Challenges and Prospects of Communication Security in Real-Time Ethernet Automation Systems

Thomas Müller*, Andreas Walz‡, Manuel Kiefer†, Hans Dermot Doran*, Axel Sikora‡

* Institute of Embedded Systems (InES)
Zurich University of Applied Sciences, Winterthur, Switzerland
Email: {mulh, donn}@zhaw.ch

† SICK AG, Waldkirch, Germany
Email: manuel.kiefer@sick.de

‡ Institute of Reliable Embedded Systems and Communication Electronics (ivESK)
Offenburg University of Applied Sciences, Offenburg, Germany
Email: {andreas.walz, axel.sikora}@hs-offenburg.de

***Abstract*—Real-Time Ethernet has become the major communication technology for modern automation and industrial control systems. On the one hand, this trend increases the need for an automation-friendly security solution, as such networks can no longer be considered sufficiently isolated. On the other hand, it shows that, despite diverging requirements, the domain of Operational Technology (OT) can derive advantage from high-volume technology of the Information Technology (IT) domain. Based on these two sides of the same coin, we study the challenges and prospects of approaches to communication security in real-time Ethernet automation systems. In order to capitalize the expertise aggregated in decades of research and development, we put a special focus on the reuse of well-established security technology from the IT domain. We argue that enhancing such technology to become automation-friendly is likely to result in more robust and secure designs than greenfield designs. Because of its widespread deployment and the (to this date) nonexistence of a consistent securiy architecture, we use PROFINET as a showcase of our considerations. Security requirements for this technology are defined and different well-known solutions are examined according their suitability for PROFINET. Based on these findings, we elaborate the necessary adaptions for the deployment on PROFINET.**

I. Introduction

As a local area network (LAN) technology, Ethernet has become very successful and widely adopted [1]. It became *the* standard network technology for Information Technology (IT) installations in computing centers, office buildings, and private households. The high rate of adoption came with a drop in cost for the hardware, which ultimately motivated the adoption of Ethernet in other, non-IT domains. The field of industrial automation is only one such example where Ethernet and Ethernet-based real-time protocols has conquered the shop-floor level.

In addition to improved cost efficiency the reuse of a well-established technology like Ethernet can bring about a number of both assets and drawbacks. While it helps enable the increasingly desired seamless vertical integration from the business level down to the field level, it also significantly lowers the bar for malicious actors inside the network. It should therefore be understandable that solutions to protect modern automation components from attacks over not any longer isolated communication networks are required.

That said, relying on mature technology also helps avoiding the probability of repeating past technological mistakes; an attitude universally accepted in the security community. We consider this a strong argument for a systematic analysis of the suitability of existing and proven security technology from the IT domain for its deployment in automation systems.

However, industrial automation systems feature requirements that clearly deviate from those of classical IT systems [2]. The most important differences are hard (real-)time requirements, multi-decade device lifecycles, uninterrupted operation, and minimal human maintenance. In other words, just as was the case with Ethernet itself, domain-specific adaptions are likely to be necessary.

In fact, various automation field busses such as Ethernet/IP [3] and OPC UA TSN [4] have considered or even adopted security technology from the IT world. As part of the German research project "Sichere Produktion mit verteilten Automatisierungssystemen (*SEC_PRO*)", the performance of security mechanisms as encryption and message authentication in real-time communication was evaluated [5], [6] as well as concepts for platform integrity, key distribution and a public key infrastructure were proposed [7], [8]. As in the *SEC_PRO* study, PROFINET serves as representative technology in the field of real-time automation systems for our work, although the findings are valid for general real-time Ethernet based field bus technologies.

In Section II we provide an overview of PROFINET as our case study automation system. Section III outlines the requirements and challenges one typically faces when industrial automation systems must be secured. As the main part of our paper, in Section IV we systematically explore the design space of communication security solutions for real-time

Ethernet automation systems. Section V presents a case study applied to PROFINET. In Section VI, an analysis of security considerations made for other technologies in the automation systems market follows. Finally, Section VII summarizes our paper and concludes.

II. PROFINET Overview

PROFINET (**Pro**cess **Fi**eld **Net**work) is an Industrial Ethernet standard for automation systems, originating from the PROFIBUS fieldbus technology and standardised by the PROFIBUS Nutzerorganistion e.V. (PNO) in Karlsruhe, Germany. PROFINET makes use of TCP/IP Communication and other common IT standards, mainly for the configuration and commissioning of industrial control systems and includes a multi-class real-time communication concept. The integration of decentralized peripherals into an industrial control system is realized by PROFINET IO (Input - Output). PROFINET IO defines the communication concept between components within such a system [9] [10].

A. Architecture

PROFINET IO defines three different device classes within their networks [9, p. 34]:

- **IO-Supervisor**: The Engineering Station (ES), usually a PC or HMI (Human Machine Interface) device for parametrization, commissioning and monitoring.
- **IO-Controller**: A Programmable Logic Controller (PLC), containing and executing the automation application programmed from the IO-Supervisor.
- **IO-Device**: A decentralized Input-Output device acting as the slave within an PROFINET IO Network. An IO-Device can either be a sensor or an actuator.

Typically, a PROFINET IO industrial control system contains of one IO-Supervisor, which is usually only temporarily connected, one IO-Controller and several IO-Devices.

B. Relations

The communication between an IO-Controller (or IO-Supervisor) and an IO-Device is organized in virtual channels, so-called Application Relations (AR). An AR is setup after the first connection request, followed by the exchange of communication parameters and device information [10, pp. 78-80]. One AR consists of multiple Communication Relations (CR), a Record Data CR for acyclic standard communication (referred by non-real-time CR in figure 1), an IO Data CR for the transmission of the Input-Output data and an Alarm CR for the acyclic alarm data (both referred by real-time CR in figure 1) [9, p. 42]. Application relations between IO-Supervisor and IO-Device (figure 1, right) can be setup for monitoring or changing device parameters.

C. Protocols

Table I lists the protocols used in PROFINET IO, divided into real-time and non-real-time communication and assigned to the corresponding ISO/OSI layer. The PROFINET IO Services, i.e. the PROFINET stack resides in the application layer. The Remote Procedure Call (RPC, layer 7b in table I) is the first non-real-time protocol, it allows execution of functions in a different address space (e.g. another device). Layer 3 and 4 hold the common standard IT protocols User Datagram Protocol (UDP), Transmission Control Protocol (TCP), used for non time critical data transmission as e.g. commissioning, and the Internet Protocol (IP), Address Resolution Protocol (ARP) as well as the Internet Control Message Protocol (ICMP)[10, pp. 40-43]. The Simple Network Management Protocol (SNMP, layer 3 in table I) is used for collecting configuration information about managed devices in IP networks [10, p. 139-140]. The real-time-class protocols 1 to 3 (and the additional non-real-time protocol over UDP) to transmit IO-Data are defined as follows [10, pp. 48-49]:

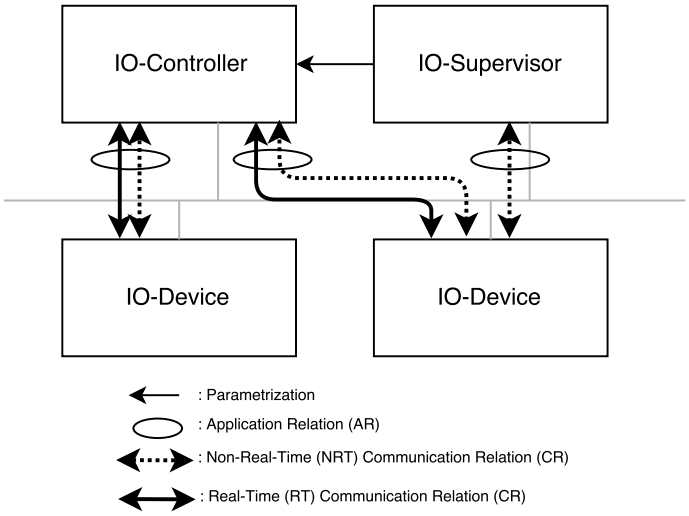

Fig. 1. PROFINET IO system architecture: Example Network consisting of an IO-Supervisor, an IO-Controller and two IO-Devices with corresponding Communication Relations (CR) and Application Relations (AR).

- Real-Time-Class UDP (RT_CLASS_UDP): Non-Real-Time (NRT) IO data traffic over UDP. Typical cycle-times of $100ms$.
- Real-Time-Class 1 (RTC1): Real-Time (RT) cyclic data transmission without any special requirements on switch hardware. No routing. Cycle-times between $5 - 10ms$.
- Real-Time-Class 2 (RTC2): Isochronous Real-Time (IRT) transmission of cyclic data. Special switch hardware required. No topology planning. Cycle-times $< 1ms$.
- Real-Time-Class 3 (RTC3): High-performance transmission. Suitable for motion-control applications. Special switches and prior (topology-based) communication planning required. Cycle times down to $31.25\mu s$.

In PROFINET IO systems, all process events (alarms) are reported using the Real-Time Acyclic (RTA) protocol [10, p. 84]. Prerequisite for time-synchronous real-time communication are synchronized clocks. This is achieved through the application of the Precision Time Control Protocol (PTCP), which is based on the IEEE 1588 standard Precision Time Protocol (PTP) [10, pp. 56-58]. The non-real-time protocols

	OSI Layer	Real-Time (RT)	Non-Real-Time (NRT)
7a		PROFINET IO Services	
7b	Application		RPC
6	Presentation		
5	Session		
4	Transport		UDP / TCP
3	Network		IP / ARP / ICMP / ICMP / SNMP
2	Data Link	RTC1-3 / RTA / PTCP / DCP	RT_CLASS_UDP / LLDP / MRP
1	Physical	100BASE-TX / 100BASE-FX	

TABLE I
PROFINET IO PROTOCOLS.

on the data link layer are the Link Layer Discovery Protocol (IEEE 802.1AB LLDP), used for topology monitoring and neighborhood detection, the Media Redundancy Protocol (MRP) for redundancy in ring topology networks and the Discovery and basic Configuration Protocol (DCP). The latter is used to configure the IP addresses and device names of PROFINET IO-Devices [10, pp. 86-87]. DCP also offers a service to reset PROFINET devices to a factory default state. Some of the DCP services rely on multicast communication. The physical layer relies on 100 Mbit/s Ethernet over twisted-pair cable (100BASE-TX) or optical fibre (100BASE-FX) alternatively (layer 1 in table I).

D. Security Issues

The introduction of Ethernet-based fieldbus protocols in automation control systems has increased performance and efficiency but also led to the appearance of security risks. PROFINET IO is one of the prevalent fieldbus protocols used in industrial control systems based on Ethernet. Such PROFINET IO based networks can be accesible from a companies office network or - under special circumstances - even the world wide web. Besides all advantages that come with this opportunity, this also opens the opportunity for well-known attacks from the IT domain carried out on the field level. In contrast to the office network environment, security weaknesses can be considered to be more severe, since system down times, e.g. caused by denial-of-service attacks, can generate enormous financial and physical damages [10, p. 419]. For this reason and the fact that automation systems can transmit potentially critical or confidential data, a solution to consolidate security in such systems is needed. Security requirements such as authentication, confidentiality through encryption, integrity and availability have to be considered just as well as the limited resources of already installed modules and equipment. To define such specific security related requirements and provide suitable countermeasures, the possible attacks on the different PROFINET IO protocols, i.e. the OSI layers need to be precisely identified.

III. REQUIREMENTS AND CHALLENGES

The requirements on automation systems differ significantly compared to the general IT office environment. Downtimes directly lead to production rejects. The potential resulting financial damage needs to be prevented under all circumstances. These specific requirements are discussed within this chapter.

A. Automation System Requirements

Based on the PROFINET Security Guideline[11], we list the requirements that must be fulfilled after establishing a security concept in the PROFINET and general (real-time) industrial control system environments:

- Availability
- Real-time capability
- Straightforward and cost-efficient commissioning and device replacement
- Long-term operation without human attendance
- Coexistence and interoperability (security-aware and legacy devices)

Availability must remain unimpaired after security mechanisms are implemented and even if an impact on the performance can not be prevented completely, real-time requirements (i.e. strict cycle times) must be met. The integration (i.e. commissioning and replacement) includes the configuration of the appropriate security measures. Minimizing this configuration effort increases the acceptance of the proposed solutions, especially as most automation and maintenance technicians do not have a deep understanding of security. Components of an automation system are designed to operate over long periods, usually without ever being shut down or manually reconfigured. A fitting security solution must adhere to this mode of operation. Thus possible manual actions needed to preserve the required level of security, such as key renewal, shall not interfere with operations. Also due to long-term operation, it can not be assumed that all legacy devices will be replaced with security-aware devices as soon as they are available. Therefore, it must be possible to integrate devices including security mechanisms seamlessly into existing automation systems and operate them in parallel with legacy devices without limitations other than missing security.

B. Security Objectives

Following objectives shall be addressed within a security solution for real-time Ethernet based automation systems:

- Device authentication: Network components (controller, device, engineering station) mutually authenticate themselves to their communication partners, where the claimed identity must be verified.
- Authorization: Ensure, that an (authenticated) communication partner is allowed to perform the intended operation.
- Message authentication: The receiver of a message must be able to verify its integrity (i.e. detect tampering) and if it originates from a known sender. Furthermore, message authentication must prevent the possibility of replayed messages.
- Message confidentiality: The content of a message shall be hidden from possible eavesdroppers that are able to intercept them.

Automation systems may transmit confidential data, i.e. perhaps their disclosure could reveal trade secrets. Nevertheless, encryption to protect systems from this threat causes a significant performance overhead. For this reason, security mechanisms that provide confidentiality should be optionally configurable. In addition to the mentioned security objectives, access to an automation network from the companies' IT infrastructure or the internet should be avoided after commissioning (except during monitoring). Also, physical access to the system should only be allowed to authorized personnel. Such physical security measures are not discussed further within this document, as the focus is on protocol security.

IV. Exploring the Design Space

In this section, we are going to explore the landscape of potential security solutions for PROFINET along three more or less orthogonal dimensions. These are related to the following questions:

1) Which OSI layer(s) is/are most suitable for hosting PROFINET security?
2) Which cryptographic building blocks can be (re)used to provide PROFINET security?
3) Which identities drive entity (e.g., device, controller) authentication and how are corresponding cryptographic keys managed?

A. Stack and Protocol Integration

The Open Systems Interconnection (OSI) computer network architecture not only delivers a common base for designing and implementing networking solutions, it also serves as a foundation for decisions where (i.e. at which layer) to place security services that satisfy the defined requirements. The ITU Telecommunication Standardization Sector has setup a security architecture recommendation (ITU-T X.800 [12], identical to ISO/IEC standard 7498-2:1989 [13]) on the position of specific security services within the OSI reference model. This architecture provides a useful overview of many modern network security concepts [14, p. 27] and will therefore be elaborated precisely respecting the defined requirements for PROFINET.

1) Principles on Layer Integration: First, we define some principles that should be used to determine the allocation of security objectives to layers within the OSI model (compare to section 6.1.1 in [12]):

(a) The number of alternative ways to achieve a security objective should be minimized. Particularly, this should be kept in mind respecting interoperability between devices of different vendors.
(b) Spreading implementations for security services over multiple layers is feasible, but
(c) Violation of layer independence should be avoided, i.e. do not provide any functionality within one layer which is technically covered by a different (upper or lower) layer.
(d) Wherever security mechanisms are dependent on other (security) mechanisms on a different layer, (possible) intermediate layers need to be designed in such way that security violation is impracticable (i.e. avoid covert channels).
(e) Where possible, security functionalities added to a specific layer should be designed as self-contained modules. This means, apart from the integration of such modules, the implementation of the original layer functionality remains unchanged.

OSI Layer							Security Service
1	2	3	4	5	6	7	
-	-	X	X	-	-	X*	Device Authentication
-	-	X	X	-	-	X*	Integrity (Message Authentication)
X**	X	X	X	-	-	X*	Confidentiality

X Implementation of security functionality in this layer feasible.
\- Implementation of security functionality in this layer not recommended.
* Implementation exclusively in layer 7 feasible, but only support of lower layer functionality is recommended.
** Physical layer functionality for confidentiality (e.g. spread-spectrum technology) unfeasible in PROFINET context.

TABLE II
Placement of security services for PROFINET within the OSI model.

2) Placement of Security within OSI model: Table II defines the possible placement of security services within the OSI model. This is a simplified subset, adapted to the requirements for PROFINET security, from the placement recommended by ITU-T (section 7.8 in [12]). All key security requirements defined for PROFINET can be met either on the network, transport or application layer or combinations of them. Confidentiality could also be achieved by solutions located in the data link or physical layer. Within the PROFINET context, confidentiality services within the physical layer are unfeasible, since this would require changes of the Ethernet standard. Implementation of security services exclusively within the application layer, i.e. the PROFINET stack, would need to be developed proprietary from each PROFINET stack vendor and even adjusted to the system-specific needs of the device vendors applying these software stacks. Also, to apply security services in the application layer, actual frame content, i.e. payload, needs to be assembled therein, what is not necessarily the case. Supporting lower layer security functionality from the application layer is recommended. This means for example, instead of directly passing application data to a lower layer, calling standardized interfaces of security mechanisms residing in an intermediary layer.

3) Recommendation for PROFINET Security: Simplifying the adoption of a security concept for PROFINET is significant for success. The effort for stack and device vendors to integrate security measures into their devices (new and legacy) should be kept as low as possible. Also, even more important, this process should not differ between the multiple adopters, to keep the number of alternative solutions minimal (and thus also the possible error sources). While stack implementations vary widely, networking functionality in lower layers (mainly) do not. Working closer to the hardware also brings the advantage of a more seamless integration of hardware acceleration for cryptographic algorithms, which can be crucial to meet strict

timing requirements. Therefore, we recommend the integration of security functionality into the network or transport layer.

B. Cryptographic Building Blocks

When devising security solutions for industrial communication systems, one often finds two major design objectives being in manifest conflict. On the one hand, security solutions shall build on well-established standards that minimize the potential to fail with a solution designed from scratch. On the other hand, almost all such standards were developed in the context of office and Internet IT; domains, whose operational requirements are hardly comparable to those of industrial automation systems. Tight timing constraints in the latter systems may, for instance, easily render any standard solution effectively or apparently inapplicable.

That said, preconceptions regarding the appropriateness of IT standard solutions for industrial automation systems sometimes seem to bias the discussion about a suitable security solution towards custom reinventions. We believe that premature disqualification of IT technology for industrial automation systems is at least questionable. At second glance, the assumptions and requirements imposed by IT standards may turn out to be less rigid than initially assumed.

Moreover is it worth noting that custom re-designs rarely follow entirely new paths, but rather are guided by existing solutions. This tendency would be without problems if the design delta held more potential to feature significant benefits than to introduce security-critical design flaws. The field of security engineering has gone through an extensive learning process with numerous painful but invaluable lessons. Any decision for building custom solutions should therefore be based on very strong arguments against using standard solutions.

If the development of a custom solution should turn out as the only viable approach, it has to be ensured that the concept as well as the implementation will be carefully revised by experts. This implies the need for disclosure of all documents, algorithms and generally the source code of the designed solution. Following Kerckhoffs's principle [15, p. 226], security of a cryptographic system shall never rely on non-disclosure of its technical details.

1) Fundamental Conceptual Elements: At its minimum, protecting PROFINET communication from unauthorized spoofing and tampering requires a cryptographic checksum calculated over relevant data and added to corresponding messages. Note that the checksum's precise implementation (i.e., the localization of its application within the PROFINET stack or the cryptographic algorithm it is based on) is beside the point of the current consideration. Furthermore, observe that encryption (as a means against unauthorized disclosure of data) is an optional feature, which is straightforward to add if cryptographic integrity protection is already in place.

In all practical scenarios, means for supplying authenticated and renewable cryptographic keys to the data protection algorithms are required. A corresponding authenticated key exchange between communicating entities should rely on cryptographic long-term credentials whose possession is the technical embodiment of entity legitimization.

The decision between standard and custom solutions is to be answered on at least two levels. The first level is related to the choice of cryptographic algorithm(s). Given the intricate nature of cryptographic algorithm design and the availability of a wide range of proven algorithms, most likely one resorts to existing cryptographic algorithms.

On the second level, however, the answer might be less clear. Here, the question to answer is whether the selected cryptographic algorithms need to be embedded in a custom protocol, or whether one can make use of standard protocols. As an example, consider the Diffie-Hellman key exchange.

In the following, we are going to provide a systematic analysis of existing standard solutions with respect to their applicability for PROFINET communuication security. We show that standard solutions are not necessarily less efficient than a sound and secure custom solution would be.

C. Identity Management

For communication relying on symmetric cryptography, namely message authentication and symmetric encryption, the two (or more) parties intending to communicate with each other must share the same secret key. Such a key can either be a session key (i.e. valid for a fixed duration or until a connection is closed) or a permanent key. In both cases, access to key material by unauthorized third parties needs to be prevented. Standard IT networks often apply a public key infrastructure (PKI), consisting of certificates (i.e. X.509v3) issued by certificate authorities (CA). Such certificates contain a public key of an entity and are distributed to possible communication partners, which can use this public key to encrypt the traffic for the establishment of a shared secret key. Before starting such an establishment, the validity of a received certificate can by checked by verifying the signatures of the issuer and tracing back through the hierarchy up to the root CA - categorised as trustworthy[14, pp. 131-141]. Such a trace back requires online interaction with the root CA, what is not indeed practicable in usually physical separated automation system networks. In this environment, following design decisions for identity management solutions may be considered:

- Key authority: Who assigns (public or symmetric) keys to a device?
- Key establishment: How is the acutal symmetric key to protect the traffic established?
- Key sharing: Which devices share the same secret (pairwise vs. site- or system-wide)?
- Key storage: How is key material stored within a device to prevent malicious access?

In the automation system environment, all network participants are usually known, i.e. number of devices and their tasks remain the same. Even if a device will be replaced, it inherits the functionality of its predecessor. Therefore, the application of certificates is not benefitial and could be substituted by raw public keys assigned statically to devices (including the

corresponding private keys). Possible authorities to assign a key could be the device vendors at production or the site manager before commissioning. A solution to this - also to the other questions - is highly dependent on a consensus of all involved parties (device vendors, provider of needed configuration tools and site managers i.e. customers). Therefore, no specific proposal has been elaborated within this publication.

V. Case Study: Communication Security for PROFINET

In this section we examine standard communication security mechanisms according their suitability for PROFINET and work out which adaptions are needed to fulfill the requirements mentioned in section III.

A. Standard Network Security Mechanisms

Standard IT security solutions were not designed explicitly for the usage in industrial control system environments. Therefore, advantages and drawbacks of existing well established solutions applied to PROFINET need to be discussed.

1) Network Access Control (NAC): IEEE 802.1X is a standard for port based network access control (NAC). Basically it defines an extension of the Extensible Authentication Protocol (EAP) for the usage within the network layer, named EAPOL (EAP over LAN). A device aiming to get access to a local area network sends an EAPOL request packet to an authentication server (AS) including the credentials. The port of this device remains isolated from the network as long as the authentication request was not confirmed by the AS[14, pp. 161-163]. A modification of the EAPOL protocol - defined in the 802.1AE standard for MAC security (MACSec) - ensures confidentiality through point-to-point encryption of the traffic[16]. This combination seems to provide a valid solution for PROFINET at first glance. In fact, the EAPOL protocol makes use of an own Ethertype and the 802.1AE standard defines additional fields to the standard Ethernet packet what violates the requirement of interoperability with legacy devices. Also, the need of an authentication server within a PROFINET network is not advisable given the challenge to develop a cost-efficient solutions. Finally, the defined cipher-suite AES-GCM[1] is around 3 to 4 times slower than other (hash- or block cipher based) message authentication methods[6].

2) Internet Protocol Security (IPSec): IPSec is a protocol suite for authentication and encryption of traffic sent over a local area network (LAN) standardised by the Internet Engineering Task Force (IETF)[17]. It consists of two protocols, Authentication Header (AH) and Encapsulating Security Payload (ESP), whereby the usage of AH is deprecated since it only provides integrity while ESP provides both integrity and confidentiality[14, p.286]. ESP even supports an authentication-only configuration (see NULL encryption algorithm defined in RFC2410[18]). IPSec can operate in two different modes, the transport mode or tunnel mode. In the transport mode, the original IP header is preserved while in tunnel mode, the added security fields and the header are protected and a new (outer) header is added to the frame. In the latter configuration, the end-points can be addressed directly within the outer IP header and no intermediates (thus also eavesdroppers) are able to examine the original inner header. Establishment and management of key material is realized by the application of the Internet Key Exchange protocol (IKEv2, see [14, pp. 305-313]). While IPSec operates on the network layer (layer 3, table I), PROFINET real-time packets are directly built on the data link layer below. These would require major changes on the IPSec implementation to secure real-time traffic, what is not advisable due to its high complexity. Also, the flexibility to use cipher suites other than standardised is not explicitly addressed.

3) (Datagram) Transport Layer Security ((D)TLS): As its name suggests, Transport Layer Security (TLS) builds upon the transport layer (layer 4, table I) and allows establishing a protected channel between two communication end points. As TLS relies on a reliable transport protocol (i.e. TCP with implicit packet reordering)[19], an adaption for datagram based protocols (DTLS) was designed, where reliability is provided explicitly[20]. During the initial (D)TLS handshake, the communication partners - i.e. server and client - authenticate each other, establish the cryptographic keys and negotiate the message authentication and (optional) encryption algorithms collected in so-called cipher suites (e.g. TLS_RSA_WITH_AES_128_CBC_SHA256, asymmetric encryption of key exchange using RSA[2], symmetric encryption of application data using 128-Bit AES-CBC[3], SHA-256[4] hash-based message authentication code). After a successful handshake, application data from upper layers are captured by the record protocol, where the negotiated cryptographic algorithms are applied before the frames are passed on to the underlying transport protocol[14, pp. 185-189]. Even if technically designed for the use over TCP or UDP respectively, both TLS and DTLS may be used over any transport protocol (though some adjustments to meet its requirements may be needed, as e.g. padding to minimal frame length). For reasons of limited resources or performance, it may be considered to add own cipher suites and even define them as mandatory within the PROFINET context. We therefore rate (D)TLS as a flexible and suitable base for the development of a security solution for PROFINET.

B. (D)TLS for PROFINET

Besides the already mentioned prospects of reusing standard IT security solutions over a custom re-design in section IV-B, using the well-established (D)TLS protocol as candidate for

[1]Advanced Encryption Standard in Galois Counter Mode: A block cipher mode for authenticated encryption, providing integrity and confidentiality [6].

[2]RSA algorithm: A public key encryption algorithm named after its inventors Ron Rivest, Adi Shamir and Leonard Adleman [14, pp. 98-100].

[3]Advanced Encryption Standard in Cipher Block Chaining mode: Each plaintext block is combined with its previous ciphertext block (XOR) before the acutal encryption is executed [6].

[4]Secure Hash Algorithm 2: A cryptographically secure one way compression function producing fixed length output (in this case of 256 bits) [14, pp. 84-88]

Type	Version		Epoch	Sequence Number	Length
	Major	Minor			
(1 Byte)	(1 Byte)	(1 Byte)	(2 Bytes)	(6 Bytes)	(2 Bytes)
Payload (plaintext or encrypted)					
Message Authentication Code (MAC)					

TABLE III
DTLS RECORD FORMAT [20].

adaption to PROFINET brings several advantages (compare to [21]):

- No need to define a new cryptographic negotiation, authentication, and key exchange protocol between communicating devices
- No need to train software developers on how to use new cryptographic protocols or libraries
- Automatically benefit from new cipher suites by simply upgrading to the standard TLS software stack
- Automatically benefit from new features, bugfixes (measures against emerged security weaknesses), etc. in TLS software stack upgrades

The concept of protected communication channels between end-points can be directly mapped to the natural device-pairing approach of PROFINET, i.e. the Application Relations (AR) and Communication Relations (CR) (see section II-B). Nevertheless, there are some challenges to be met in the PROFINET context that may require adjustments on the existing (D)TLS standard:

a) Performance: To meet strict real-time requirements, not only the execution times of the cryptographic algorithms defined within the cipher suite need to analyzed, also the impact on the actual transmission times by the frame overhead (due to additional security specific protocol fields) deserves more careful consideration. Generally, all transmission of useless or redundant information shall be prevented. With DTLS, application data are transmitted within so called records, including header fields for the type of subprotocol (handshake, alert, change cipher spec and record), the protocol version number (e.g. DTLS 1.2), epoch (counter on cipher state changes), a sequence number and the frame length (see table III). While the protocol version could be omitted for the use in PROFINET it also may be thought of validating the frame length implicitly. The lenght of required fields (number of bytes) also needs to be examined and shortened where possible.

b) Channel Multiplexing: The default transport protocols for (D)TLS, UDP and TCP, make use of ports, associated with an IP address, which allows them to manage multiple independent channels. The tupels of port numbers and transport addresses serve as identifiers used by (D)TLS to select the correct connection state (and therefore know how long a session key is still valid). Since Ethernet does not make use of something similar to ports, this feature is missing. Without adaption of the standard, key renewal (i.e. performing a new handshake), blocks the actual channel. This could result in a negative impact on the real-time behavior. To prevent this, an additional field to differentiate between multiple channels needs to be introduced - named e.g. channel ID. Properly implemented, the channel multiplexing can preserve real-time capability and ensures a new key is available when it is needed.

c) Broadcast/Multicast: Some protocols in PROFINET rely in broad- or multicast communication (e.g. DCP). Obviously, protecting this communication does not fit within an end-to-end device pairing. These protocols need to be examined precisely whether their protection is mandatory or this requirement can be waived. It could be considered to only protect security critical messages (for DCP this could be e.g. factory reset or name assignment), which are designed as unicast services. Protecting multicast services too could only be realized by sending the messages to all intended devices separately (using the particular negotiated cipher suite and keys). Evaluation of this possibility in regard to interoperability with legacy devices can not yet be reliably concluded.

d) Preserving of FrameID: The PROFINET real-time protocols (see data link layer in table I) makes use of a frame identifier to be distinguished from each other. In systems with very low cycle times (specially isochronous real-time capable systems), devices may be used that filters incoming frames according to this identifier in hardware before passing them on to the application. To retain the interoperability with such devices, the PROFINET FrameID shall be placed outside of the (D)TLS record and, in particular, prevented from encryption. This requires some adaption to standard (D)TLS since the frame identifier is provided to the record protocol within the application data.

e) Binding to Application: Even if the key functionality (the record protocol) of (D)TLS is located within the transport layer, the change cipher spec, handshake and alert protocols are located above the transport layer and may be integrated in to the stack. This seems to be no big challenge since they function independently of the application. Nevertheless, it may be useful to provide interfaces to the actual PROFINET stack, e.g. to respond to specific alarms. Also, the initial connection establishment needs to be started through the application. Such interfaces needs to be designed as lean as possible to minimize the effort of integration in different stack architectures.

f) Constrained Components: General PROFINET devices (in particular IO Devices) are designed as embedded systems and typically resource constraints (low memory capacity and processing power). This leads to the possibility that a fully featured (D)TLS implementation can not be ported to such devices. To minimize the memory footprint of a (D)TLS adoption for PROFINET, only the necessary functionality must be included, while other shall be omitted (e.g. certificate parser and deprecated cipher suites).

g) Different (D)TLS Instances: Non-real-time traffic as e.g. RPC (over UDP) do not require any performance optimization or other adaptions to (D)TLS. Since they are not time critical, it can be considered to protect them with an even stronger (and therefore more time-consuming) cipher suite than the real-time traffic. Besides that, interfaces to the

application layer could differ for real-time and non-real time protocols, since their paths throught the PROFINET stack vary. Therefore, an approach consisting of two separate instances of (D)TLS, one optimized for performance and usage over Ethernet, the other optimized for communication security of non-real-time traffic (over UDP), could be considered. Using such a setup expects a detailed evaluation whether secrets or connection states shall be shared between these instances or if they need to be treated as completely independent channels.

h) Time Synchronization: In standard (D)TLS, the usual entity authentication is based on certificates, which need to be validated. Since this validation needs a correctly synchronized time to be able to check the expiration date, PROFINET devices adopting this authentication method would require time synchronization. If the defined identity management (see chapter IV-C) solution makes use of this functionality, timing synchronization would become mandatory for security-enabled PROFINET devices. As mentioned, this topic is not addressed further within this publication. Despite this, we recommend to omit any public key schemes depending on synchronized time on devices.

i) Unattended Operation: (D)TLS was originally designed to be operated by human users (in the internet), consequently in case of failure (e.g. invalid certificate, connection error or similar), the user can decide what action he wants to take (e.g. ignore or request support). In PROFINET there is no such human interaction. For this reason, it is mandatory to identify each possible operation and failure state and define which specific action has to be performed in this case.

j) Client/Server Scale Inversion: A (D)TLS handshake for connection establishment is initiated by the (D)TLS client (with a client_hello message). In a PROFINET system, the initiator of a connection is the IO Controller. Applying this to (D)TLS for PROFINET results in one (or at least only few) clients and many servers. This setup maybe unusual, but in fact will have no impact on the scalability or functionality.

k) Rare Interaction with ES: Typically, an automation system is - once programmed and configured - no longer connected to on-site components that are not needed for operation, as e.g. the Engineering Station (ES). As long as the ES is not needed as certificate authority (CA) to verify other components authenticity, such an interaction is not required in any case.

VI. Analyzing the Automation Systems Market

Security for automation systems experienced a boost in recent times. A shared understanding has emerged, that network segmentation for physical separation (so-called demilitarized zones DMZ) are not sufficient to mitigate all attack vectors. Another widespread industrial protocol besides PROFINET is Ethernet/IP, an open standard managed by ODVA[5]. In their security approach, integrity and confidentiality is addressed with the use of TLS and (D)TLS respectively while endpoint authentication can be realized with either pre-shared keys (PSK) or certificates (X.509v3)[3]. OPC UA[6] TSN (Time Sensitive Networking) is a promising, vendor independent successor technology of Ethernet-based fieldbus systems. TSN covers a variety of standards for real-time communication, time synchronization, scheduling and traffic prioritization as well as high bandwidth efficiency, while OPC UA is targeting embedded devices and comes with a security concept also relying on TLS and X.509 certificates [4]. Besides industrial process control, other fields of application as e.g. building automation has to deal with the integration of security critical services. BACnet[7] and KNXnet[8] are communication protocols also (though not exclusively) built on Ethernet for building automation systems. The security mechanisms IPSec, TLS and VPN (virtual private network) were rated respecting their suitability for the use with BACnet and KNXnet. While IPSec requires significant changes to allow the establishment of shared secrets between more than two parties, it is labelled as unsuitable for these standards. Their communication is based on multicasts, what also disqualifies the usage of TLS without adaption. Managing multicast VPN connections on a centralized server is complicated and therefore as well inappropriate for securing building automation systems. For this reason, a security layer addressing the specific requirements of these protocol standards is introduced, strongly influenced by the implementation of TLS[22].

VII. Summary and Conclusion

The increasing level of vertical integration from automation systems into a company's IT infrastructure, specifically by the adoption of Ethernet, has opened them to several attacks with possible effects that cannot be underestimated. Security objectives for solutions to protect such systems that do not violate their existing requirements needed to be elaborated first. Specially prospects on the availability, real-time capability and the usually long-term operation of industrial control systems has represented a particular challenge. A widespread representative of such real-time Ethernet based field bus protocols is PROFINET, which we used to apply our case study on. To date, no specific security solutions besides physical network separation exist for PROFINET, which means we needed to define which security measures we want to achieve. While device authentication and data integrity are mandatory to provide a minimal level of security, confidentiality was rated as optional feature probably needed where trade secrets could be disclosed. Since PROFINET is a standard adopted by multiple device vendors, security mechanisms shall be easy

[5]Open DeviceNet Vendors Association, Inc.: A global standard development organization consisting of suppliers of devices for industrial automation applications.

[6]Open Platform Communication Unified Architecture: A machine to machine (M2M) industrial communication protocol developed by the OPCFouncation.

[7]Building Automation Control Network: A network protocol for building automation systems developed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE).

[8]Konnex Bus: A standardised communication protocol managed by the KNX association, based on three predecessors European Home Systems Protocol (EHS), BatiBUS, and the European Installation Bus (EIB).

and seamless to integrate into existing devices. We therefore elaborated the optimal placement of security mechanisms within the OSI reference model. The resulting statement is, that a placement in the network or transport layer (3 or 4) is most suitable to meet the requirements. We also worked out that it is advisable to consolidate standard IT security solutions for the usage in PROFINET (as well as general industrial control protocols), even if they cannot be applied without adaption. This recommendation is based on the findings, that even if a custom solution would be developed from scratch, this process will be guided by existing solutions and end up in a similar setup, but never inherently equally secure. Given that, we analyzed the suitability of different existing security standards - namely IEEE 802.1X Network Access Control, IPSec and (D)TLS - according their adoption for PROFINET. (D)TLS turned out as the most promising candidate, therefore a deeper look on the specific adaptions which fulfil all the mentioned requirements followed. Mainly, adjustments on the record protocol header to reduce the performance overhead as well as a channel multiplexing functionality to ensure transparent key renewal are necessary changes. Regardless, we are convinced that these effort will pay off, since we can benefit from a well-established, matured solution.

ACKNOWLEDGMENT

The authors would like to thank Fabian Koch, cyber security engineer at ABB Automation Products GmbH, for his extensive review and thorough expert support.

REFERENCES

[1] J. Z. Charles Spurgeon, *Ethernet: The Definitive Guide*. O'Reilly Media, 2014.

[2] E. D. Knapp and J. Langill, *Industrial Network Security*. Elsevier, 2015.

[3] B. Batke, J. Wiberg, and D. Dubé, "CIP Security Phase 1, Secure Transport for EtherNet/IP," in *ODVA Industry Conference*, 2015.

[4] D. Bruckner, R. Blair, M. Stanica, A. Ademaj, W. Skeffington, D. Kutscher, S. Schriegel, R. Wilmes, K. Wachswender, L. Leurs *et al.*, "OPC UA TSN - A new Solution for Industrial Communication," B&R Industrial Automation, Schneider Electric, ABB Automation Products, TTTech Computertechnik, General Electric Company, Huawei Technologies, Fraunhofer IOSB-INA, Phoenix Contact Electronics, Intel Corporation, Bosch Rexroth, Cisco Systems, Hirschmann Automation and Control, Moxa, Kalycito Infotech, Tech. Rep., 2018. [Online]. Available: https://www.moxa.com/doc/white_papers/opc-ua-tsn.pdf

[5] M. Runde, C. Tebbe, and K. H. Niemann, "Performance evaluation of an it security layer in real-time communication," in *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, Sept 2013.

[6] B. Czybik, S. Hausmann, S. Heiss, and J. Jasperneite, "Performance evaluation of mac algorithms for real-time ethernet communication systems," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, July 2013.

[7] M. Runde, C. Tebbe, K. H. Niemann, and J. Toemmler, "Automated decentralized it security supervision in automation networks," in *IEEE 10th International Conference on Industrial Informatics*, July 2012, pp. 1234–1239.

[8] S. Hausmann and S. Heiss, "Usage of public key infrastructures in automation networks," in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012)*, Sept 2012.

[9] M. Popp and K. Weber, *Der Schnelleinstieg in PROFINET*. PROFIBUS Nutzerorganisation, 2004.

[10] R. Pigan and M. Metter, *Automatisieren mit PROFINET: Industrielle Kommunikation auf Basis von Industrial Ethernet*, 2nd ed. Publicis Corporate Publishing, Erlangen, 2008.

[11] "PROFINET Security Guideline," Profibus Nutzerorganisation (PNO) e.V., Karlsruhe, Tech. Rep., Nov. 2013. [Online]. Available: https://www.profibus.com/download/profinet-security-guideline/

[12] "ITU-T X.800 (03/1991) Security architecture for Open Systems Interconnection for CCITT applications," International Telecommunication Union, Geneva, CH, Recommendation, Mar. 1991. [Online]. Available: http://handle.itu.int/11.1002/1000/3102

[13] "ISO 7498-2:1989 Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture," International Organization for Standardization / International Electrotechnical Commission, Geneva, CH, Standard, Feb. 1989. [Online]. Available: https://www.iso.org/standard/14256.html

[14] W. Stallings, *Network Security Essentials: Applications and Standards*, 5th ed., ser. Always learning. Pearson, 2013.

[15] M. Hufschmid, *Information und Kommunikation: Grundlagen und Verfahren der Informationsübertragung*, 1st ed., ser. Lehrbuch : Informationstechnik. Vieweg+Teubner Verlag, 2006.

[16] "Ieee standard for local and metropolitan area networks: Media access control (mac) security," *IEEE Std 802.1AE-2006*, pp. 1–150, Aug 2006.

[17] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 4301, Dec. 2005.

[18] R. Glenn and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec," RFC 2410, Nov. 1998.

[19] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Aug. 2008.

[20] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347, Jan. 2012.

[21] O. Friel, R. Barnes, M. Pritikin, H. Tschofenig, and M. Baugher, "Application-Layer TLS (ATLS)," Internet Engineering Task Force, Internet-Draft draft-friel-tls-atls-00, Jan 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-friel-tls-atls-00

[22] W. Granzer, D. Lechner, F. Praus, and W. Kastner, "Securing ip backbones in building automation networks," in *2009 7th IEEE International Conference on Industrial Informatics*, June 2009, pp. 410–415.