Authentication Layer for IEC 61113-3 Applications

Aydin Homay
Faculty of Computer Science
TU Dresden
Dresden, Germany
aydin.homay@mailbox.tu-dresden.de

Christos Chrysoulas
Faculty of Engineering
London South Bank University
London, United Kingdom
chrysouc@lsbu.ac.uk

Mario de Sousa
Faculty of Engineering
University of Porto
Porto, Portugal
msousa@fe.up.pt

***Abstract* — Mid 2010, the Stuxnet ICS attack targeted the Siemens automation products, and after this attack the ICS security was thrust into spotlight, automation products suppliers started to re-examine their business approach to cyber security. The OPC Foundation made also significant changes and improvements on its new design OPC-UA to increase security of automation applications but, what is still missing and seems to be not resolved any time soon is having security in depth for industrial automation applications. In this paper, we propose a simple but strong security control solution, what we will call a logic application level security particularly for SCADA and DCS. This proposed method is based on message integrity and should not be viewed as the main, nor the only level of protection that an industrial automation system is expected to have, but can be a low-level security procedure that avoids intelligent attacks such as Stuxnet.**

Keywords—Stuxnet, Obfuscation, Encryption, MAC, SCADA, DCS

I. Introduction

The nation's critical infrastructures (CI) such as those found in Supervisory Control and Data Acquisition (SCADA), Distributed Control Systems (DCS) and generally industrial control systems (ICS), are essential for day-to-day operation of the economy, security and government. These are ushered by insecure connectivity to traditional network. Electric power production and distribution, water treatment and supply, gas and oil production and distribution, nuclear plants, transportation systems, and telecommunication systems are excellent examples of CI. This paper is an extension of the work originally presented in conference ETFA 2016 [1].

Protecting and assuring the availability of CI is vital to the world economies. CI assets are often privately held and can cross international borders via industrial and non-industrial networks, for example The August 2003 northeast blackout, which also affected Canada, shows how CI crosses international boundaries [2]. On June 1999, at about 3:30 p.m. a 16 - inch - diameter steel pipeline owned by The Olympic Pipe Line Company ruptured and released about 237,000 gallons of gasoline into a creek that flowed through Whatcom Falls Park in Bellingham, Washington [3]. On April 23, 2000, Vitek Boden, a man who was successfully intruded into a Queensland, Australia wastewater management system 46 times to cause damage, again in April of 2000 the "ILOVEYOU" virus rendered a petroleum refinery in Texas inoperable [4]. A December 2002 report from Mechanical Engineering cites examples of "wardriving" into SCADA-controlled utilities [4]. In August 2003, a computer virus was blamed for bringing down train signalling systems throughout the eastern U.S. The signalling outage briefly affected the entire CSX system, which covers 23 states east of the Mississippi River [5]. In May 2004, coastguard stations around the UK were severely disrupted after a computer worm rough down IT systems. The Sasser worm hit all 19 coastguard stations and the service's main headquarters, leaving staff reliant on paper maps and pens [5]. Mid 2010, the Stuxnet ICS attack targeted the Siemens automation products, and after this attack the ICS security was thrust into spotlight and all automation products suppliers started to re-examine their business approach to cyber security, eliminates gaps previously viewed low risk and improve practice in general [6].

As can be seen from the previous examples, industrial control equipment is susceptible to computer-based attacks. It may therefore be concluded that computer-based equipment used in industrial automation needs to be protected against relevant attacks. The widely accepted approach to computer security is based on security in depth, meaning that the computer system is viewed as a layered structure and security is introduced at each of the layers. With this approach, even if an attacker manages to penetrate the defenses of the outer layer, that attacker does not have automatic access to all devices inside the network as each device will itself include an additional layer of security protections.

The rest of paper is structured as following. In Section II a short introduction to IC and PLC is provided. The IEC 61131-3 standard is introduced in section III. Section IV in depth analyzes the Stuxnet virus. All related to security standards information is in detail presented in section V. The proposed approach is well presented and documented in Section VI. Finally, Section VII concludes our work.

II. Industrial Control Systems and Programmable Logic Controllers

The basic operation of an ICS is shown in Fig. 1. The ICS is a general term for several types of control systems, that includes SCADA, DCS and other control system

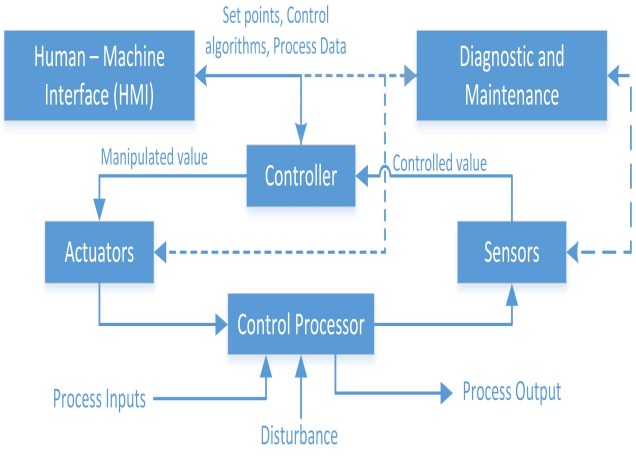

Fig. 1. The Industrial Control System operation in a general overview

configurations such as Programmable Logic Controllers (PLC).

The PLC was originally designed for small size factory automations, commonly referred to the "brain" of a factory, which did employ one or more machines with fair amount of the material transferred in line of the product. In such environment, a PLC must receive data from sensors and machines to control functionality and allow to operator visually monitor the product as they moved through the manufacturing line. Such manufacturing process has been very intensive logic control oriented with mostly high-speed requirements.

PLC devices are loaded with blocks of code and data written using a variety of languages, such IEC 61131-3 or IEC 61499. To make a PLC device functional it needs to be configured and programed through one of the above languages and usually a Windows computer based system called Control PC [6]. Once the PLC has been configured and programmed, the Control PC can be disconnected, and the PLC will function by itself.

Control loop is the most important part of DCS and SCADA that usually use one or more than one advanced PLC with a memory, processor, and network, Real-Time Operating System (RTOS) or Embedded Operating System (EOS). Control algorithms and logic which knowns by logic application or control logic, is typically written by an engineer using an engineering workstation that is distinct from the PLC, and once compiled logic applications are downloaded to the PLCs where they will run (Fig. 2). Control programs are commonly written using one or more of the programming languages defined in the IEC 61131-3 international standard. However, recently the IEC 61499 standard come in spotlight but still majority of industries have designed based on IEC 61131-3. To obtain security, both the engineering workstation as well as the PLC itself must be made secure.

III. The IEC-61131 Standard

The IEC 61131 standard standardizes the behavior of PLC systems. It is built out of several parts, which cover both the PLC hardware as well as the programming system. More specifically, part 3 of this standard (more commonly known as IEC 61131-3) defines the common concepts used in PLC programming as well as additional new programming methods. IEC 61131-3 sees itself as a guideline for PLC programming, not as a rigid set of rules.

The IEC 61131-3 standard focuses on the PLC programming languages, and how these programs should be interpreted and executed. It introduced five languages, which can be categorized into two parts: text based languages (IL - Instruction List, and ST -Structured Text) and graphical languages (LD - Ladder Diagram, FBD - Function Block Diagram, and SFC - Sequential Function Chart). Also, there is a possibility to use C language as a hosted function block inside of ST or FBD, which we call C function or C code and as we will see in the solution part of our paper to implement our idea to have an authentication protocol inside of IEC 61131-3 languages. [15] The important note is that, more than 90% of control logics around the world are developed based on this family which bring that to the spotlight [9].

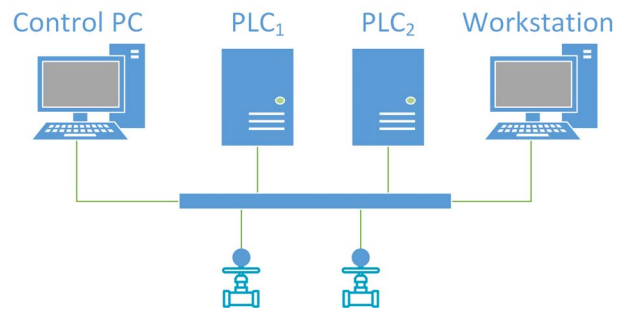

Fig. 2. Abstract respresentation of a distributed control system topology.

IV. Stuxnet Virus

The term computer virus was coined by Fred Cohen in 1985 [8]. But the new generation of viruses, particularly those ones is designed to attack the cyber-physical systems has so different behaviours than classical definitions. For example, viruses like Stuxnet, Duqu, and Flame were designed to steal information from industry or change the behaviour of control system by infecting the control logic and finally effecting on the main strategies of targeted organizations like the examples in the introduction. Such viruses, usually have a clear strategy. They want to be hidden. Therefore, they need to avoid any physical snap destructive behaviours, at least not until the end of the mission. However, the following explanation scenario is only speculation driven by the technical features of Stuxnet but it illustrates the above facts about the new generation of viruses which are going to target emerging technologies in the future of industrial automation particularly Industry 4.0 [6].

Once Stuxnet had infected a computer within the organization it aims at finding the Control PC (the PC has running WinCC/STEP7 application), which are typical Windows based computers with a data cable connection directly to a PLC to program, set configuration, define networks or configure I/O channels etc. Since most of these computers are non-networked, Stuxnet would first try to spread to other computers on the LAN through the zero-day vulnerabilities, two-year-old vulnerabilities etc. to come inside of the organization. Then, the virus tries to find the targeted computer through the removable drives. Stuxnet's goal was infecting specific type of PLC devices.

When Stuxnet finally found a suitable computer (through identifying ".tmp", ".s7p" or ".mcp" files), it would then replace the s7otbxdx.dll file to bug the communication between the Control PC and the connected PLCs. From this

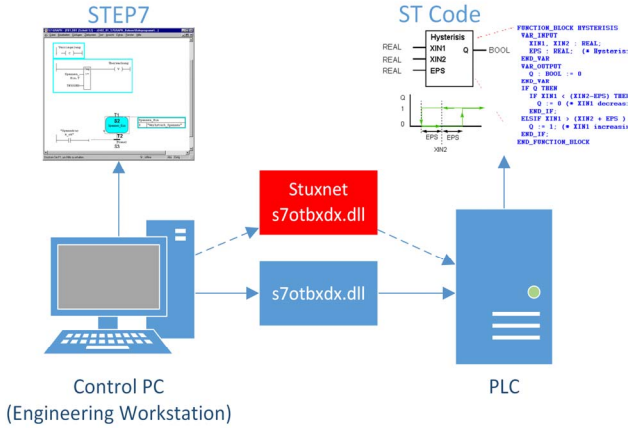

Fig. 3. Stuxnet can modify the ST code before downloading to the PLC by the bugged version of s7otbxdx.dll.

moment, the Stuxnet will be able to access the developed control loop logic on STEP7 software before downloading to the PLCs [6]. Fig. 3 shows that how Stuxnet can change the control loop logic before downloading time.

A. The Infection Process

The Stuxnet infects PLC through the code blocks and data blocks that will be injected into the PLC to alter it is behavior. The most common types of blocks are, Data Blocks (DB) contain program specification data types, System Data Blocks (SDB), contain configuration of the PLC. Organization Blocks (OB) or Program Organization Unit (POU based on the IEC 61131-3 standard terminology, which are the entry point of programs) and CPU cyclically executes them. Finally, Function Blocks (FB) which are standard code blocks.

Then, starts to attack the SDBs in order to find a DWORD at offset 50h equal to 0100CB2Ch [6]. This specifies the system uses the Profibus communications processor module namely CP 342-5 for SIMATIC S7-300 series [10]. Profibus is a standard industrial network bus used for distributed I/O. The result of this attack is to replace the original DP_RECV which is a standard function block used to receive networks frames on the Profibus by a malicious one. This way the malicious Stuxnet block takes control and can-do post processing on the packet data. Then, next step is to use a code-prepending infection technique to infect Organization Blocks.

Stuxnet writes malicious code to the beginning of OB1 after increasing the size of original block to execute malicious code at the start of a cycle. Stuxnet also infects OB35 to create a watchdog functionality and then based on the values found in these blocks, other packets are generated and sent on the wire. From the above description about the Stuxnet functionality we can extract the following facts. The first fact is that Stuxnet or any other virus to attack needs access to communication protocols and as well as to the control logic application. The second fact is that they also need some clues about the technical structure of the targeted system.

Now the question is how we can protect a PLC based system against of virus. The following section is a brief overview on the relevant security standards but as we will see at the end none of them touch the PLC level to provide a security solution.

V. Relevant Security Standards

Every secured computer system must require all users to be authenticated at login time. After all, if the operating system cannot be sure who the user is, it cannot know which files and other resources the user can access. While authentication may sound like a trivial topic, it is a bit more complicated than you might expect [11]. In the case of PLC based systems there is no IT security for logic application (control loop) level as well as for I/O level which exists in regular PC, thus the downloaded logic application on PLC is always running without any privileging, authentication, or security validation process. This means that, the execution of each instruction may raise security deficiencies and cause critical issues. However, there are several standards [3], [8], [12]–[16] that provides a set of rules and procedures to make control systems more secure but none of them touches on the security at the logic application level.

Security standards generally specify what must be done or achieved but not how to go about doing it. In this section, a very brief overview of the most important industrial control security systems is provided. One aspect that is common among all standards is that all of assumed PLCs are in low component compatible level [3], [8], [12]–[16], so they put PLCs out of the security standards scope or at least if they have procedure, is just in operating system level not in application (control logic) level, which makes PLCs more treatable.

ISO/IEC 27001:2005 - ISO/IEC 27002:2005 is addressed in all Industries. IEC 62351:2007 addressed data and communications security and used information security for power system control operations.

IEC 62210:2003 addressed power system control and associated communications - data and communication security electrical distribution. This standard applies to computerized supervision, control, metering, and protection systems in electrical utilities. It deals with security aspects related to communication protocols used within and between such systems and, the access to use of the systems. IEC TC 65 WG 10 IEC/PAS 62443-3-1:2008, addressed by the Electrical distribution/transportation ISA99.

There is an agreement between ISA and IEC by which ANSI/ISA99 standards will form the base documents for the IEC 62443 series. The U.S. Information Technology Laboratory published a guide to Industrial Control Systems (ICS) Security Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC) in 2008 but, even inside of this document there is no procedure for PLC code (logic application) level security [8].

VI. Secure Communication Platform Between Control PC – PLC and I/O

Our solution has two parts. In the first part, the communication link between Control PC and PLC devices must be secured by using a relevant solution like Massage Authentication Code, however, the other extensions of MAC like UMAC will have better functionality in this scope due to distributed nature of these systems. For example, a Control PC can program and configure at the same time more than one PLC so using a multicast authentication protocol can have better effect than single iterative MAC based solution.

Then, in the second part, the I/O communication structure between PLC and sensors/actuators must be secured by our proposed FPGA based solution or by [17] however, as we will see at the end our solution has less overhead.

Finally, in this way we can make a control system end-to-end secure and well protected against any attack from the outside/inside of the control network. However, this needs a hard and complex validations process to make sure that is really functional. This implies that to carry on our idea in the scope of paper we must make some basic assumptions such as use of OT (operational technology) based systems.

A. Part I: Message Authentication Code

Message Authentication Code (MAC) is a method of providing assurance of message authenticity, with the additional benefit of also guaranteeing message integrity [18]. It consists on the sender generating a message key from the message itself (for example, by using a hash algorithm to generate a hash of the message). This key is then cryptographically encoded using a cryptographic algorithm

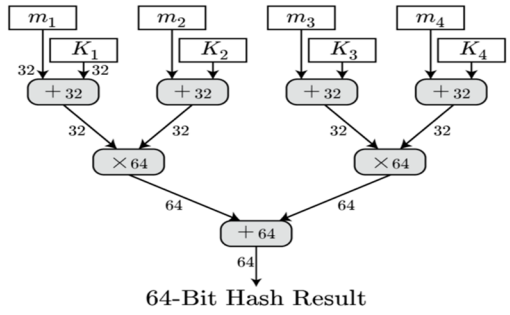

Fig. 4. Universal message authentication code.

and an encryption key. The resulting encrypted hash value (also called the MAC) is added to the message and sent to the receiver.

The receiver verifies the integrity and authenticity of the message by sending the message and the MAC code to a verification algorithm. A trivial solution for the verification algorithm consists on repeating the operations done by the sender and checking whether the two MACs match. Many triplets of the three (hashing, encryption, verification) algorithms may be used. Ideally efficient algorithms are chosen that reduce either the computation time, or the message overhead introduced by the MAC.

1) Universal Message Authentication Code

Universal Message Authentication Code (UMAC) was designed to achieve two main goals, extreme speed and provable security [19]. UMAC works based on dividing the message into m blocks, which allows the hashing and encryption algorithms to be applied to each block independently, and therefore exploiting the capabilities of Single Instruction, Multiple Data (SIMD) parallelism based CPUs. The sender should provide for the receiver the message, nonce, and tag, then the receiver can compute what should be the tag for this particular message and nonce, and see if it matches the received tag (Fig. 4).

2) Real Time Multicast Authentication Protocols

BIBA is a broadcast authentication protocol that takes the first approach, and proposes a one-time signature and broadcast authentication protocol, without trapdoors and relatively small signature [20]. Another method proposed by Reyzin [26] also uses a one-time signature, but manages to be faster than BIBA and has a slightly lower communication overhead. However, both methods are unsuitable for real-time applications due to their still considerable communication overhead. The second approach, which consists of amortizing the signature over several packets, has been adopted by Wong and Lam. This method suffers from high computation and communication overheads. Another protocol, known as TESLA has low computation overhead and low per-packet communication overhead, but does not consider packet loss rate, requires time synchronization between the sender and the receiver in order to satisfy the security condition, and the sending rate must be slower than the network delay from the sender to the receiver. There is another protocol designed by Ritesh Mukherjee [27], this protocol proposed the symmetric message authentication scheme, which is based on symmetric MAC. This protocol consumes large computation overhead. The receiver needs to calculate the MAC of the cipher, make a comparison operation, make a decryption operation, and make another comparison, which may not be practical in case of real time applications.

Finally, there is an another new protocol proposed in [21], [22] by R.Abdellatif, H.K. Aslan, and S.H. Elramly (LAR), which provides authentication but after using of erasure code function that also provides a solution to avoid packet losing problem. It uses both public key signature and symmetric key functions. It is based on the idea of dividing the stream into blocks of m packets. The sender applies the digital signature on the group key kg and the digital signature is done by any public key system like RSA [20]. The output of the erasure code function is partitioned into m symbols: {S1, S2, …, Sm}. LAR avoids the problem of signature loss and sending the signature more than one time and also has a resistance to packet loss as long as it is below a certain loss rate R. The LAR protocol overcomes the pollution attack problem as well as introducing less communication overhead compared to the other protocols used in real time applications. R.Abdellatif made LAR solution even more optimum by processing the protocol as a serial instead of parallel so the complexity of the protocol decreased with less communication overhead by about 2 bytes.

B. Part II: Handling based on FPGA Hashing

Industrial Control Systems, must be connected to physical environments through I/O equipment such as Digital I/O devices, Analog I/O devices etc. In fact, Digital input, and output modules (I/O modules) are key elements of every PLC. Nowadays, Field Programming Gate Array (FPGA) is a well-known solution to design and program such I/O devices [23], [24]. They are easy to use and fixable to merge software and hardware technical concepts. The following section is a representation of MD5 hashing algorithm on FPGA. In this work, we will use a specific type of FPGA from Xilinix products but, this hashing algorithm can be implemented in any type of FPGA. Fig. 5 shows a block diagram of MD5 on FPGA.

In [17] they used an auxiliary processor to implement Elliptic Curve Digital Signature Algorithm (ECDSA) an efficient and secure crypto-algorithm technology [25]. An optimal ECDSA implementation will use public key-based security and a certificate infrastructure along with a digital signature to the authentication process between a PLC and I/O card.

ECDSA involves elliptic curve operations over finite fields, which is a mathematically intensive operation to implement. While the authenticator IC settle on the I/O card, the PLC must also be able to compute a digital signature. This capability increases the complexity of problem for the PLC's host microcontroller. For that in the work [17] they used a coprocessor to overcome this overhead.

But, the problem of the proposed in [17] solution is that the integrity of different modules with each other from different vendors is usually hard or sometimes impossible work. Some companies have already they own products with a PLC from other vendors and I/O modules from their own production line and having a solution based on FPGA can help them to add security layer with minimum cost. The other problem of that solution is the complexity and overhead which implies to use an auxiliary processor. As you can see in our solution we proposed a built in FPGA data structure and a hashing functionality to map the physical addresses with their hashed values and create a secured lookup table for PLC I/O

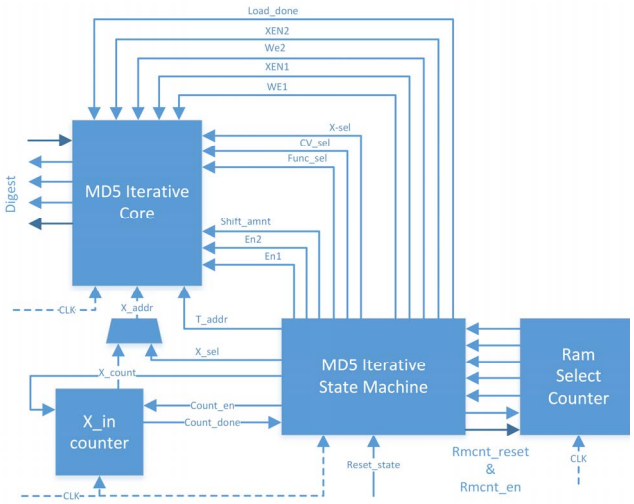

Fig. 5. Block Diagram of MD iterative design.

channels (Fig. 6). PLC will have a serial connection with FPGA and the only fields that will be transferred between PLC and FGPA are Value, Offset and Time-stamp. Since PLC has already a mapped list from physical I/O lists to their hashed values then it will have a grant access to each value and its related signal. The way of processing each signal from I/O and RTDB has been discussed in [1].

VII. Conclutions

We used FPGA to build our I/O device and implement hardware version of MAC encryption with a lookup table to protect signals right after being harvested from the plant. We provide an identical signature per peer of signal tag and value before transferring to the PLC level and also we do integrity test right after receiving a signal from the PLC. This will allow us to make sure about the validity of each signal value before injecting in the control loop and writing back on the output channel. In another word our solution protected the PLC – I/O – PLC part of control system with a very low computation overhead.

References

[1] A. Homay and M. de Sousa, "Multicast Authentication Framework for Distributed Control Systems based on IEC 61499," in Emerging Technologies and Factory Automation, 2016.

[2] R. Marsh, "Critical foundations: Protecting America´s infrastructure," *Comm. Crit. Infrastruct. Prot.*, p. 192, 1997.

[3] M. Abrams and J. Weiss, "Bellingham , Washington , Control System Cyber Security Case Study," p. 36, 2007.

[4] C. G. Billo, W. Chang, Institute For Security Technology Studies, "Cyber Warfare an Analysis of the Means and Motivations of," no. December, 2004.

[5] A. Homay and M. de Sousa, "Message Security for Automation and Control Applications based on IEC61131-3," in *Future Technologies Conference 2016*, 2016.

[6] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet Dossier," 2011.

[7] Siemens, "DCS or PLC, Seven Questions to Help You Select the Best Solution," p. 12, 2007.

[8] K. Stouffer, J. Falco, and K. Kent, "Guide to Industrial Control Systems (ICS) Security Recommendations of the National Institute of Standards and Technology," *Nist Spec. Publ.*, vol. 800, no. 82, 2008.

[9] K.-H. John and M. Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*, Second. New York: Springer.

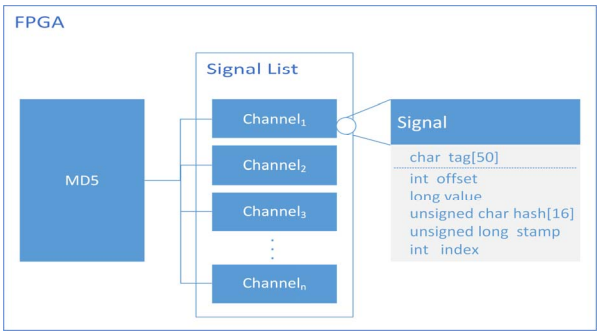

Fig. 6. The I/O lists data structure and MD5 hashing algorithm that used to hash offset.

[10] SIEMENS, "Simatic Net Manual CP 342–5." Nürnberg, pp. 1–12, 2001.

[11] A. S. Tanenbaum, MODERN OPERATING SYSTEMS Other bestselling titles by Andrew S . Tanenbaum Structured Computer Organization , 5th edition Operating Systems : Design and Implementation , 3rd edition Vrije Universiteit Amsterdam , The Netherlands Distributed Operating Syste. Pearson, 2009.

[12] H. Krawczyk, "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)," pp. 310–331, 2001.

[13] RE. Mahan, JR. Burnette,JD. Fluckiger, CA. Goranson,SL. Clements, H. Kirkham,C. Tews, "Secure Data Transfer Guidance for Industrial Control and SCADA Systems," *Rep. to US Dep. Energy, PNNL-20776*, no. September, 2011.

[14] L. Obregon, "InfoSec Reading Room Secure Architecture for Industrial Control Systems.", Oct 2015

[15] T. Phinney, "IEC 62443: Industrial Network and System Security," *(Isa)*, 2006.

[16] A. Shah, A. Perrig, and B. Sinopoli, "Mechanisms to provide integrity in SCADA and PCS devices *," *Int. Work. Cyber-Physical Syst. - Challenges Appl. (CPS-CA '08)*, 2008.

[17] H. SANOGO, "Authentication secures industrial sensor networks," 2015.

[18] F. Draft, "Draft International Standard Iso / Iec Fdis," vol. 2010, 2011.

[19] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: Fast and Secure Message Authentication," pp. 216–233, 1999.

[20] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," *Proc. 8th ACM Conf. Comput. Commun. Secur. - CCS '01*, p. 28, 2001.

[21] R. Abdellatif, H. K. Aslan, and S. H. Elramly, "New real time multicast authentication protocol," *Int. J. Netw. Secur.*, vol. 12, no. 1, pp. 13–20, 2011.

[22] R. A. Abouhogail, "New multicast authentication protocol for entrusted members using advanced encryption standard," *Egypt. J. Remote Sens. Sp. Sci.*, vol. 14, no. 2, pp. 121–128, 2011.

[23] P. C. Reconfigurable, I. O. Digital, and C. Timers, "FreeForm PC104 Reconfigurable Digital IO with Counter Timers," vol. 8979, 2011.

[24] T. Instruments, "Programmable Logic Control (PLC) Solutions Guide," 2015.

[25] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, 2001.

[26] Reyzin, L., Reyzin, R., 2002. Better than BIBA: short one-time signatures with fast signing and verifying. In: Proceedings of the 7th Australian Conference on Information Security and Privacy, Melbourne, Australia, pp.144–153.

[27] Mukherjee, Ritech, William Atwood, J., 2007. Scalable solutions for secure group communications. Computers and Security, Science Direct, 3525–3548