Industrial Protocol Security and Control System Performance: How Hardware Techniques Can Mitigate Performance Impacts of Cryptography

Jack Visoky MSc*, Warren Johnson

* Corresponding Author. Email: jmvisoky@rockwellautomation.com

Synopsis

Industrial communication protocols have historically lacked the basic security mechanisms standard in Information Technology (IT) and Internet environments. With the increasing prevalence of cyberattacks on industrial control systems, standards bodies have begun to add security mechanisms to their protocols. The list of protocols that have added significant security mechanisms is diverse and growing, including OPC UA®, EtherNet/IP™, and more. This is an overwhelmingly positive accomplishment that significantly reduces the common vulnerabilities and exposures of industrial systems. However, defining the cybersecurity protections in a standard is one thing, challenges remain for the benefits to be realized in real world applications. A major challenge is the development of Operational Technology (OT) hardware capable of achieving high-performance, yet secure, communication which will adhere to that necessary deterministic Input/Output (I/O) monitoring and control requirements of OT applications. This paper explores the current landscape of security protocols as well as advancements in OT hardware.

Keywords: Cybersecurity, Industrial Protocols, Security Hardware, Cryptography

1. Introduction: Moving away from mechanical propulsion

Cyberattacks have caused significant losses in productivity and operational availability for manufacturers, businesses, local utilities, and city/state infrastructure services. For the world's navies and commercial ship owners these security breaches have even greater impacts and dangers such as the loss of a ship's manoeuvrability, electric plant, stability or ballast systems, and damage control; all of which risk the safety of crewmembers and significant damage to the vessel itself, up to and including complete loss of the vessel. Insecure communication protocols represent a vulnerability that can be exploited by attackers to gain entry to the network or to modify important command and control data, causing damage to equipment. While network protections such as firewalls can be applied regardless of the inherent protocol security and can be a significant protection of networks, insecure protocols' session, port, or transport layers can be easily traversed by malicious users which cannot be distinguished from authentic devices and users.

Over the last decade, cyberattack incidents have increased across all sectors of the economy. Across shipyards and other critical infrastructure sites, cyberattacks have gone up tremendously. A recent Check Point Research study shows that cyberattacks increased by 50% year over year from 2020 to 2021 (Stealth Labs, 2022). Those attacks reached an all-time high in December of 2021 to approximately 925 per week per organization. In a prescient warning to such events, the International Maritime Organization and other governing bodies have issued guidelines for cyber risk management under MSC-FAL.1/Circ.3 and the National Institute of Standards and Technology (NIST) Risk Management Framework for necessary activities to implement security controls and practices in maritime operations (International Maritime Organization, 2017). Hardening and securing the ship's network data communications is one guideline cited for implementation.

Although historically industrial protocols have been lacking in even basic security protections, many advancements have been made in several protocols to provide robust cybersecurity protections. Those early communication protocols utilized in OT applications, both deck machinery and propulsion plant engineering rooms, were developed as proprietary protocols from the individual OT vendors themselves. Such protocols focused more on data throughput and data availability versus data integrity and confidentiality. In these cases, a Cyclic Redundancy Check (CRC) or other simple method of transmission confirmation sufficed. Today's OT communication protocols have transformed away from that earlier philosophy. Those same automation vendors have supported industry standards bodies in leading the effort to define open and standard interoperable protocols and to further the development and adoption of secure implementations of those protocols. EtherNet/IP and OPC-UA are examples and are adopted by OT automation products across a vast range of machinery and plant equipment applications. This paper will explore some of those developments, although the scope will be limited to the data protection properties, commonly provided by cryptographic algorithms for information assurances like data confidentiality and data authenticity. Other properties such as user authentication and authorization, logging and non-repudiation will not be covered in this paper.

 https://doi.org/10.24868/10711

2. Protocol Overview

EtherNet/IP has a couple of connotations; the first being its IP network layer representation, familiar to IT Technology users, and the second is the Common Industrial Protocol (CIP) application layer, governed by the ODVA consortium of automation manufacturers, familiar to OT Technology users. As such, it has grown in popularity to be one of the fastest growing protocols worldwide by node count (Carlsson, 2022). EtherNet/IP uses both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) transport layers to send messages either point to point, i.e., unicast, and one to many, i.e., broadcast and multi-cast. On top of that transport layer resides CIP, which is an object-oriented application layer protocol. The CIP message character is defined by the message's syntax of objects. CIP messages are objects with attributes, behaviours, and rules leading them to have tremendous flexibility handling all kinds of data types and message properties for broad use across OT applications in safety, motion, time synchronization, energy, and security. Although somewhat of a simplification, CIP has two general communication mechanisms, one being a request-response generally termed "CIP messaging", and the other being a cyclic data exchange, often termed "Class 0/Class 1", or simply "I/O".

Many robust security protections are available for CIP and EtherNet/IP. Data security protections in the CIP application layer protocol, referred to as CIP Security, is applied at the transport layer utilizing the security principles of the open industry standard Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols. TLS and DTLS are widely used and available technologies that protect much of the Internet-based communications used today (Google Transparency Report). TLS and DTLS security methods employ cryptographic nonces and encryption keys along with AES or Hashed Message Authentication Code (HMAC) cipher algorithms essential to mitigating most categories of cyber threats related to I/O and communication message traffic on the industrial and shipboard OT networks. Those threat categories include spoofing, data tampering, repudiation, information disclosure, denial of service, and elevation of privilege (Overview of CIP Security, 2020). This paper won't delve further into the definition of those threats, a review of them will be found in the referenced ODVA PUB00319R1 (Overview of CIP Security, 2020) document at the end of this paper. The mitigation of these threats requires computationally intensive operations on the OT hardware due to the cryptography used for the information assurances. Whether the cryptographic cipher used is Advanced Encryption Standard Cipher Block Chaining (AES-CBC), Secure Hash Algorithm (SHA-256), Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic Curve Diffie-Hellman Exchange (ECDHE), or others, each cryptographic method requires intensive hardware processing resources to implement quickly and efficiently for each message type.

OPC UA (Open Platform Communication Unified Architecture) is an open, standard protocol that is used for data exchange in the industrial space. One of the strengths of OPC UA is the focus on interoperability, allowing disparate products from diverse vendors to communicate and exchange data. OPC UA has a rich set of features and is augmented by the many companion specifications that provide guidance on how to use OPC UA in specific applications.

OPC UA provides many features for data exchange and communication. Two mechanisms that can be used are the "Client-Server" communication and the "Publish-Subscribe" (often abbreviated as "PubSub") communication. Although there are some nuances to each, from a high-level Client-Server is similar to CIP messaging, and PubSub to CIP I/O.

Security is one of the strengths of OPC UA, with the protocol supporting many cybersecurity protections. In terms of data protection, OPC UA allows for a few different options depending on the encoding being used. There is an option to use an Hyper Text Transfer Protocol Secure (HTTPS) transport, as well as an option for using TLS via Websockets. The OPC UA binary encoding over TCP allows for the OPC UA Secure Conversation is a popular option. All of these options are fairly similar in the types of information assurances provided for data confidentiality and data authenticity. For simplicity's sake, and because of popularity, this discussion will be limited to the OPC UA Secure Conversation, although from a general standpoint it would apply to the other two mappings as well (OPC Specification Part 2: Security Model). OPC UA Secure Conversation is quite similar to the TLS and DTLS protections utilized by CIP Security. Many of the cryptographic algorithms used are either very similar or identical, and the mechanism in which they are applied are also quite similar. Although important differences exist, for the purposes of the general investigation of this paper these can be thought of as providing similar protections by similar means.

Impact of Cyber Security Algorithms

 https://doi.org/10.24868/10711

To understand the impact of cryptography on industrial protocols, it is instructive to begin with a brief analysis of the common cryptographic operations used. Common to both OPC UA and EtherNet/IP, as well as many other protocols, is the idea of some authentication and key agreement operations at the start of the connection, followed by data authenticity and data confidentiality on the subsequent packets. For connection establishment, at least some form of asymmetric cryptography is used for authentication and key agreement. Once this is completed the data exchange portion follows, which mainly or exclusively relies on symmetric cryptography. Although performance impacts can occur in either phase, the data exchange is generally more sensitive to performance degradation since that is where real time data is being exchanged and acted upon. Consider the example of a controller and a drive, shown in figure 1

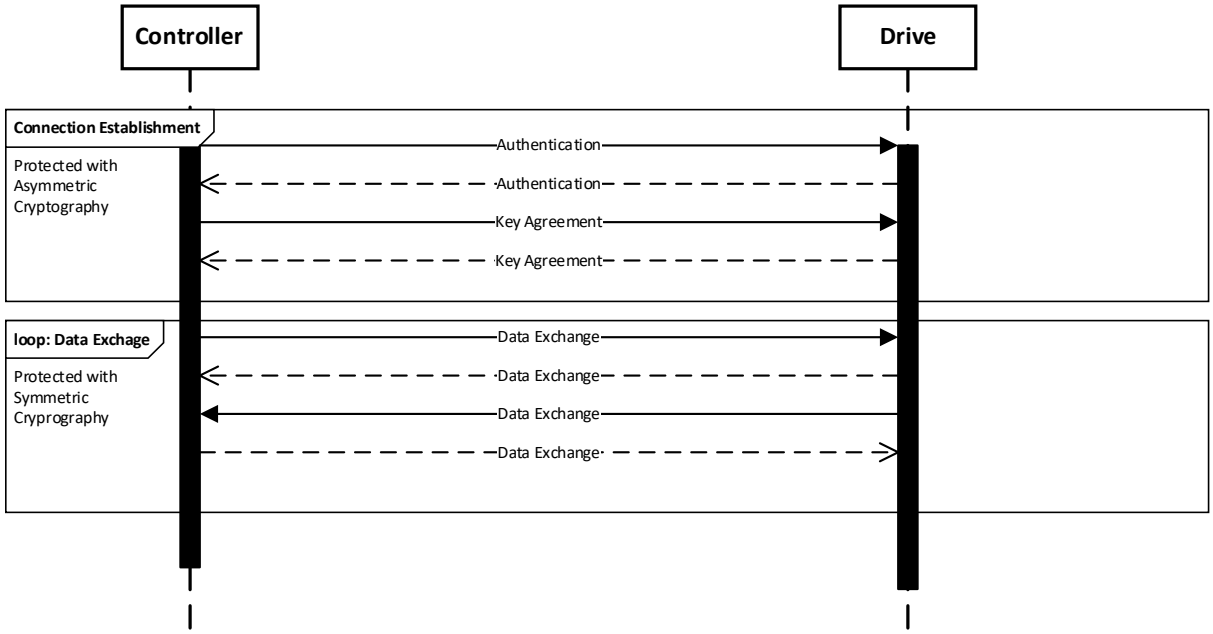

Figure 1: Connection Establishment and Data Exchange for Controller to Drive communication

In this case the connection establishment phase might occur once for a data exchange phase that lasts for days, weeks, months or even years. Taken through this view, performance degradation within the data exchange phase is likely to have a higher impact on the system than degradations seen during connection establishment, especially given that data exchange may occur at a high rate of new data every millisecond or even every hundred microseconds. Therefore, analysis will focus on the performance impact of the symmetric algorithms commonly used during the data exchange phase. An analysis of performance impact for these algorithms will help to set expectations for applying security to both messaging and cyclic communication of various protocols.

Although there are many options for symmetric algorithms, the most commonly used ones centre on the AES family for data confidentiality (encryption) and the SHA-2 family for data authenticity. Variations of each of these exist, although each variation is still based on the same core algorithm. Within the AES cipher algorithm, the AES process consists of multiple rounds of mathematical transformation. It begins with the original symmetric encryption key and the plain text data itself whereby:

1. Using the Rijndael key schedule algorithm, a key expansion is performed that derives a series of new round keys
2. Applying the XOR additive instruction, each round key is combined with the plain text data within the data's stored 4 x 4 array of 16 bytes
3. A Substitution Table is then applied to the result of step 2 which substitutes data within that 4 x 4 array.
4. The 4x4 array rows are then shifted.
5. Finally, another algorithm is applied to mix the 4x4 array's columns.

At the end of Step 5, the first Round has been completed. These steps are repeated after the first Round successively until the final round set by the AES type encryption policy: for AES-128 there are ten Rounds, for AES-192, twelve Rounds, and for AES-256, a total of fourteen Rounds (Crawford, 2019). A good measure to understand an impact to performance from a high-level perspective is to define a theoretical computation

 https://doi.org/10.24868/10711

complexity relationship to these algorithms. Specifically, this computation complexity can be viewed through Big O Notation (Knuth, 1976). Big O Notation defines the limiting behaviour of a function or relationship between two metrics such as whether there is an upper bound or that the relationship approaches a certain number like a constant or infinity. Since AES is a block cipher that relies on an S-Box construction (Daemen, 1965), and SHA-2 is a hash algorithm with a Merkle-Damågard construction (Rachmawati, 2018), the implications of these are that the computational complexity of these algorithms is a function of the input message size and. Therefore, time complexity in Big O notation for both AES and SHA-2 is linear:

$$O(n)$$

A positive implication of this is that as the packet size grows the operations necessary for security grow along with it linearly. Therefore, it is not important to be overly concerned about packet size, at least from the security performance standpoint. Computational complexity is important in understanding what type of resource usage an algorithm requires for a given input. However, there are other factors to consider. Note that even in a linear algorithm, multiplicative constants are ignored, as these constants will not impact the growth of resource usage. However, practically these constants are still important in understanding the "real-world" impact of an algorithm on performance. As a straightforward example, imagine a cryptographic algorithm that always added 10,000 assembly operations for every byte of data processed. This would still be linear, that is

$$O(n)$$

However, in this case the multiplicative constant is 10,000, which likely has a large impact on the processor. Imagine this being used in high-speed cyclic communication, which are very sensitive to latency disturbances. This might add hundreds of milliseconds to each packet that is processed, which in some applications would render the system unusable when this cryptographic algorithm is applied. In other words, computational complexity is important but is not the final consideration.

Beyond looking at computational complexity, which is a useful, albeit somewhat theoretical measure. It is also important to look at a more "real world", or experimental, measurement of the impact on performance. What matters most is what type of performance degradation a given cryptographic algorithm will cause in a given application. Of course, this brings in many other variables such as network architecture, processor design, software layers such as operating systems, etc. Given all of these variables any attempt to measure performance will not necessarily generalize to all potential cases. Nevertheless, it is still instructive to gather some data on this from various sources to get an idea of possible impacts on performance.

As an example of an attempt to measure "real world" performance of cryptographic protections in industrial protocols, one can look to a recent paper "Bottleneck Identification and Performance Modelling of OPC UA Communication Models" (Burger, 2019). In this paper, some measurements were taken around OPC UA client-server communication using a Raspberry Pi Zero. The test cases investigated involved using no security versus using the OPC UA security policy "Basic256Sha256", which uses AES 256 and SHA256. Here measurements were taken regarding CPU utilization. Results varied depending on how much data was being exchanged and how many clients there were, but in some cases CPU utilization jumped by about 10% when encryption was applied. This might not seem very significant, but even an increase of 10% can be quite significant in a system that is already running at close to capacity. Furthermore, it is important to keep in mind that this experiment was run using OPC UA client-server communication, rather than the cyclic PubSub where packet throughput is likely to be much higher.

3. Experimental Data

Some experimental data regarding performance of OPC UA with security protections was gathered through Rockwell Automation's Research and Development labs and is presented and analysed as part of this paper. Several commercial and open-source stacks were run in a Linux environment using Linux Cent OS 7. Note that tests were also run in a Windows environment but resulted in nearly the same results, so for simplicity just the Linux results are displayed and discussed. For the purpose of reporting this data, the intention is to show the effect of cryptography, not to make a comparison between the various stacks. Therefore, it is not instructive to compare performance of one stack versus another, rather the difference when security is added is the important aspect. Note that different stacks have different structure and features, therefore wide differences between the stacks might be observed. In order to prevent the reader from drawing conclusions regarding a given stack being "better" the data has been anonymized. That is, stacks are labelled with generic numeric names, essentially "Stack 1" through "Stack 5".

Each stack was evaluated in the same way. For this evaluation, OPC UA client-server communication was used, with the TCP transport and binary encoding. Naturally, OPC UA Secure Conversation was used as the security protection. Four OPC UA client-server operations were run: Read, Write, Create, and Delete. These operations were performed under an information model with 1,000 nodes and one with 5,000 nodes. This was done with security policy set to none, as well as with security policy set to sign and encrypt, which involved

 https://doi.org/10.24868/10711

applying AES encryption and SHA HMAC to the OPC UA packets. The "Basic256Sha256" security policy was used, which applies AES-256 and SHA-256, except for one of the stacks in which Basic128Rsa15 was used, where AES-128 and SHA-1 is used. Figure 2 shows the raw results of this experiment. Further analysis and discussion of the results for each stack follows.

Average RUN TIME (ms)	1000 cycles	Stack 1	Stack 2	Stack 3	Stack 4	Stack 5
executed call	# nodes	Basic256Sha256	Basic256Sha256	Basic256Sha256	Basic256Sha256	Basic128Rsa15
READ	1000	2.91	9.33	8.31	9.42	2.71
	1000 - sign & encrypt	5.24	14.47	8.24	11.11	3.36
	5000	10.04	28.25	45.73	43.3	8.97
	5000 - sign & encrypt	16.39	37.17	46.45	50.66	9.29
WRITE	1000	1.95	8	6.85	9.06	2.29
	1000 - sign & encrypt	5.02	10.79	8.18	10.3	3.57
	5000	8.89	26.94	44.1	41.2	8.38
	5000 - sign & encrypt	13.77	34.04	45.55	44.53	7.68
CREATE	1000	6.67	14.07	11.06	9.73	4.51
	1000 - sign & encrypt	9.92	18.49	11.73	12.17	5.76
	5000	18.21	47.7	59.06	40.26	16.84
	5000 - sign & encrypt	28.65	62.45	63.01	51.74	21.13
DELETE	1000	2.26	2.41	1.34	8.57	5.52
	1000 - sign & encrypt	2.56	3.26	1.87	7.85	5.36
	5000	10.59	6.68	3.49	45.44	110.61
	5000 - sign & encrypt	11.01	8.33	3.97	44.22	118.89

Figure 2: OPC UA Performance Data with and without Cryptography

	Stack 1	Stack 2	Stack 3	Stack 4	Stack 5
Operation	**Increase**	**Increase**	**Increase**	**Increase**	**Increase**
Read 1000	1.8 times	1.55 times	.99 times	1.17 times	1.24 times
Read 5000	1.63 times	1.31 times	1.01 times	1.17 times	1.03 times
Write 1000	2.57 times	1.34 times	1.19 times	1.13 times	1.56 times
Write 5000	1.54 times	1.26 times	1.03 times	1.08 times	.88 times
Create 1000	1.48 times	1.31 times	1.06 times	1.25 times	1.27 times
Create 5000	1.57 times	1.31 times	1.06 times	1.28 times	1.25 times
Delete 1000	1.3 times	1.35 times	1.39 times	.91 times	.97 times
Delete 5000	1.03 times	1.25 times	1.13 times	.97 times	1.07 times
Average	1.45 times	1.33 times	1.1 times	1.12 times	1.16 times

Figure 3: OPC UA Performance Data Ratios

Stack 1

Most of the data for Stack 1 is consistent in showing an increase of around 1.5 times. The one outlier is the Write 1,000, which is over a 2x increase However, most important is to observe the average of nearly 1.5x increase when security is applied. In some cases this might not amount to much, but for applications with very sensitive latency this will be a significant impact.

Stack 2

Again Stack 2 is consistent in terms of the time increase when security is added to the protocol. The average increase is slightly less than that for Stack 1, although certainly similar.

Stack 3

 https://doi.org/10.24868/10711

Interestingly for Stack 3 the impact of adding security is noticeably less than for the other two stacks. There is still an impact (except for the outlier case of Read 1,000, where adding security seemingly improves the performance). There could be various reasons for this apparently small degradation of performance. For one thing, some of the performance times without security are noticeably higher than for other stacks, which might allow for a minimal impact when security is added. This stack also has a somewhat unique multi-threading architecture, which could have both contributed to the increase in the base case without security and the minimal increase when security is then enabled.

Stack 4

Stack 4 shows a similar increase in performance as Stack 3, which is moderate compared to the first two stacks. Again, there are two strange outliers where the security performance appears to be better than the non-secured protocol. Stack 4 also supports multithreading, which could be an explanation here.

Stack 5

Stack 5 has a pretty large range in terms of the difference between the performance with security and without. Again, there were a few outliers where performance was better with security, but also quite a few that resembled numbers similar to Stack 1 and Stack 2.

4. Performance Analysis

The first point of discussion regarding the performance data is to note that there are important differences between the test execution environment and the execution environment of a control product like a Programmable Logic Controller (PLC). Running on Linux on a Personal Computer (PC) hardware creates a good deal of non-deterministic execution, certainly more than most PLCs. Many PLCs use real time operating systems, or possibly even run in a “bare metal” environment without any operating system at all. Execution is much more deterministic and predictable, as such the variability of the execution environment is one of the things to contend with in this type of test scenario. Therefore, it is not entirely unreasonable to occasionally see results like timing being better with security turned on, whereas in a control hardware environment one would not expect to see this. As a result, these outliers can be reasonably ignored when discussing the results of the performance test.

Also related to execution environment is the variability in results. In a more real-time execution environment one might expect execution to exhibit less variability, with the increase in execution time being more stable when security is applied.

Another point of discussion is regarding the difference between client-server communication and high-speed cyclic communication (e.g. CIP I/O or OPC UA PubSub). The measurements here were taken on client-server, which is less sensitive to any disturbances in performance. However, high-speed cyclic communication might be processing hundreds of thousands of packets a second, so even a small increase in timing, like 1.5 times or even 1.1 times often will have an outsized effect. In the case of motion control delays may be unacceptable as the application will not be able to function properly if positioning does not match expected time. Therefore, the delays here may be acceptable for client-server, but in many cases would not be acceptable for cyclic communication. Note that the cryptographic algorithms used would be the same or very similar for the cyclic communication, so it can be expected that similar delays will be seen.

A further area of discussion is around multi-threading. Stack 1 and Stack 5 were using a single thread, whereas Stack 2, Stack 3, and Stack 4 all made use of some type of multiple threaded execution. One thing that can be noticed is that the single-thread environment generally leads to shorter overall execution time, but a larger impact of cryptography. Multi-threaded environments are the opposite, longer overall execution time but smaller, and in some cases negligible impact when cryptography is added. This is a noteworthy result that has impact on the design of control systems hardware. In many cases, control systems hardware will not be able to absorb the “cost” of a multi-threaded environment, both in terms of complexity and non-determinism, especially for high-speed data processing (“Embedded Systems”). As such, in many cases the performance impact will be more similar to the single-threaded environment, where the impact of cryptography is higher. However, it is also noteworthy that for systems which are not processing real-time data and have appropriate resources, multi-threading could be a viable path to mitigating the impact of cryptography. Yet for those products where this is not an option, other solutions muse be explored.

5. Hardware Support

Thus, it is recognized that many popular industrial communication protocols have robust security features which provide significant information assurance properties. However, as shown in the experimental data, the protections provided by these protocols can incur performance losses, especially when cryptographic algorithms

are being applied to real-time data exchange. To this end, hardware security features must be designed in carefully so that appropriate performance can be maintained. There are various effective strategies currently employed to mitigate the performance degradation that occurs when cryptographic algorithms are used. The first is to increase the processing bandwidth of the CPU such that the processor is powerful enough to absorb the additional operational cost of cryptography. In many cases this could simply mean an increase in clock speed, which translates to more instructions performed per second. However, this speed increase may not be possible due to heat and cost constraints on an embedded control system. A more complex method might be to use multiple CPU cores, although that assumes that there is an opportunity for parallelization, which may not be the case. To increase performance even more, another method for mitigation is to add some specialized hardware for processing cryptographic operations. This is referred to as a cryptographic hardware accelerator; dedicated hardware that are designed specifically for the cryptographic operations. Cryptographic hardware accelerators have been made to offload the cryptographic operations from the Central Processing Unit (CPU) to free up its utilization which has led to gains in speed of nearly 50% (Jiang, 2019). Many chip vendors now include options for this, whether as a dedicated coprocessor with cryptographic primitives implemented directly in logic gates, or a Field Programmable Gate Array (FPGA) to offload cryptographic operations, or special gates to embed in an ASIC/ASSP (Application Specific Integrated Circuit/Application Specific Standard Part). The next generation of OT hardware must exploit these principles of hardware accelerators and methods like System-on-Chips (SoC) to allow multiple users or I/O devices to share the use of the same crypto accelerator or multiple accelerators on a single PLC. This would allow the PLC's hardware accelerator to process data from different users simultaneously via hardware-supported time sliced multi-tasking. It was already shown through experimental data how multithreading can mitigate some of the performance impact of cryptography, supporting this in PLC hardware would allow for more deterministic execution. Implementing hardware information flow control by adding security type or security-annotated Hardware Description Language (HDL) and enforcing information tracking logic leads to additional performance benefits when cryptographic operations are used with data exchange (Jiang, 2019). Hardware accelerators implementing information flow security policies through HDL achieve greater efficiency and performance. Future advancements in OT hardware performance will be achieved through integration of these technologies; multi-tasking time slice concurrency of encrypting/decrypting data from information flow control where each I/O device's data has an independent security label. While this paper has touched the surface of these options, suffice to say there are many cryptographic hardware acceleration options available and future OT hardware will advance as control system vendors implement those options.

6. Conclusions

This paper explored some of the advancements in cybersecurity data protection in commonly used industrial protocols, especially CIP and EtherNet/IP, and OPC UA. These developments are very positive and will provide robust protections for industrial equipment and data. However, it was also shown that these protections come at a cost of increased processor time to process a packet protected with cryptography. In certain cases as much as 1.5X more time than without cryptographic security. Yet hardware vendors can take steps to mitigate these impacts, especially through careful design of their hardware. Of course, today's increasing processor speed is one way to provide mitigations. However, there may be limitations due to heat or cost that prevent a simple clocking increase. In these cases, adding dedicated hardware to process cryptographic operations is an attractive and effective option, and something that vendors should seriously consider on all hardware products going forward. Implementations that future OT hardware could have like multi-tasking time slice concurrency for encrypting/decrypting data through HDL information flow control have tremendous potential for significant performance gains.

Acknowledgements

The views expressed in this paper are that of the authors and do not necessarily represent the views and opinions of Rockwell Automation.

Special thanks to Ondrej Flek, Marek Pavlicek, and Kristian Sranko for running the experiment on OPC UA Client-Server communication with and without security

References

Stealth Labs, "Number Of Cyber Attacks In 2021 Peaked All-time High", https://www.stealthlabs.com/news/cyberattacks-increase-50-in-2021-peaking-all-time-high-of-925-weekly-attacks-per-organization/, January 19, 2022

"GUIDELINES ON MARITIME CYBER RISK MANAGEMENT", International Maritime Organization MSC-FAL.1/Circ.3, 5 July 2017

"Overview of CIP Security TM", ODVA Technology Overview Series Pub00319R1

Crawford, Douglas; "How Does AES Encryption Work", https://proprivacy.com/guides/aes-encryption February 4, 2019

Zhenghong Jiang, Hanchen Jin, G. Edward Suh, Zhiru Zhang, "Designing Secure Cryptographic Accelerators with Information Flow Enforcement: A Case Study AES". The 56th Annual Design Automation Conference 2019 (DAC '19) June 2-6, 2019.

Carlsson, Thomas, "Industrial networks keep growing despite challenging times", HMS Networks https://www.hms-networks.com/news-and-insights/news-from-hms/2022/05/02/industrial-networks-keep-growing-despite-challenging-times#:~:text=Industrial%20network%20market%20shares%202022,grow%20by%208%25%20in%202022 May 2, 2022.

Google Transparency Report https://transparencyreport.google.com/https/overview?hl=en

ODVA PUB00319, "Overview of CIP Security", https://www.odva.org/wp-content/uploads/2020/05/PUB00319R1_CIP-Security-At-a-Glance.pdf

Knuth, Donald, "Big Omicron and Big Omega and Big Theta", SIGACT News, Apr – June 1976 https://dl.acm.org/doi/10.1145/1008328.1008329

Daemen, Joan and Rijmen, Vincent, "The Design of Rijndael", Springer, 1965

D Rachmawati et al 2018 J. Phys.: Conf. Ser. 978 012116, https://iopscience.iop.org/article/10.1088/1742-6596/978/1/012116/pdf

Burger, Andreas et al 2019, "Bottleneck Identification and Performance Modeling of OPC UA Communication Models", ICPE 2019.

Wikibooks, "Embedded Systems" https://en.wikibooks.org/wiki/Embedded_Systems/Threading_and_Synchronization

OPC Specification, "Part 2: Security Model", OPC Foundation.