the structure of the digital signature. In this way, we will go throughout the structure of the electronic signature that we will divide into groups of 8 bytes; these will be transformed into double-type values that will later be transmitted through AnalogOutputDouble64 values.

3) VALIDATING THE CORRECT USE OF THE DNP3 PROTOCOL

To verify the correct implementation of data structures in the DNP3 protocol, we tested the RTU prototype against the Automatak DNP3 simulator. This simulator is based on the OpenDNP3 library (available at https://dnp3.github.io) and is available for .Net Framework applications through the NuGet package manager (available at https://www.nuget.org/packages/opendnp3). The simulator correctly receives the values sent by the RTU (Raspberry Pi) and is also able to successfully transmit a simple LATCH_ON command that will turn on the light emitting diode (LED) on the Raspberry Pi (see Fig. 18).

![FIGURE 18. Validating the correct use of the DNP3 protocol on the RTU platform.](image-url)

4) DIGITAL SIGNATURE GENERATION AND VERIFICATION

The implementation of the proposed architecture involved generation and verification of digital signatures, on both RTU and SICADD-MTU. The process of generating a digital
signature on the RTU experimental platform is illustrated in Figure 19. One can observe the steps that produce the digital signature (its value can then be traced in Figure 20).

The exchange of DNP3 messages containing this digital signature value (from the RTU to the SICADD-MTU) can be easily observed in Figure 20 and Figure 21 (these Wireshark captures are highlighting the content of DNP3 messages with a focus on the beginning and the end of the digital signature value).

On the SICADD-MTU, the verification of the RTU’s digital signature is illustrated in Figure 22.

The verification of the digital signature involves the use of a digital certificate (containing the value of the public key that corresponds to the private key that was used for signing). As mentioned before (see subsection 3.B.3), a digital certificate is issued by a trusted third party. If this is a case, then the verifier of a digital signature could be involved in a process called certificate path (or chain) verification. This involves additional successive verifications of digital signatures made on successive digital certificates in a hierarchy of certification authorities, until a trusted certification authority is found. The process is illustrated in Figure 23.

The topic of certificate chain verification is rather complicated for certain categories of RTU and therefore the certificate authentication chain mechanism needs to be minimized as much as possible.

In our experimental platform, we have assumed that every RTU that is present in the SCADA infrastructure, is manually enrolled at some initial moment; the MTU simply adds the RTU’s digital certificate into its own cryptographic store. In what concerns the generation of the RTU’s digital certificate, there are at least two alternatives. So far, we have focused on the implementation of the first one described below.

Previous to the manual enrollment, the RTU software is responsible for generating the private and public keys, and also a digital certificate (containing among other useful information, the identity of the RTU, the value of its public key and...
most important, a digital signature made with the endorsement key of the associated TPM used – Trusted Platform Module. The TPM endorsement key cannot be modified or read but can be used to sign a digital certificate, through a protected mechanism. In this way, the digital certificate of the RTU is generated and then the MTU manually imports this public-key certificate and consequently trusts the certificate’s signature based on the trust invested in the TPM manufacturer’s key (the above-mentioned endorsement key).

A second alternative (which was not the main objective of our investigation) could involve indeed installing a digital certificate in the RTU. This could happen two ways. First, the RTU software generates the pair of asymmetric keys, followed by a certificate signing request (CSR) that is forwarded to a certification authority trusted by the MTU. Then, the CA makes the due verifications (concerning the association between the submitted public key and the identity of the owner of that public key) and consequently, if everything is fine, issues a digital certificate for the RTU. This second alternative would require the services of a public-key infrastructure (either an internal or external one) by the MTU. In this case, the RTU will need to handle a PKCS#12 token/file (a functionality for which support was made available in our software implementation) in order to accomplish the verification/validation of a certification path (up to the root CA certificate).

5) CRYPTOGRAPHIC SERVICES AVAILABLE VIA TRUSTED PLATFORM MODULES (TPM) ON RTU

Another interesting result we obtained during the tests for the Raspberry Pi-based RTU, was the successful set-up and use of the dedicated cryptographic Infineon SLB9670 TPM board. TPMs are widely used in critical applications to provide trust and performant cryptographic services. The board that we used, exposes the crypto chip over an SPI interface and has a Raspberry Pi compatible header allowing direct connection to the Raspberry Pi onboard GPIO header.

To use the cryptographic capabilities of the TPM, different software tools and a public-key cryptography standard PKCS#11 API library (available online at https://github.com/tpm2-software), were downloaded and compiled on the Raspberry Pi, and then integrated into the RTU platform.

As mentioned in the section above, we have used the endorsement key of the TPM to digitally sign the public key of the RTU.

6) PROPOSING A REPLAY ATTACK PREVENTION SCENARIO

We need to use this algorithm for clock syncing as the first task. This can be achieved in the case of Raspberry Pi through the ntpd service (based on the network time protocol - NTP). The NTP server will be configured similarly in both the MTU and RTU.

The blue part in Fig. 24 represents initialization of the prevention algorithm, a sequence in which the RTU and MTU exchange their own current times, so that the value of

LastRtuTimeStamp is memorized in the MTU, and LastMtuTimeStamp is memorized in the RTU.

This initialization sequence must be run every time the RTU or the MTU starts. In case of clock desynchronization, the connection will be dropped by the part that first detects the desynchronization. A desynchronization event is triggered if the peer’s timestamp is greater than the entity’s own timestamp. After the initialization sequence, DNP3 data exchange is handled according to the yellow part in Fig. 24.

Every message illustrated in the replay attack prevention proposed in Fig. 24 must be digitally signed by its originator (i.e., both the RTU and the MTU digitally sign their own messages, respectively). The timestamp will be included in a digitally signed DNP3 message as a countermeasure to a replay attack.

Thus, the replay prevention rule is that a data package received from the peer entity must always have a trusted timestamp that is strictly greater than the previous one. Timestamp insertion will also assure a unique and sequential signature on every data package.

VII. DATA AND SOFTWARE AVAILABILITY

The source code for the SCADA components is not made available as open-source at the moment of writing of this paper; it can be consulted on-request by contacting the authors.

VIII. CONCLUSION

The main contribution and novelty of the paper is validating the hypothesis that digital signatures can be used via the DNP3 protocol for end-to-end authentication by both MTU and IED/RTU components in practice. For this, we have developed a software prototype and measured the impact on the protocol configuration and the performance of the communication. The second contribution was the actual infrastructure and testbed for the SCADA architecture designed as a multitenant cloud-based operational prototype. This infrastructure can be used additionally for testing different vulnerabilities and attacks that are associated with SCADA.
and DNP3, thus allowing training programs for SCADA operators.

REFERENCES

[1] S. Karnouskos, “Stuxnet worm impact on industrial cyber-physical system security,” in Proc. IECON-37th Annu. Conf. IEEE Ind. Electron. Soc., Melbourne, VIC, Australia, Nov. 2011, pp. 4490–4494, doi: 10.1109/IECON.2011.6120048.

[2] R. M. Lee, M. J. Assante, and T. Conway, “German steel mill cyber attack,” Ind. Control Syst., vol. 30, no. 2, pp. 1–15, Dec. 2014. [Online]. Available: https://dx.doi.org/10.1109/MSCS.2014.6843932.

[3] B. I. D. Messaoud, K. Guennoun, M. Wahbi, and M. Sadik, “Advanced key-management system for SCADA based industrial control systems,” in Proc. 4th Int. Conf. Anti-Cyber Crime, Marrakesh, Morocco, Oct. 2016, pp. 1–6, doi: 10.1109/ACOSIS.2016.7843932.

[5] Dragos. (Dec. 2017). TRISIS Malware: Analysis of Safety System Targeted Malware. Accessed: Apr. 30, 2020. [Online]. Available: https://blog.dragos.com/trisis/trisis-01.pdf.

[6] N. Moreira, E. Molina, J. Lázaro, E. Jacob, and A. Astarloa, “Cybersecurity for industrial control systems: Issues, technologies, and challenges,” Comput. Netw., vol. 165, Dec. 2019, Art. no. 106946, doi: 10.1016/j.comnet.2019.106946.

[10] B. Babu, T. Ijyas, P. Muneer, and J. Varghese, “Security issues in SCADA and DNP3, thus allowing training programs for SCADA Int.”

[13] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, “Cyber security of cyber-physical systems: Issues, technologies, and challenges,” Comput. Netw., vol. 165, Dec. 2019, Art. no. 106946, doi: 10.1016/j.comnet.2019.106946.

[20] M. Lezzi, M. Lazio, and A. Corallo, “Cybersecurity for industry 4.0 in the current literature: A reference framework,” Comput. Ind., vol. 103, pp. 97–110, Dec. 2018, doi: 10.1016/j.comind.2018.09.004.

[21] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, “Cybersecurity for industrial control systems: A survey,” J. Comput. Secur., vol. 89, Feb. 2020, Art. no. 101677, doi: 10.1016/j.jocs.2019.101677.

[22] J. F. Martins, A. Gonçalves-Pronto, V. Delgado-Gomes, and M. Sanduleac, “Smart meters and advanced metering infrastructure,” in Pathways to a Smarter Power System. New York, NY, USA: Academic, Apr. 2019, ch. 4, pp. 89–114, doi: 10.1080/08-0102592-5.00004-1.

[23] K. Hopkinson, G. Roberts, X. Wang, and J. Thorp, “Quality-of-service considerations in utility communication networks,” IEEE Trans. Power Del., vol. 24, no. 3, pp. 1465–1474, Jul. 2009, doi: 10.1109/TPWDR.2008.2002966.

[24] S. Karnouskos, A. W. Colombo, T. Bangemann, K. Manninen, R. Camp, M. Tilly, M. Sikora, F. Jammes, J. Delsing, J. Eliasson, P. Napper, J. Hu, and M. Graf, “The IMC-AESOP architecture for cloud-based industrial cyber-physical systems,” in Industrial Cloud-Based Cyber-Physical Systems. Cham, Switzerland: Springer, May 2014, pp. 49–88, doi: 10.1007/978-3-319-05624-1_3.

[25] D. Choi, H. Kim, D. Won, and S. Kim, “Advanced key-management architecture for secure SCADA communications,” IEEE Trans. Power Del., vol. 24, no. 3, pp. 1154–1163, Jul. 2009, doi: 10.1109/TPWDR.2008.2005683.

[26] M. Y. Lim, “IKMS—An ID-based key management architecture for SCADA system,” in Proc. 7th Int. Conf. Netw. Comput. Geeyongangbuk-do, South Korea: IEEE, Sep. 2011, pp. 139–144.

[27] A. P. Premnath, J.-Y. Jo, and Y. Kim, “Application of NTRU cryptographic algorithm for SCADA security,” in Proc. 11th Int. Conf. Inf. Technol., Nov., Generat., Las Vegas, NV, USA, Apr. 2014, pp. 341–346, doi: 10.1109/ITNG.2014.38.

[28] R. J. Robles, M. Baltanás, R. Caytles, Y. Gelogo, and T.-H. Kim, “Comparison of encryption schemes as used in communication between SCADA components,” in Proc. Int. Conf. Ubiquitous Comput. Multimedia Appl., Daejeon, South Korea, Apr. 2011, pp. 115–118, doi: 10.1109/UCMA.2011.33.

[29] A. Rezai, P. Keshavarzi, and Z. Moravej, “Advance hybrid key management architecture for SCADA network security,” Secur. Commun. Netw., vol. 9, no. 17, pp. 4358–4368, Nov. 2016, doi: 10.1002/sec.1612.

[30] A. Rezai, P. Keshavarzi, and Z. Moravej, “Key management issue in SCADA networks: A review,” Eng. Sci. Technol., Int. J., vol. 20, no. 1, pp. 354–363, Feb. 2017, doi: 10.1016/j.jestch.2016.08.011.

[31] F. Zafar, A. Khan, S. Suhail, I. Ahmed, K. Hameed, H. M. Khan, F. Jabeen, and A. Anjum, “Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes,” J. Netw. Comput. Appl., vol. 94, pp. 50–68, Feb. 2017, doi: 10.1016/j.jnca.2017.06.003.

[32] J. Qian, C. Hua, X. Guan, T. Xin, and L. Zhang, “A trusted-ID referenced key scheme for securing SCADA communication in iron and steel plants,” IEEE Access, vol. 7, pp. 46947–46958, Apr. 2019, doi: 10.1109/ACCESS.2019.2909011.

[33] S. Lee, S. Lee, H. Yoo, S. Kwon, and T. Shon, “Design and implementation of cybersecurity testbed for industrial IoT systems,” J. Supercomput., vol. 74, no. 9, pp. 4506–4520, Dec. 2017, doi: 10.1007/s11227-017-2219-z.

[34] A. Sajid, H. Abbas, and K. Saleem, “Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges,” IEEE Access, vol. 4, pp. 1375–1384, 2016, doi: 10.1109/ACCESS.2016.2549047.

[35] V. H. Nguyen, Q. T. Tran, and Y. Besanger, “SCADA as a service approach for interoperability of micro-grid platforms,” J. Sustain. Energy, Grids Netw., vol. 8, pp. 26–36, Dec. 2016, doi: 10.1016/j.sjen.2016.08.001.

[36] E. Kajati, P. Papcun, C. Liu, R. Y. Zhong, J. Koziolek, and I. Zolotova, “Cloud based cyber-physical systems: Network evaluation study,” Adv. Eng. Informat., vol. 42, Oct. 2019, Art. no. 100988, doi: 10.1016/j.advengin.2019.100988.

[37] M. A. Ferrag, M. Babaghashou, and M. A. Yazici, “Cyber security for fog-based smart grid SCADA systems: Solutions and challenges,” J. Inf. Secur. Appl., vol. 52, Jun. 2020, Art. no. 102500, doi: 10.1016/j.jisac.2020.102500.

[38] M. Talaat, A. S. Alsayyari, A. Alblawi, and A. Y. Hatata, “Hybrid-cloud-based data processing for power system monitoring in smart grids,” Sustain. Cities Soc., vol. 55, Apr. 2020, Art. no. 102409, doi: 10.1016/j.scs.2020.102409.
[39] M. Marian, A. Cusman, D. Popescu, and D. Ionica, “A DNP3-based SCADA architecture supporting electronic signatures,” in Proc. 20th Int. Carpathian Control Conf. (ICCC), Krakow, Poland, May 2019, pp. 1–6, doi: 10.1109/CarpathianCC.2019.8765963.

[40] M. Marian, D. Popescu, F. Stinga, A. Cusman, and D. Ionica, “Exper-tise based SCADA Platform using Digital Signatures,” in Proc. 21st Int. Carpathian Control Conf., IEEE, High Tatras, Starý Smokovec, Slovak Republic, Oct. 2020. [Online]. Available: https://iccc.fberg.tuke.sk/

[41] G. Goth, “Software-as-a-service: The spark that will change software engineering,” IEEE Distrib. Syst. Online, vol. 9, no. 7, pp. 1–3, Jul. 2008, doi: 10.1109/MSO.2008.21.

[42] R. Khan, K. McLaughlin, B. Kang, D. Laverty, and S. Sezer, “A seamless cloud migration approach to secure distributed legacy industrial SCADA systems,” in Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT), Perth, WA, Australia, Feb. 2020, pp. 1–5, doi: 10.1109/ISGT45199.2020.9087760.

[43] S. V. B. Rakas, M. D. Stojanovic, and J. D. Markovic-Petrovic, “A review of research work on network-based SCADA intrusion detection systems,” IEEE Access, vol. 8, pp. 93083–93108, May 2020, doi:10.1109/ACCESS.2020.2994961.

[44] G. M. Coates, K. M. Hopkinson, S. R. Graham, and S. S. Sezer, “A trust system architecture for SCADA network security,” IEEE Trans. Power Del., vol. 25, no. 1, pp. 158–169, Jan. 2010, doi: 10.1109/TPWRD.2009.2034830.

MARIUS MARIAN (Member, IEEE) was born in Craiova, Romania, in 1975. He received the B.S. and M.S. degrees in computer science from the University of Craiova, in 1998 and 2000, respectively, and the Ph.D. degree in security of information systems from Politecnico di Torino, Italy, in 2003. Since 2003, he continued his postdoctoral research activity at Politecnico di Torino, on certificate distribution and certificate revocation mechanisms associated to public-key infrastructures. Since 2005, he has been a Lecturer with the Department of Computers and Information Technology, University of Craiova, Romania. His research interests focus on security of information systems and industrial control systems, and on security risk management. His professional memberships include ACM and SRAIT (Romanian association of control, computing and information technology engineers).

ADELIN CUSMAN was born in Tg. Carbunesti, Romania, in 1980. He received the degree in electronic engineering from the Polytechnic University of Timisoara, Romania. He is currently the Chief Technical Manager with the Central de Calcul S.A., an EU Accredited Certificate Authority that operates in Romania. With over 15 years of experience in electronic and software development, his main areas of research and expertise focus on cryptographic applications, software engineering, digital electronics, IoT design and IoT systems development, smart contracts, and blockchain technology for the automation of multi-step processes.

FLORIN STÎNGĂ was born in Craiova, Romania, in 1975. He received the B.Eng., M.S., and Ph.D. degrees in system engineering from the University of Craiova, in 2000, 2003, and 2012, respectively. He is currently a Lecturer with the Department of Automation and Electronics, Faculty of Automation, Computers and Electronics, Craiova. His research interests are in embedded systems, estimation and control of dynamical systems. He is a member of two national professional associations SRAIT (Romanian Association for Automation and Information Technology) and SRR (Romanian Robotics Society).

DRAGOS IONICĂ was born in Craiova, Romania, in 1980. He received the B.S. degree in mathematics-informatics and the M.S. degree in artificial intelligence from the University of Bucharest, Romania, in 2003 and 2005, respectively. He is currently a Chief Executive Officer with the Centrul de Calcul SA, one of the six European Union Accredited Certification Authorities that operate in Romania. In recent years, he has been the main drive behind the development of a number of technical skills and research ideas within the company all related to the field and practical applications of PKI. He is also interested in using new trends in PKI, data protection, smart contracts, and blockchain technology.

DAN POPESCU (Member, IEEE) was born in Craiova, Romania, in 1953. He received the Dipl. Eng. degree in automation and computers, and the Ph.D. degree in automatic control from the University of Craiova, in 1977 and 1997, respectively. He is currently a Professor with the Department of Automation and Electronics, University of Craiova. From 2004 to 2008, he was the Dean of the Faculty of Automation, Computers and Electronics, and from 2008 to 2016, he was a Vice-Rector for Research at the University of Craiova. His research interests include control engineering, modeling and simulation, robust control, and time-delay systems. He has been a member of IFAC—Robust Control Technical Committee, since 2008.

* * *