SANDIA REPORT
SAND2023-09883
Printed October 2023

Sandia National Laboratories

Barriers and Alternatives to Encryption in Critical Nuclear Systems

Christopher C. Lamb, Daniel Sandoval

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico
87185 and Livermore,
California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

ABSTRACT

Over the past decade, cybersecurity researchers have released multiple studies highlighting the insecure nature of I&C system communication protocols. In response, standards bodies have addressed the issue by adding the ability to encrypt communications to some protocols in some cases, while control system engineers have argued that encryption within these kinds of high consequence systems is in fact dangerous. Certainly, control system information between systems should be protected. But encrypting the information may not be the best way to do so. In fact, while in IT systems vendors are concerned with confidentiality, integrity, and availability, frequently in that order, in OT systems engineers are much more concerned with availability and integrity that confidentiality. In this paper, we will counter specific arguments against encrypting control system traffic, and present potential alternatives to encryption that support nuclear OT system needs more strongly that commodity IT system needs while still providing robust integrity and availability guarantees.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page left blank

ACRONYMS AND TERMS

Acronym/Term	Definition
3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
CA	Certificate Authority
CRL	Certificate Revocation List
DNP3	Distributed Network Protocol 3
DTLS	Datagram Transport Layer Security
HTTP	Hypertext Transfer Protocol
IAEA	International Atomic Energy Agency
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IPSEC	Internet Protocol Security
ISO	International Organization for Standardization
IT	Information Technology
NPP	Nuclear Power Plant
NSS	Nuclear Security Series
OCSP	Online Certificate Status Protocol
OT	Operational Technology
PKI	Public Key Infrastructure
RC4	Rivest Cipher 4
RSA	Rivest–Shamir–Adleman
TCP	Transmission Control Protocol
TLS	Transport Layer Security
VLAN	Virtual Local Access Network

1. INTRODUCTION

Security versus safety concerns in the nuclear power production are deep and complex. The low probability characterization of the high consequence risks in nuclear power are due to decades of careful engineering and expert design. Possibly making it more difficult, our modern digital revolution presents very tempting addition to this already complex operational engineering task. In these systems, safety will always be paramount, but the undeniable advances in digital control agility and associated information security from the commercial IT industry require a closer assessment of the implications of using modern computer architectures and encryption technologies for a nuclear future. To begin this discussion, we review arguments against and in favor of and present alternatives to using encryption in OT.

The common way to perceive information security is through the confidentiality, availability, and integrity lenses. Historically, integrity and availably are most important because they directly contribute to the safe operation of a plant. This we do not challenge. What we do here is to try to understand that if confidentiality can be added without compromising the others. We studied related works around this subject, weighed the arguments for and against commercial encryption in industrial environments, and suggest alternatives to encryption when it is found inappropriate.

This page left blank

2. RELATED WORK

There is a wide array of innovative works that contribute to information and industrial systems control security. In this section we document our exploration of this current and related work. We reviewed areas of encryption that range from local processor implications to the delays in data due to encryption across multiple continents. We also investigated international standards that set information security best practices and specifically nuclear power plant security implementation guidance. Finally, we looked at the possible future impacts of quantum computing on modern encryption technologies.

The study by Restuccia et. al. "Low-Power IoT Communication Security: On the Performance of DTLS and TLS 1.3" [1] includes a performance comparison of active but deprecated versions of TLS 1.2 and DTLS 1.2 versus the current versions DTLS 1.3 vs TLS 1.3. They found that "TLS 1.3 and DTLS 1.3 improvements are accomplished with only small overhead in Flash memory and RAM requirements, compared to TLS/DTLS 1.2." [1] This is encouraging because IOT devices are smaller in size and target market, they are similar in function and purpose.

Other activity in this arena by Kotuliak et. al., "Performance Comparison of IPsec and TLS" [2]. This study discusses the convergence of IT across multiple areas of telecommunications, internet, entertainment, and the need to prevent security incidents across them all. They studied response time and throughput. They found that there are dramatic response time differences between plaintext ethernet traffic when compared to encrypted response times. The largest response time delay was almost half a second over the round trip when using the IPSEC Protocol to send a 3DES encrypted file. Their findings on throughput were also dramatic when comparing unencrypted traffic passing at around 500Mbps against the next fastest of AES over IPSEC at just over 120 Mbps.

A comprehensive guide about securing industrial control systems for the nuclear sector is "Computer Security Techniques for Nuclear Facilities" [3] by the International Atomic Energy Agency. It describes the multifaceted defense techniques to enhance computer security postures. They promote defensive computer security architectures that create distinct computer security boundaries. These boundaries can be handled by applying appropriate security controls at system boundaries are that are then maintained through the plant life cycle.

A broad sweeping standard that is helpful in applying a great security defense to many industries is ISO/IEC 27002 "Information security, cybersecurity, and privacy protection — Information security controls" which includes Organizational Controls that include policies, roles and responsibilities, and documentation, People Controls like screening, workforce education, and remote work aspects of cyber security. It also convers Technical Controls such as firewalls, segregation, and cryptography, and how they can be used in conjunction with Physical Controls like facility perimeter defense, the prevention of unauthorized physical access to offices, rooms, and facilities [4].

Finally, forward leaning research such as the "Post-Quantum Authentication in TLS 1.3: A Performance Study" by Sikeridis et. al. [5] discusses the potential development of large-scale quantum computing and he associated concerns of its ability to solve discrete logarithms, which may cause significant problems across multiple telecommunication industries. They tested the performance impact on existing non-quantum digital computing systems. Findings here show that Post quantum encryption algorithms perform well in lab environments but could have significant performance impacts when used at a larger scale.

This page left blank

3. ARGUMENTS AGAINST ENCRYPTION IN OT SYSTEMS

Encryption can be an effective method for protecting data and communications in OT systems, but its performance can be affected by several factors.

First, many OT systems have limited resources, such as processing power and memory, which can impact the performance of encryption and by extension the performance of the OT device. This can be particularly true for older or legacy systems, which may not have the resources to support encryption algorithms that are considered acceptable by today's standards.

Encryption can also add latency to communications, which can be a concern in some OT systems, such as those that require real-time control or monitoring. Many systems in Nuclear Power Plants (NPPs) do operate under hard real-time constraints, and adding additional processing overhead has the potential to impact functional performance.

Managing encryption keys and certificates can also be a challenge in OT systems. The keys used for encryption must be protected, and a secure method for distributing and updating keys must be in place. Furthermore, OT devices need to check certificate revocation lists or similar constructs to validate used certificates, leading to another dependency and potential time sink depending on implementation.

Finally, encrypted traffic is opaque to monitoring systems. Clear text traffic can be extracted from communication systems and saved into data historians for later analysis and recreation of system state. In encrypted systems, this becomes more difficult. Typically, interleaving systems either do not have access to encrypted information in transit, or they need to implement an intercepting proxy to extract data.

Despite these challenges, encryption can be an effective method for protecting data and communications in OT systems. By understanding the alternative methods available and the potential performance issues, organizations can make informed decisions about how best to protect their OT systems.

Securing operational technology systems is essential for maintaining the integrity, confidentiality, and availability of their processes. While encryption is a widely used method, there are other alternatives such as network segmentation, access control, and optimistic command and control coupled with security monitoring and incident response that can also be effective in providing a layered defense. Encryption performance in operational technology systems can be affected by resource constraints, latency, and key management. That said, organizations can make informed decisions by understanding the alternative methods available and the potential performance issues.

Arguments against using encryption in OT systems center around four different thrusts – performance and latency, opacity, and complexity. First, in nuclear environments safety always trumps security, and implementing encryption in a way that impacts the function of a device is deeply problematic. Furthermore, ciphertext processing will certainly require additional processing power above that needed for cleartext – the question is how much, and if it can cause a safety impact. The second concern is latency. Processing ciphertext will certainly take additional time. How much time is required depends on the device used, and the impact of the time needed is based on the real-time communication constraints in a given context. Key and certificate management is an understood problem. Public key infrastructure (PKI) systems have been in place at a variety of scales for years in information technology (IT) systems. Tying OT systems to PKI increases complexity, overall attack surface, and may impact system availability. Finally, ciphertext is opaque by design;

after all, that is how encryption provides confidentiality. This also makes it correspondingly difficult to save communication easily to data historians.

4. COUNTER ARGUMENTS SUPPORTING ENCRYPTION

Common securable control protocols include IEC 60870-5 (-101 and -104), IEC 61850, Modbus/TCP, and IEEE 1815 (DNP3). The IEC protocol protections are both based on IEC 62351. The Modbus/TCP Security standard defines how Modbus communication over TCP connections should be protected. IEEE 1815-2012 defines specific protection standards for DNP3 communication [6] [7].

Currently, IEC 62351-3 requires TLS v1.2, but does support TLS v1.0 and v1.1 for backward compatibility. IEC 62351-9 requires the use of X.509v3 digital certificates as well as mutual client/server authentication. This standard requires Diffie-Hellman key exchange with RC4 and regular/ephemeral exchange [8].

Modbus/TCP security requires TLS v1.2 as well, recommending AES counter mode cipher suite to support authenticated encryption. It also requires X.509v3 certificates with mutual client/server authentication. Modbus/TCP security mandates key exchange support TLS client-server exchange based on RSA public keys and suggests using TLS client-server key exchange supporting elliptic curve cryptography [9].

IEEE 1815-2012 recommends IPSec or similar VPN access to remote sites, and TLS v1.2 for intra-site communication. TLS v.1.0 is supported for backward compatibility, but no lower. X.509v3 certificates with mutual client/server authentication is required by IEEE 1815-2012 as well, and TLS use should comply with IEC 62351. IEEE 1815-2012 supports RSA as well as regular and ephemeral Diffie-Hellman key exchange [10].

Table 1: A comparison of IEC 60870, IEC 61850, Modbus TCP, and IEEE 1815-2012 security requirements regarding encryption, identification, and key exchange

Standard	Encryption	Identification	Key Exchange
IEC 60870 with security controls defined by IEC 62351	TLS v1.2 with potential fallback to v1.0 and v1.1	X.509v3	Diffie-Hellman with RC4 and regular/ephemeral exchange
	Note: This is defined by IEC 62351		
IEC 61850 with security controls defined by IEC 62351	TLS v1.2	X.509v3	Diffie-Hellman with RC4 and regular/ephemeral exchange
	Note: This is defined by IEC 62351		
Modbus/TCP	TLS v1.2	X.509v3	TLS with RSA or TLS with ECC
IEEE 1815-2012 with required compatibility with IEC 62351	TLS v1.2	X.509v3	RSA and Diffie-Hellman
	Note: This is compatible with IEC 62351		

Table 1 summarizes the types of encryption protections required by various standards. Notably, three of the four standards surveyed mandate compatibility with or are based on IEC 62351. Furthermore, all standards support TLS v1.2 preferably for encryption [11] [12]. Some of the recommendations or requirements need to be updated however as the encryption algorithms specified have known flaws [11].

TLS 1.2 will impose additional network overhead to communications. Specifically, at the beginning of each session the client and server will negotiate acceptable protocols and optionally authenticate each other. This handshaking process will require at least one, and frequently two, round trips from the client to the server and back. Furthermore, both the client and the server may check revocation status of exchanged certificates via checking a Certificate Revocation List (CRL) or via Online Certificate Status Protocol (OCSP). This will require a round trip to the Certificate Authority (CA) that issued the certificates. TLS resumption however will allow sessions to renew TLS connections with a previously established session identifier or ticket. This will lower the number of handshake round trips to only one, between the client and the server [13]. Depending on overall network design, this can potentially impose significant communication latency based on the location of the CA and the speed of the network.

A TLS 1.2 handshake has four exchanges - CLIENT HELLO, SERVER HELLO, CLIENT KEY EXCHANGE, and SERVER EXCHANGE CIPHER SPEC. After the handshake, the client and server begin exchanging encrypted information based on the algorithms and keys negotiated during the handshake step.

Client and Server Hello. The first step, CLIENT HELLO, exchanges the TLS version, the supported cipher suite, and a random number. The SERVER HELLO, sent in response to the CLIENT HELLO, contains the servers TLS version, cipher suite, and a random number. At this point, we have exchanged information between the client and the server. We have in fact had three round trips, an initial SYN and SYN+ACT from the client to the server during the CLIENT HELLO, and then an ACK and CLIENT HELLO to the server, followed by an ACT message. The SERVER HELLO is sent immediately after the ACT. The server will then send its certificate to the client, and then send a SERVER HELLO DONE message. Optionally by the TLS specification, but required by IEC 62531, the server will also send a CLIENT CERTIFICATE REQUEST message during this exchange. The server will then validate the client using that certificate.

After this phase, we have exchanged CLIENT and SERVER HELLO messages only. We have yet to calculate any hashes or encrypt any data. This phase of the handshake exchange is low in CPU and memory overhead and is primarily affected by network communication latency as we have *three round trips between the client and server*.

Client Key Exchange and Change Cipher Spec. As the server requests a certificate by IEC 62531, the client will send that certificate to the server first. Next, the client will validate the server certificate. This involves checking the validity period of the certificate, checking the name of the server against the certificate, checking to see if the CA is trusted, checking to see if the certificate has been revoked, and then verifying the digital signature on the certificate. This involves simple algebraic operations can comparisons until the last two steps. Checking the revocation list requires a round trip the CA in the worst case, whether using CRLs or OCSP. We have network induced latency as well as some impact on CPU and memory. If RSA is selected for key agreement and authentication, the client will generate a 48-byte premaster secret, encrypt it with the public key from the server, and send that to the server. A digitally signed CERTIFICATE VERIFY message may be sent as well where the message is signed with the client's private key to ensure that the client in fact has that key. In this step, we have a *round trip to the CA, we verify a digital signature, we may digitally sign a message with our private key, and we encrypt a 48-byte generated sequence with a public key from the server*.

Next, the client sends a CHANGE CIPHER SPEC message. This message has a single encrypted byte value encrypted via the current connection state. This notifies the server that messages will be

protected in the future under the new cipher specification and keys. Then it sends a FINISHED message. *Here, we have a single byte encryption operation.*

The server then sends a CHANGE CIPHER SPEC message and a FINSHED message, and the client and server begin to exchange application data based on the negotiated algorithms. *We have another single byte encryption.* At this point, the client and the server begin exchanging encrypted data.

Evaluating the impact of TLS 1.2 encoding. To evaluate the end-to-end impact of TLS 1.2 encryption, we ran a series of tests evaluating the connection timings between three different hosts and https://requestb.in, a commercial endpoint system for testing HTTP requests. The hosts were an Intel based system with a 3.5 GHz Intel Xeon E5 and 64 GB of RAM (referred to as **Intel**), a Raspberry PI 3 B+ with a 1.4 GHz ARM8 Cortex-53 SoC and 1GB RAM (referred to as **C53**), and a Raspberry PI 4 with a 1.5 GHz ARM8 Cortex-72 SoC with 4 GB RAM (referred to as **C72**).

In the below tables, CS1 is AES128-SHA, CS2 is ECDHE-RSA-AES128-SHA, CS3 is ECDHE-RSA-AES256-SHA, CS4 is ECDHE-RSA-AES128-SHA256, CS5 is ECDHE-RSA-AES256-SHA384, XS6 is ECDHE-RSA-AES128-GCM-SHA256, and CS7 is ECDHE-RSA-AES256-GCM-SHA384. All the measurements are in milliseconds as well. We specifically used a wide range of available algorithms to examine differences in algorithmic performance.

Table 2: The mean and standard deviation of the difference between clear text and enciphered unoptimized communication in milliseconds. Note that although we do verify the certificates, the library used for OpenSSL assess does not support CRL checking in TLS 1.2 communication.

		Cipher Suite													
Payload		CS1		CS2		CS3		CS4		CS5		CS6		CS7	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
No Payload	**Intel**	-26.8	353	-5.11	44.4	22.53	303	-2.11	57.11	0.345	41.72	0.753	57.46	5.91	65.98
	C53	32.43	502.8	-14.9	92.95	8.327	57.75	-4.69	28.02	-1.01	47.83	-2.56	44.03	-3.63	33.37
	C72	37.47	624.8	-0.19	41.58	-0.76	45.25	-7.21	65.82	2.448	83.01	-7.05	54.04	-2.27	69.69
512 Byte Payload	**Intel**	4.462	34.25	-5.53	32.89	-11.2	72.62	1.293	47.54	0.875	38.24	-4.53	56.14	-7.30	65.28
	C53	-2.29	55.63	-4.66	26.74	-4.5	46.84	-4.01	22.19	29.67	311.2	-2.42	64.77	-23.9	300.3
	C72	-2.50	49.57	0.153	57.26	7.142	51.82	-2.51	63.04	6.121	132.2	14.24	106.8	11.43	79.43
1024 Byte Payload	**Intel**	-0.46	40.5	-1.4	59.3	2.22	65.6	-9.25	44.3	2.22	49.4	6.23	125.5	-2.7	44.0
	C53	3.50	54.69	53.44	348.9	-1.63	42.47	2.77	43.86	4.352	50.02	-6.07	106.0	-2.98	34.70
	C72	-1.89	50.11	-5.86	47.32	0.541	83.81	1.35	92.56	24.67	520.9	10.56	73.50	-4.61	134.4

We evaluated two distinct cases in each measurement. We implemented the test program in python, using the SSL and Socket modules. In each case we sent an HTTP 1.1 POST request with either no attached data, 512 bytes of data, or 1024 bytes of data. We sent the same HTTP message over a connection with TLS 1.2 and the selected cipher suite, and over a bare socket connection. We sent them consecutively, measuring the time required to send the message and receive a response. We then measured the difference in the time required, subtracting the time required for the clear socket communication from the time required for the TLS 1.2 protected communication. We executed communications with each cipher suite and payload size 100 times per platform. We also deactivated connection optimization (i.e., session tickets and compression) to generate worst-case communication timings. We then recorded the mean and standard deviation of these differences for each test.

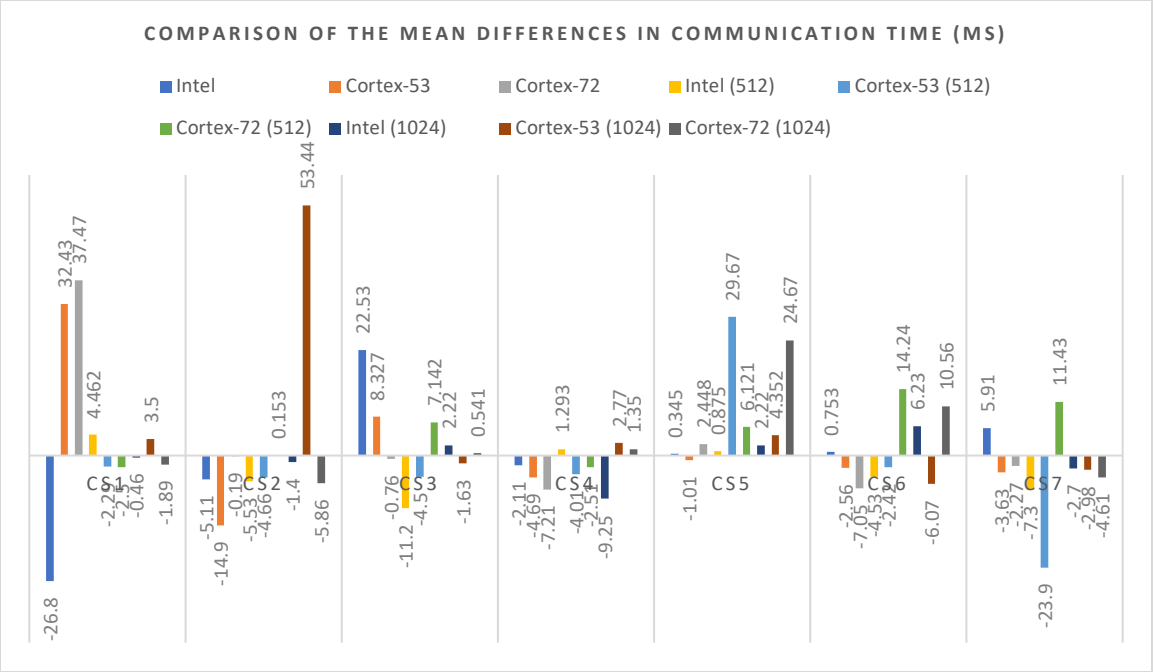

Figure 1: A comparison of the mean of the round-trip time to deliver a single payload packet and receive a response. The size of the payload, if sent, is included parenthetically in the legend.

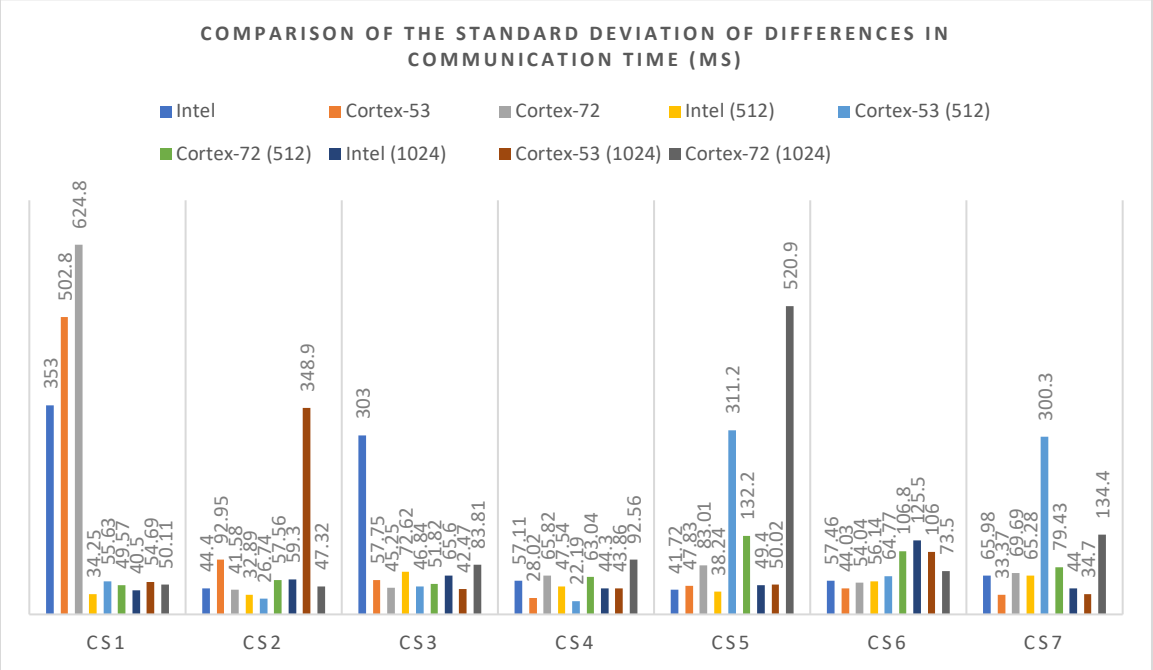

Figure 2: A comparison of the standard deviations of the differences in communication time between TLS 1.2 protected communications and clear communications. Again, the size of the payload, if sent, is included parenthetically in the legend.

Overall, we had remarkable volatility in communication timings, with relatively large and unpredictable deviations. Furthermore, we had cases where unprotected communication was slower than protected. This experiment connected to remote internet systems to evaluate timings and used

commercial certificate authorities for server certificate verification. In a nuclear facility, the network would be much smaller, and any network dependencies like a CA would be managed by the organization rather than an external third party. This would likely lead to a drastic decrease in the deviation values as well as a decrease in mean time for communication as in our testing we were using remote resources.

Depending on the specific application, even these timings may be acceptable in nuclear control systems, depending on specific real-time constraints. The highest standard deviation was 0.6[1] seconds, but typical deviations were around 50 milliseconds (ms). Likewise, though the max mean difference was 53.44 ms, more typical differences were roughly 5 ms. And these measurements are on internet connected systems, not systems in a typical nuclear plant.

Performance of encrypted connections is likely an issue with straightforward engineering solutions. The imposed latency and timings can be engineered around for most applications, even when executed on low powered computational equipment, as they were in our testing. Interestingly, there seemed to be no difference in performance based on platform. Furthermore, the additional infrastructure required to implement TLS1.2 protected communication (i.e., PKI systems, including CAs) is stable, widely implemented, and well understood. Certificate distribution to protected systems requires administrative oversight to ensure certificates never become invalid.

Encryption will not solve data opacity issues however without using intercepting proxies.

[1] This is a result of the way that the socket and ssl modules implement socket creation in python. These libraries use lazy socket initialization, causing initial socket creation to take significantly longer. That socket is then reused. This leads to a longer initial startup time, which is reflected in these results. Nevertheless, we still had deviations nearly this high even when the socket was reused.

This page left blank

5. ALTERNATIVES TO OT DATA ENCRYPTION

Encryption is not the only method available for protecting OT systems. There are alternative methods for securing OT systems.

A common alternative method used today for securing OT systems is network segmentation, dividing a network into smaller, isolated segments. This can be done using firewalls, virtual local networks (VLANs), or other network-segmentation tools like data diodes. By segmenting a network, an attacker's ability to move laterally within the network is more limited, making it more difficult for them to access critical systems. IAEA NSS 17-T suggests this approach for nuclear systems specifically via the risk-informed layered approach to secure zone design [3]. When coupled with strict physical security systems, well managed network segmentation is a valid design alternative.

Frequently, in areas of particular concern in reactor control systems encryption or other information protection technical approaches should not be used. Safety systems or similar critical systems with stringent hard real-time constraints are examples of the kinds of systems where encryption or other specific integrity and confidentiality controls can potentially cause functional degradation of the system and the risk of using those technical approaches cannot be accepted by an operator. When this happens, engineers should minimize the specific systems operating without integrity protection, and strongly protect the system perimeters. In IT systems, perimeter protection is widely recognized as insufficient, driving adoption of zero-trust approaches. Nuclear control systems have specific cases where strong perimeter protection may in fact be the best option.

Once a system is compromised however, though segmentation makes lateral movement much more difficult, the lack information protection with that compromised segment allows an attacker the ability to easily execute a variety of attacks, including command injection, false data injection, and other attacks that can compromise the integrity of the segmented system. This also violates defense in depth from a cybersecurity perspective.

Today, data integrity is tightly coupled to encryption techniques. Typically, in everyday computing users are concerned with confidentiality as well as integrity but are willing to sacrifice some availability. Nuclear control systems have a different hierarchy of attributes, where availability and integrity are vital. Confidentiality may be useful, but it is not as important.

TLS and similar protocols use digital signatures to negotiate a shared secret and then use that secret to enable symmetric encryption for data transfer. This approach works well as symmetric encryption is much faster than public key cryptography. In industrial environments however, where the frequency of data transfer may be significantly lower, an integrity-guaranteeing protocol using digital signatures may be a better fit. In these cases, the data would be transferred in cleartext, but would be signed by the originator. The benefit of this scheme is the ability to easily monitor data as it travels along the wire, but it comes at the cost of more public key encryption which may lead to lower performance.

This page left blank

6. CONCLUSIONS AND FUTURE WORK

Overall, encryption techniques outlined in the current standards or more modern alternatives can be engineered into future control systems. The performance of TLS 1.2 seems acceptable for control system use with appropriate engineering of the control system and dependencies. Despite some large performance deviations, most likely caused by communications latency, the difference in round trip time was typically around 5 ms. Typical deviations were around 50 ms, but as these experiments used internet resources, certificate validation and revocation checking as well as initial handshaking had large travel times and would be exposed to more potential performance volatility.

As our initial tests seem promising, our future work may include an effort to evaluate the performance of TLS 1.2 algorithms in a more representative environment with a local CA to test our hypothesis that the variation in performance is related to using internet rather than local services. We can also begin experimenting with new protocol designs to enhance integrity without requiring confidentiality, specifically measuring the potential impact of increasing the number of public key verifications and encryption operations.

REFERENCES

[1] G. Restuccia, "Low-Power IoT Communication Security: On the Performance of DTLS and TLS 1.3," IFIP/IEEE PEMWN 2020 , 2020.

[2] I. Kotuliak, "Performance Comparison of IPsec and TLS," 9th IEEE International Conference on Emerging eLearning Technologies and Applications, Stará Lesná, The High Tatras, Slovakia, 2011.

[3] International Atomic Energy Agency` , "Computer Security Techniques for Nuclear Facilities," International Atomic Energy Agency, Vienna, Austria, 2021.

[4] International Electrotechnical Commission, "EC 27002 - Information Security, Cybersecurity, and Privacy Protection," International Electrotechnical Commission, 2022.

[5] D. Sikeridis, "Post-Quantum Authentication in TLS 1.3:," Cryptology ePrint Archive, Albuquerque, New Mexico, USA, 2020.

[6] Commission, International Electrotechnical, "IEC 60870 - Telecontrol Equipment and Systems," International Electrotechnical Commission.

[7] Commission, International Electrotechnical, "IEC 61850 - Communication Networks and Systems for Power Utility Automation," International Electrotechnical Commission, 2023.

[8] Commission, International Electrotechnical, "IEC 62351 - Power Systems Management and Associated Information Exchange - Data and Communications Security," International Electrotechnical Commission, 2023.

[9] Modbus Organization Inc., "MODBUS/TCP Security," Modbus Organization Inc., 2018.

[10] IEEE Power and Energy Society, "IEEE Standard for Electric Power Systems Communications— Distributed Network Protocol (DNP3)," IEEE Standards Association, 2012.

[11] J. Johnson, I. Onunkwo, D. Saleem, W. Hupp, J. Peterson and R. Cryar, "Distributed Energy Resource Cybersecurity Standards Development – Final Project Report," Sandia National Laboratories, Albuquerque, 2022.

[12] I. Onunkwo, "Recommendations for Data-in- Transit Requirements for Securing DER Communications," Sandia National Laboratories, Albuquerque, 2020.

[13] Internet Engineering Task Force (IETF), "RFC5246 - The Transport Layer Security (TLS) Protocol Version 1.2," Internet Engineering Task Force (IETF), 2008.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Ben Cipiti	8845	bbcipit@sandia.gov
Technical Library	1911	sanddocs@sandia.gov

Email—External

Name	Company Email Address	Company Name
Katya LeBlanc	katya.leblanc@inl.gov	Idaho National Laboratory

This page left blank

This page left blank

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.