View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by ZHAW digitalcollection

Protecting PROFINET Cyclic Real-Time Traffic: A Performance Evaluation and Verification Platform

Thomas Müller, Hans Dermot Doran

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

***Abstract*—PROFINET is a widely adopted, real-time capable Industrial Ethernet standard, that as other automation system technologies, is subject to an increasing level of vertical integration into company's existing IT infrastructure. This integration exposes automation systems to well-known cyber attacks, which leads to a growing need for suitable security solutions. The challenge in protecting PROFINET automation systems is ensuring the suitability of solutions for use with minimal PROFINET cycle times of 250 µs needed to fulfill high-speed motion control market expectations. We develop a prototype of a transparent security switch, designed to apply protection mechanisms on-the-fly. We use this platform to test an initial implementation of a protection system, present preliminary results and further work.**

I. Introduction

Introducing Ethernet-based fieldbus protocols as, e.g., PROFINET IO[1], into industrial control systems has led to an increase in performance and efficiency but also opened the system for potential cyber attackers, which are now able to perform similar attacks well-known from the office IT environment. Contrarily, the consequences of security weaknesses in automation systems are completely different to them in a company's office network. Denial-of-service attacks causing system downtimes can lead to enormous financial damage e.g., the production of rejects or physical damage to the installation. A possible countermeasure to such attacks explicitly targeting automation systems is integrity protection, i.e., ensuring the message could not be maliciously modified during transmission. Also, services to verify the authenticity of a message are needed to make sure only messages from known and trusted communication partners will be processed. Both of these objectives can be met by including a so-called message authentication code (MAC), subsequently referred by integrity check value (ICV) to avoid confusion with the term media access control in networking terminology, calculated over and appended to the packet. The calculation of such an ICV comes with a significant performance overhead, highly dependent on implementation and platform. This overhead needs to be elaborated precisely in the context of real-time (RT) and particularly isochronous real-time (IRT) transmission in PROFINET systems with low cycle times. Considering motion control applications, we explore the possible impact on high-performance PROFINET systems due to time needed for cryptographic processing as well as the transmission overhead of RT frames extended with protocol fields for security.

Security in PROFINET has already been explored within the scope of a German research project. Besides concepts for platform integrity, key distribution and a public key infrastructure [1], [2], the performance of symmetric and asymmetric cryptographic algorithms for confidentiality (encryption) as well as of hash- and block cipher based message authentication codes [3], [4] was evaluated. These results, collected in a final report [5], were rated according to a working assumption of a PROFINET systems with a configured cycle time of 1 ms. While this assumption may be correct for a wide range of general PROFINET systems, for motion control applications it is not. A body of work on optimizing PROFINET IRT for fast cycle times [6] states typical system boundaries of such applications: 8 to 256 Bytes payload and 250 µs cycle time with a 50:50 real-time to non-real-time traffic duty cycle. The challenges on finding suitable solutions for protecting PROFINET and general real-time Ethernet automation systems as well as corresponding requirements were investigated in [7],

In Section II we elaborate the performance overhead produced by additional protocol fields for protecting PROFINET real-time traffic as well as a brief theoretical background on proposed cryptographic building blocks. Section III describes the ongoing work of the prototypal implementation of a security switch. This switch serves as platform to evaluate performance and shall also represent a base for a device that can be used for verification and validation by device vendors implementing security mechanisms according to a prospective standard. Section IV summarizes and concludes our work in progress and depicts further work planned on this subjet.

II. Security Protocol Extension: Proposal and Performance Impact Analysis

A. Cryptographic Building Blocks

Encryption of the complete message payload to prevent eavesdropping on process data produces a significant performance overhead, especially on resource constraint embedded devices, what applies to most of PROFINET IO-Devices but also to some of the IO-Controllers. In this paper we therefore

[1] Acronym for Process Field Network Input/Output: An Industrial Ethernet fieldbus standard, standardized in IEC 61158, IEC 61784 and by PROFIBUS Nutzerorganistion e.V. (PNO) in Karlsruhe, Germany

focus on solutions to protect the integrity of PROFINET real-time traffic, i.e., an integrity checksum, taking a secret key shared between both communicating parties as input parameter, calculated over the message payload and transmitted together with the original message. The HMAC algorithm is a well fitting candidate for generating such an ICV. It is designed for the usage with any cryptographic hash algorithm, a one-way function applied on an input of arbitrary length producing an output of fixed length.

a) HMAC:Keyed-Hash Message Authentication Code: HMAC is a simple mechanism to use cryptographic hash functions for message authentication. According to its definition in RFC 2104 [8], it is designed to be used with any available cryptographic hash function H without modification but easy replaceability. Authenticity is ensured by a shared secret key k provided as input parameter to the HMAC amongst the actual message. The key k is padded with zero bits up to the number of bits b of the underlying hash function. If the length of the originally negotiated key is greater than b, the key will be hashed once with the same hash function H as used for the inner and outer hash execution. To omit usage of the same derived key k twice within one HMAC execution, it is XORed once with the inner padding block $ipad$ (0x36 repeated for block length b) and once with the outer pad $opad$ (0x5C repeated for block length b). Both XOR ($\oplus$) operations result in flipping another set of half of the bits in k. The inner hash function gets the message concatenated ($||$) to the key k XORed to the inner padding block $ipad$ as input. Its output then is appended to the result of k XORed with $opad$ Eq. 1 [9, pp. 88-91].

$$HMAC(k,m) = H\Big[(k \oplus opad)||H\big[(k \oplus ipad)||m\big]\Big] \quad (1)$$

The size of the output of the HMAC function is the same as of the underlying hash function H, although this can be truncated. To still meet an adequate level of security, the minimal length of the truncated output is limited to half of the hash output size and shall not be less than 80 bits (compare to Section 5 in [8]). Considering a minimal impact on the performance, truncation shall be taken into account when protecting PROFINET real-time traffic.

b) SHA-3: Secure Hash Algorithm 3: SHA-3 is chosen as candidate to be used as cryptographic hash function in HMAC. Under the name Keccak, this algorithm was selected as the winner of the NIST[2] hash function competition and therefore standardized as SHA-3 in the FIPS[3] Publication 202 [10]. SHA-3 is based on a completely different mathematical structure as its predecessors SHA-2 and SHA-1. SHA-1 is known to be vulnerable against collision attacks and is therefore not recommended to be used in new designs [11]. Although these weaknesses do not affect the security of SHA-2 to the present date, emerging issues can be expected in the future since it shares the mathematical structure of SHA-1. Especially in the automation systems environment, where devices are designed for long-term operation and will not be updated or rather replaced frequently, we therefore recommend to choose SHA-3. SHA-3 is standardized for four different output sizes, 224, 256, 384 and 512 bits (detailed description of mathematical operations of SHA-3 in [10]).

The ICV relies on a shared secret, whose presence is assumed as given in a first iteration. The establishment of this key material is out of scope of this publication.

B. Protocol Fields

For high-performance isochronous real-time transmission in PROFINET, the real-time protocol RTC3 (Real-Time-Class 3) is used, which enables cycle times in the sub millisecond domain. To achieve this a special switch hardware is required [12, p. 49]. The RTC3 protocol consists of the following fields, encapsulated as payload in a standard 802.3 Ethernet frame: (1) FrameID (2 bytes); (2) RT payload (padded up to min. 40 bytes); (3) the ADPU (Application Protocol Data Unit) represented by the Cycle Counter (2 bytes), Data Status and Transfer Status (1 byte each) [13, pp. 100-101] [12, pp. 62-66].

The minimum size of an Ethernet frame is defined by 64 Bytes, including header fields as source and destination MAC addresses, Ethertype (0x8892 for PROFINET) and FCS (frame check sequence, a 4 byte cyclic redundancy checksum). In RTC3, there is no VLAN Tag needed since prioritization is handled with configured duty cycles [12, p. 66]. Therefore, available IO data payload is 40 bytes. These fields must now be extended by the fields to ensure protection. To minimize the transmission overhead on the wire, only important fields of minimal size shall be added. Starting with the necessary ICV, we can choose a HMAC with SHA3-224, truncated to half of its output (14 bytes) to have a minimal sized protocol field providing sufficient security. To prevent replay attacks, i.e., the malicious retransmission of an already sent packet, a sequence counter is needed. The PROFINET RTC cycle counter represents the relative transmission time in multiples of $31.25\,\mu s$. This may appear to fulfill the functionality of a sequence, unfortunately, the cycle counter is only 2 bytes long, which results in an overflow every $2^{16} \cdot 31.25\,\mu s = 2.048\,s \sim 2\,s$. For this reason, an extension of the cycle counter has to be introduced within the security fields consisting of 2 additional bytes, incremented on every overflow of the PROFINET cycle counter. Otherwise, keys would have to be refreshed each 2 seconds to still be able to prevent replay attacks. The additional 2 bytes extends the expiration period to $2^{16} \cdot 2.048\,s \approx 37.3\,h \sim 1.5\,d$. Performing a renegotiation of keys every day is feasible, what could be scheduled to times of low utilization (e.g., at midnight) to ensure that the system will never be disturbed in its normal operation. Nevertheless, it also needs to be ensured that the establishment of a new key can never interfere the cyclic data exchange. Therefore, the key negotiation needs to be scheduled to be performed before expiration of the actual key, i.e., at each time of operation, there needs to be at least one new key available to replace the current. This leads to

[2] National Institute of Standards and Technology

[3] Federal Information Processing Standards

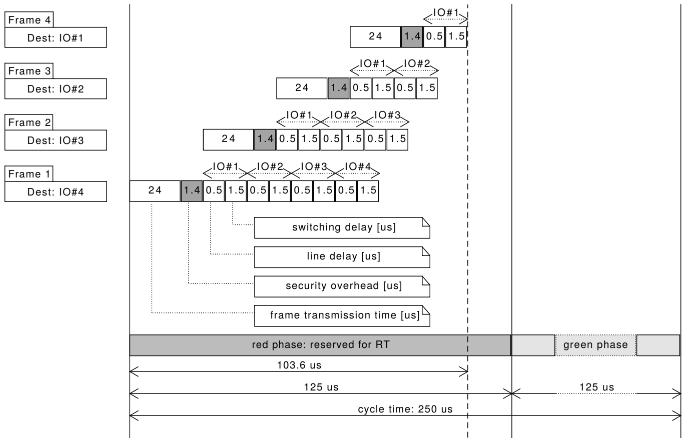

Fig. 1. PROFINET IRT frame scheduling in IO-Controller.

the introduction of another mandatory field: An identifier which key context is applied to the currently processed frame. Again, to keep the overhead as small as possible, this field can consist of a single byte. Assuming a key refreshment cycle of one day, this identifier overflows every 256 days. We presume that it will be never necessary to keep more than 256 key contexts available within one connection. All these fields collected result in a security overhead of 17 bytes. This will be our working assumption for the following calculations.

C. Performance Considerations

To analyze the performance, we defined the following system boundaries corresponding to a worst case configuration (i.e., upper limit of IO data payload) of a motion control system running PROFINET IRT:

- Cycle time: 250 µs, 125 µs reserved for RT traffic and 125 µs open for NRT traffic (50:50 duty cycle).
- IO data payload size: 256 byte.
- Ethernet bandwidth: 100Mbit/s, full-duplex.
- Wire length between two network devices: 100 m.
- Switching delay: 1.5 µs [6].

The minimum time of 125 µs open for non-real-time traffic originates from 123 µs transmission time for a maximum sized Ethernet frame of 1538 bytes on the wire (1500 bytes payload, 18 bytes header as described in Section II-B and 20 bytes for preamble, start of frame delimiter and interframe gap), which is 123 µs in 100Mbit/s Ethernet. A RTC3 frame with 256 bytes payload results in a frame of 300 bytes that will be transmitted within 24 µs. Without security extensions, 5 such frames could be transmitted within the reserved RT bandwidth, but the defined 17 bytes of additional protocol fields produce a transmission overhead of 1.4 µs (Eq. 2). This leads to a physical limitation of 4 RTC3 motion control frames to be transmitted within 125 µs (Eq. 3).

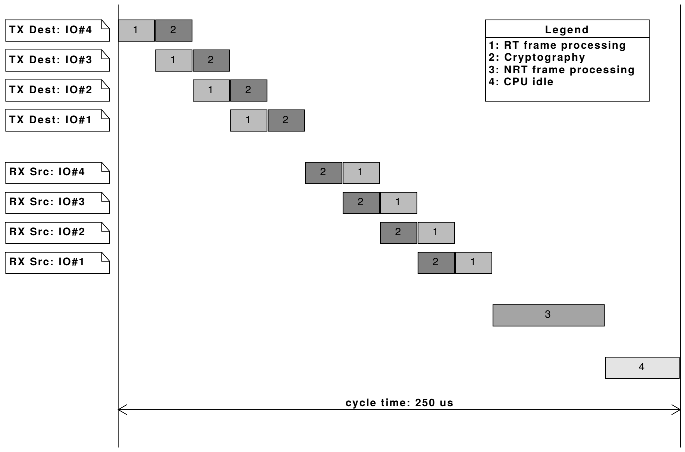

Fig. 2. PROFINET IRT task scheduling in IO-Controller: RT frame processing (task 1), cryptographic protection of frames (task 2), non-real-time frame processing (task 3) and CPU idle time (task 4).

Task	Execution Time by RT Payload Size [µs] (Average over N=1000 Iterations)	
	8 Bytes	256 Bytes
SHA-3-224	14.53	26.02
HMAC-SHA-3-224	19.27	28.41
RT Frame Processing	0.23	0.54

TABLE I
PERFORMANCE MEASUREMENTS OF PURE SHA-3 AND HMAC-SHA-3 (224-BIT VERSIONS) IMPLEMENTATIONS AND THE RT FRAME PROCESSING OVERHEAD ON INTEL CORE I7 (3.4 GHZ).

$$\frac{(17*8)bits}{100*10^6\,\frac{bits}{s}} = 1.36\,\mu\text{s} \approx 1.4\,\mu\text{s} \tag{2}$$

$$\left\lfloor \frac{125\,\mu\text{s}}{(24\,\mu\text{s} + 1.4\,\mu\text{s})} \right\rfloor = 4 \tag{3}$$

The scheduling of an IO-Controller serving 4 IO-Devices with one RTC3 frame per cycle can be seen in Fig. 1. The first frame to be sent targets the furthermost IO-Device, because in addition to its switch it has to pass the 3 intermediate switches (1.5 µs delay) and 4 line delays of 0.5 µs [13, p. 113].

Each frame transmitted by the IO-Controller must be processed within the stack preliminary. Subsequently, the cryptographic protection of the frame is performed. The same tasks needs to be applied in reverse order for the received frames of each IO-Device, i.e., the integrity needs to be verified before the frame can be processed further. This leads to a critical path of 8 times the processing time of cryptographic protection t_C (assuming a symmetric setup, i.e., IO-Devices are configured to provide and consume one frame within every cycle) and 2 times the needed RT frame processing time within the PROFINET stack t_{RT}, having both tasks implemented to run independently. Additionally, some time to process non-real-time application tasks (t_{NRT}) and CPU idle time (t_I) is scheduled (Fig. 2). A proof of concept implementation of a PROFINET RT protection routine on a

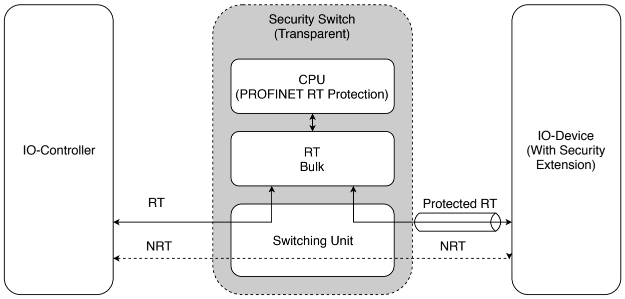

Fig. 3. PROFINET Security switch architecture.

workstation machine (intel Core i7, 3.4 GHz) delivers some reference values for high-performance platforms for further estimations: $t_{RT} = 0.54\,\mu s$ and $t_C = 28.41\,\mu s$ (see Table I). Using these values, 21.64 µs remain for non critical tasks ($t_{NRT} + t_I$) besides the high priority RT processing (Eq. 4).

$$250\,\mu s - 8 \cdot t_C - 2 \cdot t_{RT} = 21.64\,\mu s = t_{NRT} + t_I \qquad (4)$$

III. Security Switch Architecture

To keep the effort for integration of security mechanisms into PROFINET devices as low as possible, the processes for testing and verification of implementations shall be applied equally by different device and stack vendors. For this purpose, a transparent switch that is capable to manage a secure communication relationship between a legacy, security-unaware IO-Controller and an IO-Device with security extensions, was designed (Fig. 3). The transparent switch is based on an industry proven three-port switch optimized for PROFINET IRT communication supporting intelligent, dynamic traffic filtering and fast frame forwarding [6]. NRT frames will be forwarded as usual, while RT frames are passed on to PROFINET RT protection application within the CPU, where they are converted to protected frames by adding the necessary field (as described in Section II-B), or vice-versa (i.e., verification of integrity and back-conversion to standard RT frame). This security switch is implemented on a standard FPGA development board (Xilinx Zynq-7000 with ARM Cortex-A9 processing system), which provides an optimal platform for performance evaluation of an initial implementation of protection mechanisms for PROFINET RT (this includes IRT) communication, both purely software-based and hardware accelerated.

IV. Conclusion and Further Work

Focusing on integrity and ensuring authenticity, we propose a scheme for the protection of PROFINET RT traffic based on HMAC-SHA-3, producing a protocol overhead of 17 bytes in total. With a proof of concept implementation of the protection mechanisms we can show that it is theoretically feasible to provide security for high-performance motion control systems, even if the transmission of the additional fields limits the maximum number of IO-Devices to be served by an IO-Controller within the 125 µs bandwidth for RT traffic to 4. A concept for a transparent switch as inline unit for managing a secure channel between security-aware and unaware devices, is presented. This prototype is used to analyze the performance of the proposed solution and shall provide a base platform for a generic, vendor independent security testing and verification device. Amongst the state-of-the art process of outsourcing cryptographic algorithms into FPGA fabric, approaches to apply hardware acceleration to protocol functionality will be analyzed. Using profiling techniques, the system bottleneck can be identified and the potential for optimizations can be evaluated. Also, a statement about actual time left for protection mechanisms and if around 22 µs is enough time for execution of other, not real-time relevant tasks can be made thereof. Further investigations on optimization during compile time will be performed to elaborate the suitability of purely software-based solutions for high-performance applications.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] M. Runde, S. Hausmann, C. Tebbe, B. Czybik, K.-H. Niemann, S. Heiss, and J. Jasperneite, “SEC_PRO : sichere Produktion mit verteilten Automatisierungssystemen,” Fakultät I - Elektro- und Informationstechnik, Tech. Rep., 2014. [Online]. Available: https://serwiss.bib.hs-hannover.de/frontdoor/index/index/docId/499

[6] D. Gunzinger, C. Kuenzle, A. Schwarz, H. D. Doran, and K. Weber, “Optimising PROFINET IRT for Fast Cycle Times: A Proof of Concept,” in *2010 IEEE International Workshop on Factory Communication Systems Proceedings*, May 2010, pp. 35–42.

[7] T. Müller, A. Walz, M. Kiefer, H. D. Doran, and A. Sikora, “Challenges and Prospects of Communication Security in Real-Time Ethernet Automation Systems,” 2018 IEEE International Workshop on Factory Communication Systems Proceedings, June 2018.

[8] H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” RFC 2104, Feb. 1997.

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

[10] “SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions,” National Institute of Standards and Technology / Federal Information Processing Standards, Gaithersburg, Maryland, Standard, Aug. 2015. [Online]. Available: http://dx.doi.org/10.6028/NIST.FIPS.202

[11] Q. Dang, E. B. Barker, W. E. Burr, S.-j. Chang, L. Chen, D. F. Dodson, M. Dworkin, J. Kelsey, R. Perlner, W. T. Polk, and A. Regenscheid, “SP 800-107. Recommendation for Applications Using Approved Hash Algorithms,” Gaithersburg, MD, United States, Special Publication, Aug. 2012.

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

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