U.S. Department
of Transportation
Federal Railroad
Administration

Office of Research,
Development and Technology
Washington, DC 20590

PTC Communications: Cybersecurity Technology Review and Concept of Operations

DOT/FRA/ORD-23/39

Final Report | December 2023

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
08-12-2023	Technical Report	28-09-2020 – 28-02-2022

4. TITLE AND SUBTITLE

PTC Communications: Cybersecurity Technology Review and Concept of Operations

5a. CONTRACT NUMBER
693JJ620C000034

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
#04.01

6. AUTHOR(S)

Andy Collins: ORCID 0000-0002-4925-8120

Armando Stettner: ORCID 0000-0002-7350-036X

Neil Evans: ORCID 000-0002-4088-4416

Steven Brog: ORCID 0000-0001-9902-646X

Peter Mayer: ORCID 0000-0002-4454-4363

5d. PROJECT NUMBER
FR20RPD33000000007

5e. TASK NUMBER
MCC WBS Task 5.4

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Meteorcomm, LLC
1201 SW 7th St
Renton, WA 98057

8. PERFORMING ORGANIZATION REPORT NUMBER
DCN 00004816-A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Department of Transportation
Federal Railroad Administration
Office of Railroad Policy and Development
Office of Research, Development, and Technology
Washington, DC 20590

10. SPONSOR/MONITOR'S ACRONYM(S)
FRA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
DOT/FRA/ORD-23/39

12. DISTRIBUTION/AVAILABILITY STATEMENT
This document is available to the public through the FRA website.

13. SUPPLEMENTARY NOTES
COR: Francesco Bedini Jacobini

14. ABSTRACT
Researchers sought to determine how to improve the confidentiality of information passing through the Positive Train Control network without significantly affecting network performance. Phase 1 of the project reviewed requirements and assessed existing technologies. Phase 2 focused on implementation, migration, and deployment challenges. A recommendation to consider widely available algorithms covered by recognized open standards resulted in a project decision to focus on the National Institute of Standards and Technology Lightweight Encryption Algorithms and the ELLI algorithm covered by ISO/IEC 29192-4. Phase 3 identified technology gaps and areas for further research outside the scope of the project. Phase 4, the focus of this report, pulls together the work from Phases 1-3 to propose solutions to improve confidentiality, including a concept of operations for each solution.

15. SUBJECT TERMS
Positive Train Control, PTC, railroad, shared network, Interoperable Train Control Messaging, ITCM, cybersecurity, encryption, National Institute of Standards and Technology, NIST, ELLI

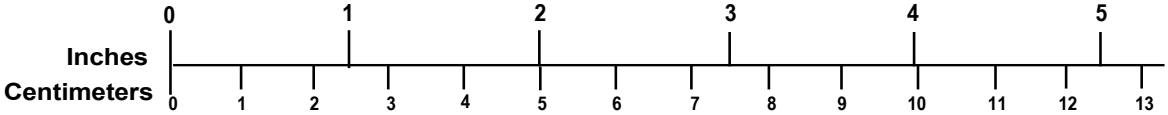
16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Andy Collins
Unclassified	**Unclassified**	**Unclassified**		55	**19b. TELEPHONE NUMBER** *(Include area code)* (253) 216-8194

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

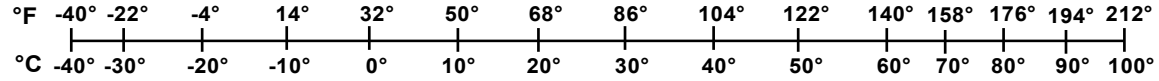
METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC	METRIC TO ENGLISH
LENGTH (APPROXIMATE)	**LENGTH (APPROXIMATE)**
1 inch (in) = 2.5 centimeters (cm)	1 millimeter (mm) = 0.04 inch (in)
1 foot (ft) = 30 centimeters (cm)	1 centimeter (cm) = 0.4 inch (in)
1 yard (yd) = 0.9 meter (m)	1 meter (m) = 3.3 feet (ft)
1 mile (mi) = 1.6 kilometers (km)	1 meter (m) = 1.1 yards (yd)
	1 kilometer (km) = 0.6 mile (mi)
AREA (APPROXIMATE)	**AREA (APPROXIMATE)**
1 square inch (sq in, in^2) = 6.5 square centimeters (cm^2)	1 square centimeter (cm^2) = 0.16 square inch (sq in, in^2)
1 square foot (sq ft, ft^2) = 0.09 square meter (m^2)	1 square meter (m^2) = 1.2 square yards (sq yd, yd^2)
1 square yard (sq yd, yd^2) = 0.8 square meter (m^2)	1 square kilometer (km^2) = 0.4 square mile (sq mi, mi^2)
1 square mile (sq mi, mi^2) = 2.6 square kilometers (km^2)	10,000 square meters (m^2) = 1 hectare (ha) = 2.5 acres
1 acre = 0.4 hectare (he) = 4,000 square meters (m^2)	
MASS - WEIGHT (APPROXIMATE)	**MASS - WEIGHT (APPROXIMATE)**
1 ounce (oz) = 28 grams (gm)	1 gram (gm) = 0.036 ounce (oz)
1 pound (lb) = 0.45 kilogram (kg)	1 kilogram (kg) = 2.2 pounds (lb)
1 short ton = 2,000 pounds (lb) = 0.9 tonne (t)	1 tonne (t) = 1,000 kilograms (kg) = 1.1 short tons
VOLUME (APPROXIMATE)	**VOLUME (APPROXIMATE)**
1 teaspoon (tsp) = 5 milliliters (ml)	1 milliliter (ml) = 0.03 fluid ounce (fl oz)
1 tablespoon (tbsp) = 15 milliliters (ml)	1 liter (l) = 2.1 pints (pt)
1 fluid ounce (fl oz) = 30 milliliters (ml)	1 liter (l) = 1.06 quarts (qt)
1 cup (c) = 0.24 liter (l)	1 liter (l) = 0.26 gallon (gal)
1 pint (pt) = 0.47 liter (l)	
1 quart (qt) = 0.96 liter (l)	
1 gallon (gal) = 3.8 liters (l)	
1 cubic foot (cu ft, ft^3) = 0.03 cubic meter (m^3)	1 cubic meter (m^3) = 36 cubic feet (cu ft, ft^3)
1 cubic yard (cu yd, yd^3) = 0.76 cubic meter (m^3)	1 cubic meter (m^3) = 1.3 cubic yards (cu yd, yd^3)
TEMPERATURE (EXACT)	**TEMPERATURE (EXACT)**
[(x-32)(5/9)] °F = y °C	[(9/5) y + 32] °C = x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSIO



For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50 SD Catalog No. C13 10286

Updated 6/17/98

Acknowledgments

The authors welcomed the input and assistance from the project's Technical Advisory Group comprised of Class I railroads (BNSF, CSX, Norfolk Southern Railway, and Union Pacific Railroad), the Association of American Railroads and its subsidiaries, the American Public Transportation Association, the American Short Line and Regional Railroad Association, and cybersecurity consultants (ANDRO Computational Solutions, AgilePQ®, and Systra Consulting).

Contents

Executive Summary 7

1. Introduction 9
   1.1 Background 9
   1.2 Objectives 9
   1.3 Overall Approach 10
   1.4 Scope 10
   1.5 Organization of the Report 10

2. Background ..... **Error! Bookmark not defined.**
   2.1 PTC Communications 11
   2.2 Phase 1 13
   2.3 Phase 2 14
   2.4 Phase 3 15
   2.5 Other TAG Input 15

3. Symmetric and Asymmetric Algorithm Evaluations by ANDRO 18
   3.1 NIST Lightweight Encryption Algorithms 18
   3.2 Elliptic Curve Cryptography 26
   3.3 Conclusions and Next Steps 29

4. Candidate Solutions 30
   4.1 Candidate 1: Application-Layer Encryption (Non-WSM) 30
   4.2 Candidate 2: Application Layer Encryption for Wayside Status 37
   4.3 Candidate 3: Radio D-Frame Encryption 43
   4.4 Alignment with Roadmap 46

5. Final Summary and Recommendations 48
   5.1 Phases 1 to 3 48
   5.2 Common Themes 48
   5.3 Recommendations Based on The NIST Lightweight Cryptography Project 49
   5.4 Recommended Applications of Lightweight Encryption 49

6. Conclusion 51

7. References 52

Abbreviations and Acronyms 52

Illustrations

Figure 1. Key Elements within The PTC Communication System 11
Figure 2. NIST SP 800-63 Identity Assurance Roles 16
Figure 3. Example Input and Output of Test Vector with PT and AD as Non-Zero 19
Figure 4. Example Input/Output of NIST Test Vector with PT and AD Set to 0 19
Figure 5. Testing Methodology 20
Figure 6. Average Encrypt Times, Varying Plain Text, and Constant Associated Data 20
Figure 7. Average Encrypt/Decrypt Times, Constant Plain Text, and Associated Data 21
Figure 8. Comparison of Encrypt/Decrypt Times on Intel vs. Pi Platforms 23
Figure 9. Large Encrypt/Decrypt on Pi B+ 23
Figure 10. Summary of Initial Down Selection and Remaining Algorithms 24
Figure 11. Wayside Status Message Packet Capture 25
Figure 12. Algorithm Results on The Intel Platform 25
Figure 13. Algorithm Results on Pi Platform 26
Figure 14. ECC-Based Diffie Hellman Establishment of Shared Secret 27
Figure 15. ELLI Algorithm 28
Figure 16. Explicit Steps for ELLI 28
Figure 17. Categories of Candidate Solutions 30
Figure 18. Application-Layer Encryption (Non-WSM) 31
Figure 19. Wayside Status Message Header and Body 33
Figure 20. Application-Layer Encryption for Wayside Status 37
Figure 21. Radio D-frame Encryption 43

Tables

Table 1. Cryptographic Primitives 14
Table 2. Algorithm Submissions Tested and Reference Websites 19
Table 3. Initial Summary of Evaluation Metrics on Intel i5 Laptop 22
Table 4. Comparing ECC to RSA 27
Table 5. Relevant Version and Type Fields in The EMP Header 32
Table 6. Candidate 1 Risks and Mitigations 37
Table 8. Candidate 2 Risks and Mitigations 43
Table 9. Candidate 3 Risks and Mitigations 46

Executive Summary

The primary goal of this project was to determine how to improve confidentiality of information on both the wireless and wired portions of the Positive Train Control (PTC) network, while minimally affecting network performance. Meteorcomm, LLC (MCC) conducted the project with input and assistance from a Technical Advisory Group (TAG) comprised of representatives from Class I railroads, the Association of American Railroads (AAR), MxV Rail, the American Public Transportation Association, the American Short Line and Regional Railroad Association, and cybersecurity consultants from ANDRO Computational Solutions, AgilePQ®[1] and Systra Consulting.

In Phase 1 of the project, researchers reviewed requirements and recommended candidate solutions after completing a technology review and assessment. The AAR S-9010 Data Protection standard [1] outlines railroad data classification standards and confidentiality requirements, and was critical to the project, particularly Phase 1. Based on their analysis, the research team concluded that most messages exchanged over the PTC communications network meet the requirements of S-9010. The TAG concluded that wayside status messages (WSMs) also meet the requirements and do not need to be encrypted; thus, the Interoperable Train Control committee has only identified crew login as a confidentiality requirement. Recognizing that requirements can change, the project sought to identify how and where encryption could be applied if required. The authors also note that when S-9010 requires protection at the application layer, implementing protection at lower layers does not alleviate the requirement.

Phase 2 focused on implementation, migration, and deployment challenges. The research team recommended considering widely-available algorithms with recognized open standards and validated libraries. Thus, the project focused on the National Institute of Standards and Technology (NIST) “Final 10” Lightweight Encryption Algorithms and the Elliptic Light (ELLI) algorithm in ISO/IEC 29192-4 [2]. ANDRO played a pivotal role in Phase 2, assessing these algorithms with a focus on performance.

Phase 3 identified technology gaps and areas for further research outside the scope of the project. This report (Phase 4) incorporates Phases 1–3 to propose improvements to confidentiality, including a concept of operations. The research team identified several areas in which confidentiality could be improved.

Applying symmetric encryption using NIST lightweight algorithms to non-WSM application messages could improve confidentiality while minimizing the additional load on endpoints. Existing protocol features enable migration in a straightforward manner. The Ascon, Spark, TinyJAMBU, and Xoodyak algorithms are recommended for consideration, though this list may be revised based on NIST standardization efforts. These algorithms could replace existing algorithms or be reserved for messages not yet encrypted.

A hybrid approach could be used for WSMs, in which a lightweight asymmetric algorithm (such as ELLI) would be used to securely generate and distribute symmetric wayside Operational Private Keys (OPKs). The symmetric OPKs would be short-lived, improving security. They could also be made compatible with the existing hash-based message authentication code

[1] AGILEPQ is a trademark of AgilePQ, Inc.

algorithm, in which case WSM payload encryption would not be supported. Alternatively, NIST lightweight encryption could be used, and may offer payload encryption. Given the current loading of onboard systems, this approach would likely only be practical if coupled with adoption of WSM filtering and/or optimized remote Interoperable Train Control Messaging (ITCM©) software to counterbalance the added processing load from encryption. In either case, applications would revert to the current pre-shared wayside OPKs as necessary during migration, which would require management by an interoperable key exchange service.

Encryption of the Radio D-frame using NIST algorithms is possible but would also introduce additional processing load. This would likely require encryption/decryption within ITCM because it is optimizable; radios, on the other hand, pose significant optimization difficulties.

Based on the industry roadmap for PTC communications, network loading will increase significantly over the next decade. Therefore, improved security must be considered with other demands, and the industry should implement measures to increase capacity where possible.

1. Introduction

The primary goal of this project was to identify confidentiality improvements for information on the wireless and wired portions of the Positive Train Control (PTC) network, while minimally affecting network performance. Meteorcomm, LLC (MCC) conducted the project with input and assistance from a Technical Advisory Group (TAG) comprised of representatives from Class I railroads, the Association of American Railroads (AAR), MxV Rail, the American Public Transportation Association (APTA), the American Short Line and Regional Railroad Association, and cybersecurity consultants (ANDRO Computational Solutions [ANDRO], AgilePQ®[2] and Systra Consulting).

1.1 Background

With the continuing rollout of Positive Train Control (PTC) systems, U.S. railroads and government agencies are working to promote improved security and explore innovative approaches for maintaining confidentiality of rail communications. Large geographical operational areas and bandwidth constraints necessitate that proposed solutions be highly bandwidth efficient and compatible with existing PTC radios and equipment. This project proposes solutions to improve confidentiality for Interoperable Electronic Train Management System (I-ETMS). The results from Phases 1–3 of MCC's PTC Communications Cybersecurity Technology Review are synthesized in this report, which is Phase 4 of the project. The report also considers the research into lightweight encryption algorithms performed by ANDRO, issues of authentication/identity assurance, and data integrity, which are essential parts of any solution. The authors document proposals to create a concept of operations (ConOps).

1.2 Objectives

The purpose of this report and MCC's PTC Communications Cybersecurity Technology Review project is to propose solutions to improve confidentiality for I-ETMS and document the proposals as a concept of operations. Solutions and proposals include:

1. Align with the industry roadmap for Interoperable Train Control (ITC) (developed by Train Control, Communications, and Operations [TCCO]) covering the next 5–10 years, including Quasi-Moving Block (QMB) and Automated Train Operations (ATO).
2. Use MCC's communications roadmap, intended to align with the TCCO roadmap.
3. Capture MCC's recommendations for the minimum communication security requirements (including where protection measures should be applied) and the security solution high-level design.
4. Recommend a strategy for implementation and migration which considers interoperability requirements. This will provide a detailed communication security picture, including migration to future railroad operations.
5. Estimate the impact of changes to PTC network bandwidth requirements through analysis and modeling.

[2] AGILEPQ is a trademark of AgilePQ, Inc.

6. Identify measures to minimize and mitigate the impact of recommended security changes.
7. Identify the major risks and provide recommendations for risk mitigation.

1.3 Overall Approach

Phase 4 synthesizes the work from Phases 1–3 to propose improvements to confidentiality, including a concept of operations for each one.

- Phase 1 of the project reviewed requirements and assessed existing technologies.
- Phase 2 focused on implementation, migration, and deployment challenges.
- Phase 3 identified technology gaps and areas for further research outside the scope of the project.

Based on the previous phases, four candidate solutions were identified and investigated further. The research team recommended considering widely-available algorithms with recognized open standards, resulting in a project decision to focus on the National Institute of Standards and Technology Lightweight Encryption Algorithms and the Elliptic Light (ELLI) algorithm covered by ISO/IEC 29192-4.

1.4 Scope

This document reports the findings from Phase 4 of the PTC Communications Cybersecurity Technology Review project, which surveyed the performance and applicability of lightweight encryption algorithms for confidentiality in the PTC network.

1.5 Organization of the Report

The document is structured as follows:

- Section 2 provides background on the proposals, including information about the PTC Communications System and inputs from Phases 1–3.
- Section 3 captures the findings of the research performed by ANDRO into performance of lightweight symmetric and asymmetric algorithms.
- Section 4 builds upon the previous sections to present four candidate solutions for lightweight encryption, providing a ConOps for each.
- Section 5 provides a final summary and recommendations.

2. Overview

The purpose of this section is to provide an overview of the architecture and protocols used in PTC communications.

2.1 PTC Communications

Figure 1 illustrates key elements within the PTC communications system. In the diagram:

- The red boxes represent the 220MHz radio transport.
- The blue boxes represent elements of ITCM (also referred to as simply messaging) system.
- The green boxes represent elements of the Interoperable Train Control Systems Management layer (referred to as ITCSM or simply systems management).

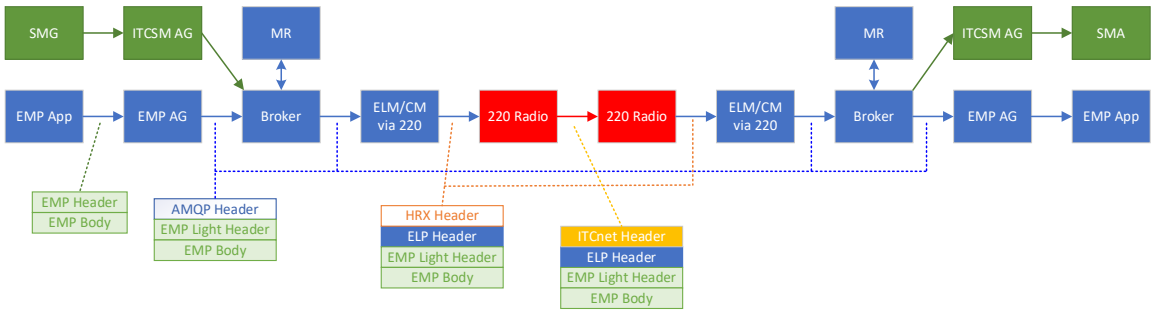


Figure 1. Key Elements within The PTC Communication System

Each element is expanded upon in the following sections.

2.1.1 Radios

- 220 MHz radios are comprised of base radios connected to the Back Office (BO), locomotive radios, and wayside radios. Radios use ITCnet, a proprietary MCC protocol. Wayside radios are responsible for sending wayside status messages (WSMs) to locomotives.
- Radios operating in other bands (44 MHz, 450 MHz, and 900 MHz) may be added as PTC transports supported by messaging in the future.

2.1.2 Messaging (ITCM)

- Messaging may use other Internet Protocol (IP)-based transports beyond radio (not shown on the diagram). These include hard IP paths, cellular communications paths, satellite, and Wi-Fi.
- ITCM is based on Red Hat Enterprise Linux Messaging Real-time Grid (MRG), MRG messaging brokers, and a few custom components. The messaging functionality is based on Qpid, an Apache open-source implementation of the Advanced Message Queuing Protocol (AMQP).
- Applications (apps) interact with an ITCM Application Gateway (AG). They can use a Class D connection with an Edge Message Protocol (EMP) message or an AMQP

connection with the EMP message embedded in the AMQP message body. EMP and Class D are governed by AAR standards S-9354 and S-9356, respectively.

- The message router (MR) selects the appropriate transport method and path to deliver a message.
- Brokers deliver messages and consist of exchanges and queues. They act as servers for AMQP requests. Client apps connect to brokers to produce and consume messages.
- External Link Managers (ELM) are in the BO and deliver messages to and from remote ITCM instances over 220 MHz radio transports. A variant of the ELM referred to as the IP ELM does the same over IP transports.
- The connection manager (CM) provides the same functionality as ELMs for locomotives and waysides (collectively referred to as remotes).
- 220 MHz ELMs and CMs communicate with 220 MHz radios using the External Link Protocol (ELP), a proprietary MCC protocol. The ELM strips messages down to their original application message payload and then adds a small, efficient ELP header. This is based on the original AMQP header to maintain critical routing information.

2.1.3 Systems Management (ITCSM)

- ITCSM is designed to securely pass information about status, events, and configuration for the different ITC assets over the ITC communications platform. It also provides a secure method for railroad BO applications to configure and manage each ITC asset remotely. ITCSM also supports the transfer and loading of software, security, data, and configuration kits, and the remote running of commands.
- The major system components of ITCSM are the ITC Systems Management Gateway (SMG) and the ITCSM Agents (SMA). The SMG communicates with the railroad BO systems and with the SMGs of other railroads. The SMG also communicates with ITC assets through their SMAs. SMAs may be embedded in assets (as in the case with 220 MHz radios) or remote agents (external to assets they manage but connected to them).
- ITCSM components use either Class D or AMQP to connect to the ITCM AG to receive and send messages through the ITC messaging system in a similar manner to ITCM apps.

2.1.4 Wayside Status Messages and Operational Private Keys

- WSMs report the status of signals and switches connected to wayside interface units (WIUs).
- WSMs account for most of the traffic handled by the PTC system. This is due to the large number of wayside devices, each of which streams periodic broadcasts when a WIU is beaconing. This results in at least one message per WIU every 4 seconds.
- Typically, the communication system duplicates WSM broadcasts to ensure message delivery.
- WSMs are generated at the wayside, typically by the WIU, and consumed by the on-board computer (OBC) on the locomotive (i.e., the train management computer [TMC]).

- To address coverage and reception issues, a BO application known as the Wayside Status Repeater Service (WSRS) must also be able to read and forward WSMs to selected base radios and subscribing locomotives.
- Generation, protection, and validation of WSMs are vital (i.e., safety-critical) functions that must be performed by vital applications.
- At present, application-layer WSM security is implemented by signing the status messages with a hash-based message authentication code (HMAC). The HMAC signature (i.e., the vital data integrity value) verifies the data integrity and authenticity of the WSM. This prevents invalid messages, whether accidental or malicious, from being acted upon by the TMC.
- The HMAC algorithm uses a cryptographic hash function in combination with a symmetric secret key, the Wayside Operational Private Key (Wayside OPK).
- If an attacker acquires a wayside OPK, they could subvert the authenticity checks in the HMAC signature and intentionally send invalid messages to the locomotive. Secrecy of the Wayside OPK is therefore of prime importance.
- Because Wayside OPKs are used for symmetric encryption and decryption, a copy must be shared with every locomotive which may require status from that wayside, as well as any BO applications (e.g., WSRS). This broad sharing of long-lived Wayside OPKs represents a risk to PTC security.

2.2 Phase 1

The goal of Phase 1 was to recommend candidate solutions for further research. The research team considered the following:

1. Risks, attack surfaces, and vectors, AAR requirements for data protection, and the prudent protection mechanisms to be engaged
2. Critical characteristics for ensuring confidentiality, data integrity, and authentication
3. Technology from the Internet of Things (IoT) space, industrial control systems (ICS), mobile device management (MDM), and other transportation sectors
4. The NIST Lightweight Cryptography project

The research team established a framework for assessing candidate solutions that included:

1. Considering the AAR Standard S-9010 data classification requirements covering data protection and its implications for PTC communications. This standard provides guidance on classifications and the protection mechanisms that should be applied as follows:
    a. Railroad Restricted: requires privacy, integrity, and authentication protocols.
    b. Railroad Operational Sensitive: requires integrity and authentication protocols. Also requires privacy if information contains attributes identifying specific shippers or commodities in addition to location.
    c. Railroad Official Use: requires integrity and authentication protocols.
    d. Public Information: no security required.

2. Reviewing key interfaces, transferred data, and protection mechanisms
3. Identifying key characteristics of protection mechanisms and the environment required for PTC, including low bandwidth/overhead, low latency, key length requirements, and the ability to support intermittent connectivity, interoperability, and key management for remotes
4. Recognizing constraints related to intellectual property rights and associated restrictions
5. Capturing best practices for ensuring confidentiality, data integrity, and authentication

Table 1 [3] lists the cryptographic primitives, protection (or service) properties that may be used to meet these requirements.

Table 1. Cryptographic Primitives

Service	Symmetric Algorithms	Asymmetric (Public-Key) Algorithms	Digital Signature Algorithms	Key Agreement Algorithms	One-Way Hash Algorithms	MAC
Confidentiality	Yes	Yes	No	Yes	No	No
Authentication	No	No	Yes	Optional	No	Yes
Integrity	No	No	Yes	No	Yes	Yes
Key Management	Yes	Yes	No	Yes	No	No

Based on input from the TAG, the protection mechanisms in S-9010 are intended to be applied at the application layer. Use of the application at lower layers (e.g., the transport layer) does not alleviate the requirement.

As confirmed by a review of key interfaces, most messages exchanged over the PTC communications network are protected in accordance with AAR requirements at the application level. One exception was identified and was in the process of being addressed. However, there are benefits in encrypting the radio frequency transport at the transport layer, in much the same way that virtual private networks (VPNs) are used to encrypt all traffic they carry.

The following areas were recommended for further consideration in Phase 2:

- NIST Lightweight Encryption Project Round 2 Finalists [4] (“Final 10”)
- Format Preserving Encryption
- Datagram Transport Layer Security
- Open-sourced AgilePQ Lightweight, Post-Quantum algorithms
- The ELLI lightweight asymmetric encryption algorithm
- Partnering with a subcontractor to evaluate lightweight encryption algorithms, including the NIST “Final 10” and additional algorithms

2.3 Phase 2

Candidate solutions were further evaluated in Phase 2, with a focus on implementation, migration, and deployment challenges. The authors considered whether solutions could be applied to radios, to ITCM, and/or to applications. The analysis identified several migration challenges, particularly for long and short broadcast messages, but also identified potential solutions for further investigation. The analysis also considered PTC security system constraints

and lessons learned from other initiatives and projects. During Phase 2, AgilePQ also shared information about their IoT encryption product SLiM.

Major findings and conclusions from Phase 2 included the following:

1. Prior to deployment, all aspects of deployment, phased migration, operation, and maintenance must be carefully considered, documented, and reviewed by stakeholders.
2. The solution should consider the needs of all PTC-enabled railroads impacted, from Class I railroads to the smallest roads, hosts, and tenants. Where possible, reference implementations and supporting tools should be developed to provide an integrated solution.
3. The solution should outline requirements, standards, and compliance mechanisms to ensure successful deployment in a reasonable timeframe.
4. The project should consider algorithms that are widely available, will be covered by recognized open standards such as NIST, and will have validated libraries available.
5. The solution should use ITCSM, which provides a strong foundation for security and represents a large, ongoing investment by the industry.
6. The solution should use ANDRO to help evaluate lightweight encryption algorithms focusing on performance.

2.4 Phase 3

Phase 3 identified the following technology gaps and areas for further research outside of the project:

1. Improving cybersecurity standards and governance for PTC
2. Developing a proof of concept (POC) prototype based on the ConOps to be delivered during Phase 4.
3. Extending the POC to develop reference implementations that may be used by large and small railroads to roll out new security features
4. Adding cryptographic validation to 220-MHz connections and associated Transport Network Updates (TNUs).
5. Exploring algorithms and primitives (e.g., asymmetric algorithms) not included in the NIST Lightweight Cryptography project, which is focused on symmetric encryption.
6. Evaluating the potential use of machine learning for access control policy verification (based on NISTIR 8360 [5]).

2.5 Other TAG Input

During the project, the researchers also reviewed identity assurance and Zero Trust concepts based on input from the TAG and focusing on National Security Agency and NIST material. [6] This section captures some key points from the review.

2.5.1 Zero Trust

- Per John Kindervag, the creator of Zero Trust concepts, "The hallmark of zero trust is simplicity. When every user, packet, network interface, and device is untrusted, protecting assets becomes simple." [7] Zero Trust therefore eliminates implicit trust in any one element, node, or service.
- Traditional network security prioritizes preventing security breaches, whereas Zero Trust assumes a breach is inevitable and has occurred.
- Zero Trust requires that organizations or systems never trust, always and continuously verify, authenticate, and explicitly authorize to the least privilege.
- Zero Trust implies comprehensive security monitoring and granular risk-based access controls. [8]

2.5.2 Identity Assurance

- NIST Digital Identity Guidelines are captured in the SP 800-63 series. [9]
- The guidelines are based upon the concept of credential service providers, which perform certain enrollment, identity proofing, and issuance processes. Identity assurance roles are used, consisting of applicants, subscribers, claimants, verifiers, and reliant parties (Figure 2).

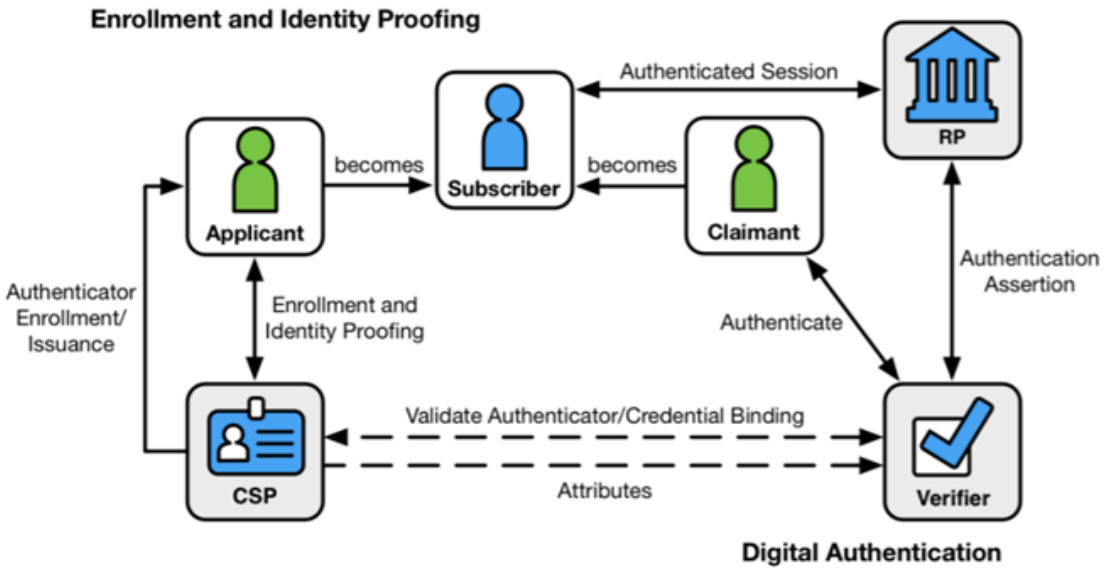


Figure 2. NIST SP 800-63 Identity Assurance Roles

- There are three identity assurance levels and three authenticator assurance levels. Depending on the levels, physical presence for identity proofing, multi-factor authentication, and the use of hardware-based cryptographic authenticator that resists impersonation may be required.

2.5.3 Considerations for Mapping These Concepts to PTC

- Sender and receiver identity verification, considering the following questions:
    - What is the source of their identity?
    - How is it authenticated and verified?
    - Can it be spoofed?
    - How is identity established and maintained throughout the lifecycle?
- Applications would remain responsible for end-to-end encryption.
- For devices, identity assurance could include validating serial numbers and tying them to an ID used by the device. This may require a human in loop step for identity proofing.
    - When a device sends a public key to a receiver, it does not provide identity assurance, even with a certificate. A root of trust is needed.
- Admission control for connection to the PTC Communications Network could be useful, but would need identity assurance.
- The ITC committee has not identified requirements for confidentiality except crew login.
- Certificates allow for a distributed trust model in which a compromise does not impact the entire industry.
- The use of sessions can introduce security issues [10], and maintaining sessions as locomotives transition between bases or travel through difficult terrain is challenging. However, sessions are less problematic for wayside-to-base connections. Current forms of wayside-to-locomotive communications are connectionless, as they should remain.
- Adding identity to lightweight elliptic curve seems to have potential.
- Most legacy devices do not have a hardware root of trust for identity (e.g., trusted platform modules), hence the project should not depend on them.
- Confidentiality is required when location information is associated with other sensitive information (e.g., commodity or shipper information). Per TAG input:
    - Confidentiality is required on all transports (i.e., cell and radio).
    - It is easier to manage at the application layer (i.e., for forward and backwards compatibility).
    - If encryption is implemented at the transport layer, it may still need to be implemented at the application level.
    - Encryption cannot be implemented without identity and authentication.
        - The important piece is associating keys with identity .
    - A solution using asymmetric key pairs instead of shared keys for waysides is attractive because they provide better encryption integrity.

3. Symmetric and Asymmetric Algorithm Evaluations by ANDRO

This section summarizes the methodology and findings from ANDRO's activities as documented in their final report [11]. ANDRO partnered with MCC from Phase 2 of the project, focusing on implementation and testing of lightweight encryption and asymmetric key establishment. The scope of work was centered on evaluation of the 10 NIST Round 2 finalists [4] from its lightweight encryption competition and began with a literature review. These algorithms were designed for low complexity platforms. In tandem, ANDRO investigated lightweight asymmetric key approaches enabled by elliptic curve cryptography (ECC), specifically, the ELLI protocol that combines a traditional Diffie Helman interaction with ECC mathematics.

3.1 NIST Lightweight Encryption Algorithms

This section reviews the NIST algorithms at a high level, including inputs/outputs of the algorithms, and NIST's submission criteria and performance measurements.

NIST submission packages included documentation, source code, test vectors, and intellectual property disclosures. Algorithms were required to accept variable length plain text and associated data in conjunction with fixed nonce and key lengths. They were also required to be deterministic and testable by encrypting known plain text and comparing against the resulting ciphertext.

NIST's general criteria for the lightweight algorithm competition focused on balancing design goals, target devices' physical constraints, and security capabilities [12]. Security considerations included minimum security strength (in bits), attack models, and side-channel resistance. Submitters were required to include statements about expected security strength, including rationale. ANDRO's analysis focused on implementation and time/cycle count metrics and did not assess security level since this is expected from third parties.

The NIST algorithms fall under the category of authenticated encryption with associated d ata (AEAD) as defined by RFC5116 [13]. These algorithms support both confidentiality and authenticity of data. Some of the NIST algorithms also included additional hashing functionality. AEAD algorithms have four inputs and one composite output. Inputs include:

1. A key, assumed to be symmetric and pre-provisioned
2. A one-time use number or nonce
3. Plain text: The part of the message to be encrypted
4. Additional authentication data (AAD or AD): Non-encrypted data

The output cipher text (CT) is the encrypted plain text (PT), which is the same size as the input plain text with an associated data authentication tag appended for authentication functionality. The size of this tag is user configurable. NIST's documentation indicated that a tag size of at least 64 bits is required. The exercise software downloaded with the algorithms used a tag size of 256 bits.

The key point is that the increased overhead cost of using the AEAD algorithm is primarily attributed to the extra bits associated with the tag. The nonce was not considered in any overhead assessment. Figure 3 shows the input and output using the first test vector in the NIST exerciser.

In this case, plain text and associated data are both empty. The output is 32 characters, or 256 bits, which is comprised solely of the tag.

```
Count = 1089
Key = 000102030405060708090A0B0C0D0E0F10111213
Nonce = 000102030405060708090A0B0C0D0E0F
PT = 000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F
AD = 000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F
CT = CC4E07E5FB13426EFFD17B0F51A6A830BF484C9651D77679971E8EB4A8EDB5A00782A94C72B2B02D87DCF4AF75DB6996
```

Figure 3. Example Input and Output of Test Vector with PT and AD as Non-Zero

Figure 4 shows a similar test vector, but in this case PT is non-zero. The output shown in the CT = line includes the encrypted text with the tag appended.

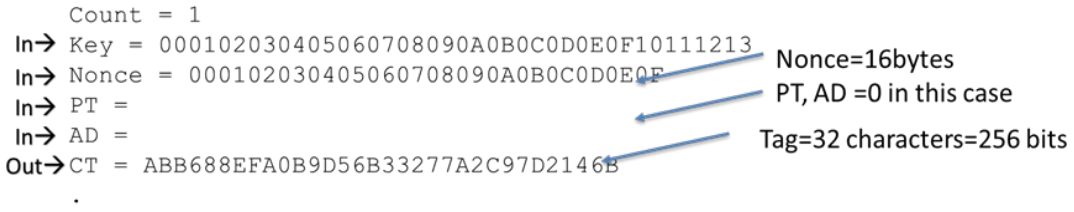


Figure 4. Example Input/Output of NIST Test Vector with PT and AD Set to 0

3.1.1 NIST Test Bed System Model and Evaluation Methodology

ANDRO used the NIST test exerciser with some modifications to exercise algorithms [14]. The test exerciser generated 1089 entries using the cipher text (CT), plain text, associated data, nonce, and key as parameters. Each entry represented different combinations of plain text and associated data. ANDRO's initial test bed was validated against results obtained by NIST for the top 10 shown in Table 2.

Table 2. Algorithm Submissions Tested and Reference Websites [4]

Algorithm	Directory Name	Website
ASCON	Ascon128v12	https://ascon.iaik.tugraz.at/
Elephant	Elephant200v2	https://www.esat.kuleuven.be/cosic/elephant/
GIFT-COFB	Giftcofb28v1	https://www.isical.ac.in/~lightweight/COFB/
Grain-128AED	Grain128aedv2	https://grain-128aead.github.io/
ISAP	Isapa128av20	https://isap.iaik.tugraz.at/
PHOTON-Beetle	Photonbeetleaead128rate128v1	https://www.isical.ac.in/~lightweight/beetle/
Romulus	romulusn	https://romulusae.github.io/romulus/
SPARKLE	Schwaemm256128v2	https://sparkle-lwc.github.io/
TinyJambu	Tinyjambu128v2	https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-submissions/tinyjambu.zip
Xoodyak	Xoodyakround3	https://keccak.team/xoodyak.html

Performance metrics included both elapsed time and count of cycles for encrypt and decrypt , as shown in Figure 5.

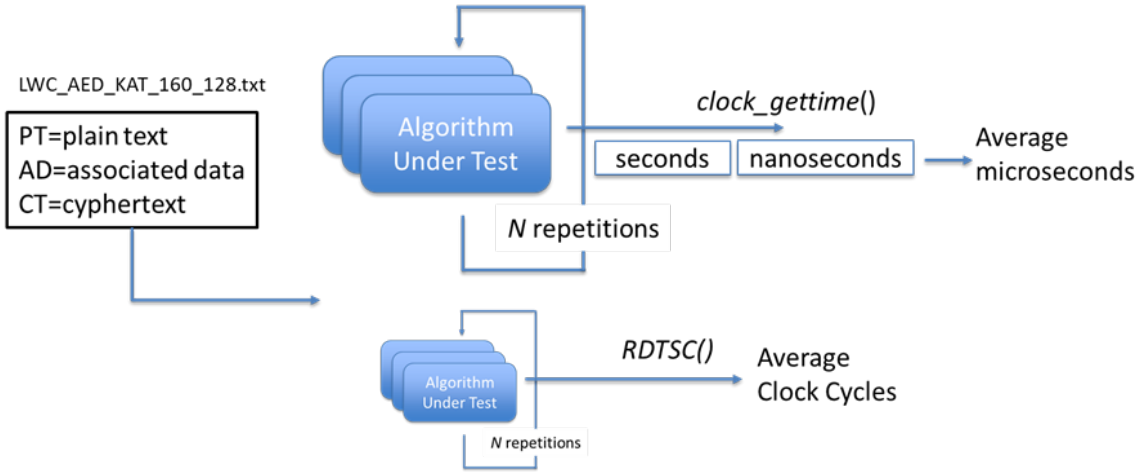


Figure 5. Testing Methodology

The test exerciser allowed for testing at varied plain text and AD sizes. Initially, the AD was fixed at 16 bytes and plain text size was set to 0, 8, 16, 24, and 32 bytes. Each of the points in Figure 6 represents the average of 20 encrypt times. This initial test showed that photon and grain performed the slowest.

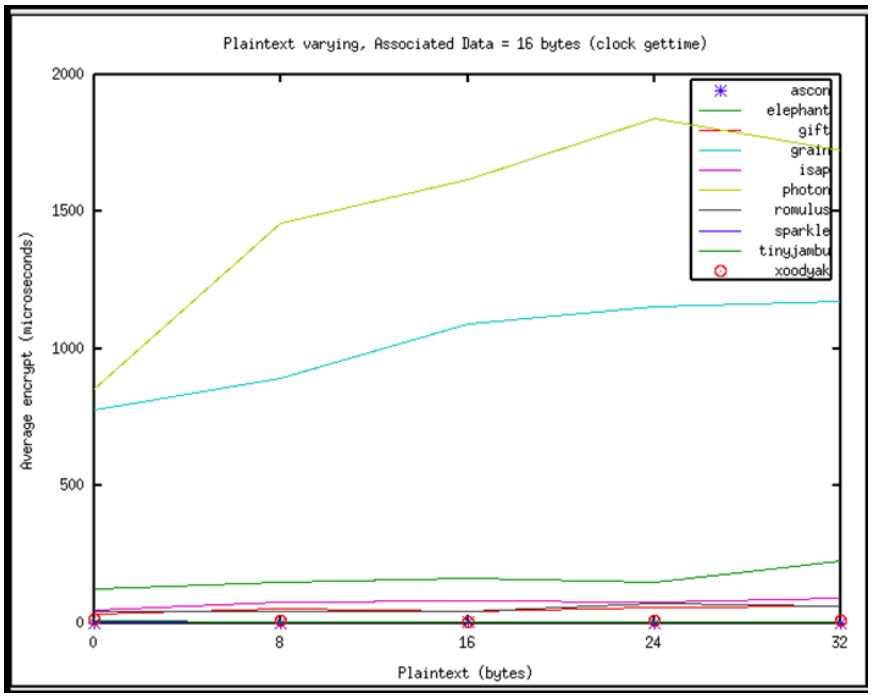


Figure 6. Average Encrypt Times, Varying Plain Text, and Constant Associated Data

The testing then focused on measuring clock cycles by holding both PT and AD constant and presenting the data in bar chart format. Figure 7 displays all 10 algorithms' primary variants. Each was exercised with plain text and had its associated data sizes held constant at 16 bytes. This graph also represents the average of 20 executions. Photon and grain were again the worst performers because they require the most clock cycles to perform encryption and decryption.

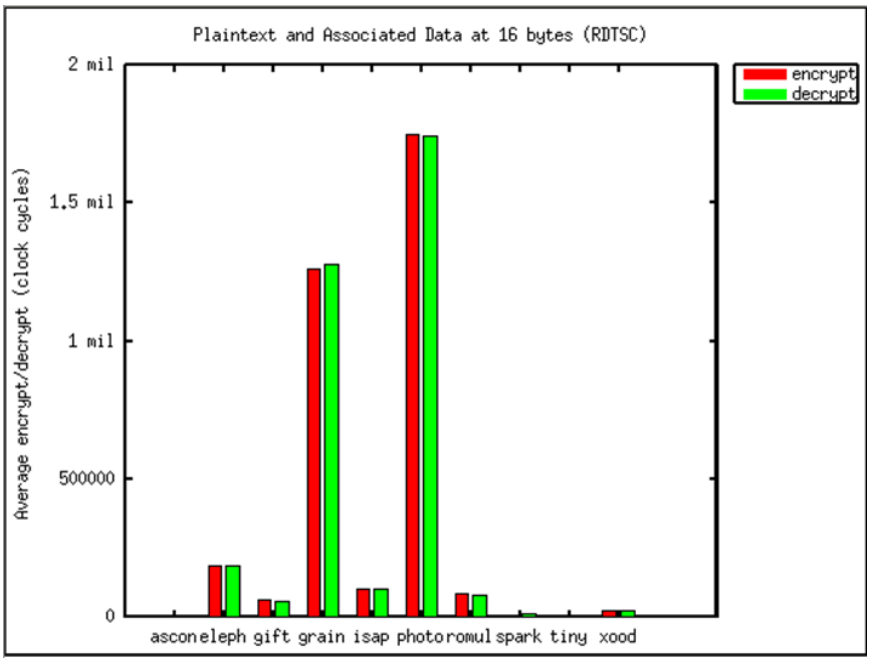


Figure 7. Average Encrypt/Decrypt Times, Constant Plain Text, and Associated Data

3.1.2 Comparing Performance on a Low-Complexity Platform

Table 3 summarizes initial performance metrics for the 10 algorithms on an Intel i5 laptop. The Intel laptop does not represent the limited computation capabilities of legacy devices in the field. Similarly, NIST's data was measured on a low-complexity Cortex-M0+. ANDRO identified a Raspberry Pi model B+ with an ARMv6Z processor, 512 MB RAM, and a 700 MHz clock as a similar device with low risk for porting the algorithms.

Values were calculated as follows:

1. Average encrypt/decrypt times were generated using 16 bytes each for PT and AD.
2. Throughput values were derived from dividing the combined PT and AD data values (32 bytes) by the average encrypt/decrypt time.
3. Each algorithm's clock cycle metrics were calculated by dividing the algorithm's clock cycle count by its combined PT and AD size (32 bytes).
4. The processor utilization values for each algorithm were retrieved by using the Linux perf tool [15].
5. Static footprint metrics for each algorithm were listed (e.g., the amount of memory used, number of lines of code, source code size, and compiled code size).
6. The security metrics (in bits) listed for each algorithm have been taken from the algorithms' corresponding specification literature.

Later, ANDRO evaluated the algorithms on a lower complexity platform and with a larger plain text size. Results showed an expected slowdown in execution time on the Pi compared to the Intel processor. Figure 8 compares the encrypt/decrypt times with 16 bytes of PT and

AD on the Intel and Pi Platforms. In general, algorithms that were slow on the Intel platform were also slow on the Pi.

Table 3. Initial Summary of Evaluation Metrics on Intel i5 Laptop

Metrics	ascon	elephant	gift	grain	isap	photon	romulus	sparkle	tinyjambu	xoodyak
Time										
Average encrypt (µs) PT, AD=16 bytes	3.9771	162.36	43.652	1089.9	81.945	1616	43.339	5.8194	3.362	7.1119
Average decrypt (µs) PT, AD=16 bytes	3.9992	160.18	45.077	1082.9	78.693	1693.8	44.24	5.7445	3.3394	8.1125
Clock Cycles										
Average encrypt (clock cycles) PT, AD=16 bytes	5,263	187,850	61,877	1,259,400	101,430	1,746,700	87,951	10,011	4,537	25,642.0
Average decrypt (clock cycles) PT, AD=16 bytes	5,260	185,240	56,960	1,278,800	101,290	1,745,400	80,747	12,343	6,377	24,729.0
Throughput (assume Bytes=AD+PT)										
Average encrypt throughput (bytes/µs) PT=AD=16 Bytes	8.046	0.197	0.733	0.029	0.391	0.020	0.738	5.499	9.518	4.500
Average decrypt throughput (bytes/µs) PT=AD=16 bytes	8.002	0.200	0.710	0.030	0.407	0.019	0.723	5.571	9.583	3.945
Speed (same metric used by NIST) Assume bytes=AD+PT										
Average encrypt speed (cycles/byte) *compare against NIST	164	5,870	1,934	39,356	3,170	54,584	2,748	313	142	801.31
Average decrypt speed (cycles/byte) *compare against NIST	164	5,789	1,780	39,963	3,165	54,544	2,523	386	199	772.78
Static Footprint										
Number of lines of code (lines)	431	7953	421	516	525	665	735	979	201	997
Size of source code file (bytes)	48 k	272 k	20 k	20 k	36 k	36 k	48 k	44 k	12 k	60 k
Compiled file size (genkat_aead_gettime, bytes)	17,832	18,120	17,984	18,104	18,352	18,608	22,680	22,576	13,448	19,384
Security										
Bits (from documentation)	128	127	n/a	128	128	121	128	120	n/a	128

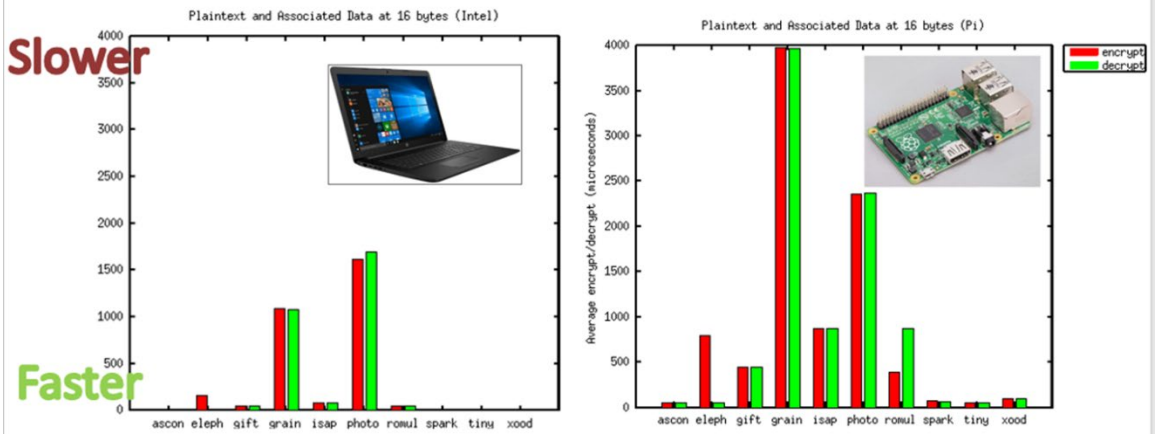


Figure 8. Comparison of Encrypt/Decrypt Times on Intel vs. Pi Platforms

3.1.3 Algorithm Benchmarking on Pi with Large Plain Text Size

ANDRO repeated some testing using a significantly larger PT size of 1.5 MB to assess changes in the relative performance of the algorithms. Figure 9 shows the corresponding encrypt/decrypt times on a Pi platform and allows for the following observations:

1. Grain and photon were the slowest 16 bytes of PT and remained the slowest with the larger plain text size.
2. On the Pi, ISAP went from being among the slowest performers to being more closely aligned with the fastest performers.
3. Ascon had been consistently one of the fastest algorithms in previous testing. Here, its speed was more in line with ISAP and Sparkle.
4. Some algorithms that had been considered for removal showed improvements under larger PT size, so they are considered borderline.

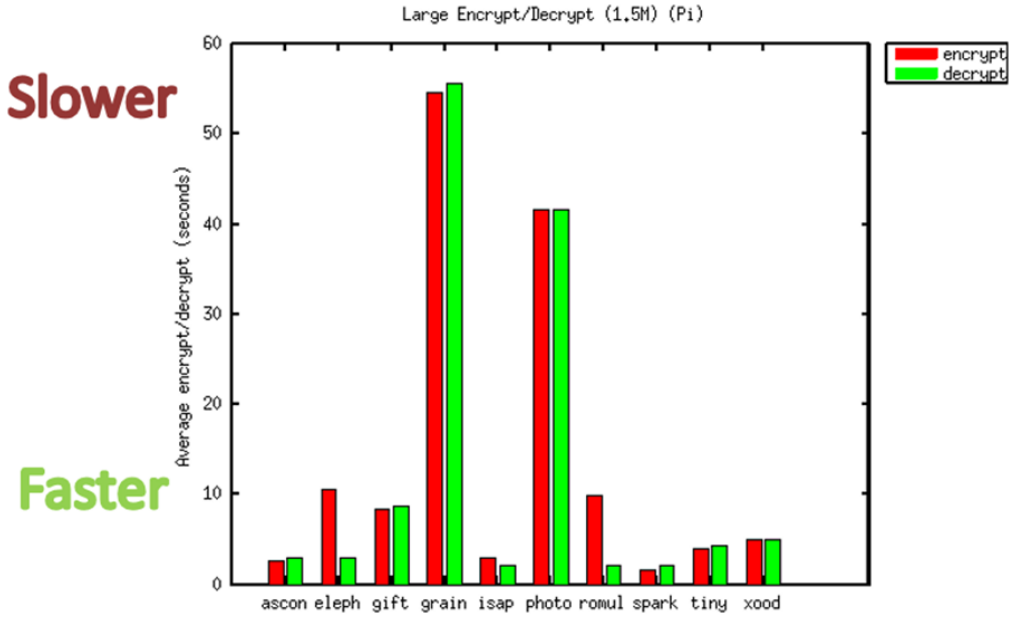


Figure 9. Large Encrypt/Decrypt on Pi B+

3.1.4 Initial Down-Selection

After the benchmarking, the research team made the following determinations:

1. Elephant, grain, and photon consistently showed slower performance across the different platforms and message sizes and were therefore eliminated from further consideration.
2. Gift, ISAP, and Romulus were considered borderline given their slightly slower speeds.

Note that NIST lists elephant as the slowest algorithm [16], and it has the most lines of code and the largest file size of all the algorithms (see Table 3). Elephant was selected by NIST because it supports parallelization, [17] but this is not a significant advantage for lower-complexity platforms with limited cores. Figure 10 summarizes the down selection.

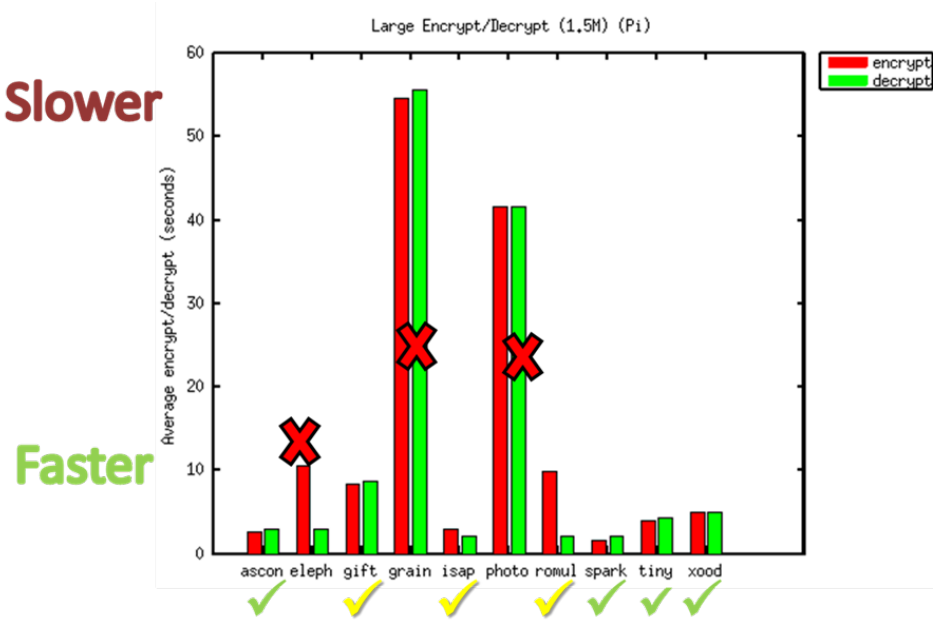


Figure 10. Summary of Initial Down Selection and Remaining Algorithms

3.1.5 Final Selection

The research team performed additional testing emulating WSMs. The WSM fields were mapped to the four key inputs of the lightweight encryption algorithms: the PT to be encrypted, the associated data, the key, and the nonce. The output includes CT and additional platform performance metrics, such as cycles and time. Additional configuration parameters included the size of the tag, which impacts final overhead costs. The NIST test exerciser configured the tag to 256 bits.

Operational constraints (e.g., ensuring the WIU Identifier field is always sent in the clear) were factored in when exercising the NIST algorithm on an EMP message. The associated data benefits the authentication capability of these algorithms but does not impact confidentiality or overhead. Finally, the encryption/decryption process was repeated 100-150 times/second to replicate worst-case conditions of message transfers for each test case. A packet capture for a WSM is shown in Figure 11, in which:

1. The EMP message body is shown in the red box.

2. The WIU Identifier bytes are highlighted in blue and sent in the clear (i.e., not encrypted).
3. Device status array, indicating states of signals and switches, is shown in the green box.
   a. The team encrypted 9 bytes of the device status array (PT= 9 Bytes).
   b. They allocated 4 bytes for AD.
4. The cyclic redundancy check (CRC) is shown in the yellow box.
5. The vital data integrity value is shown in the purple box.

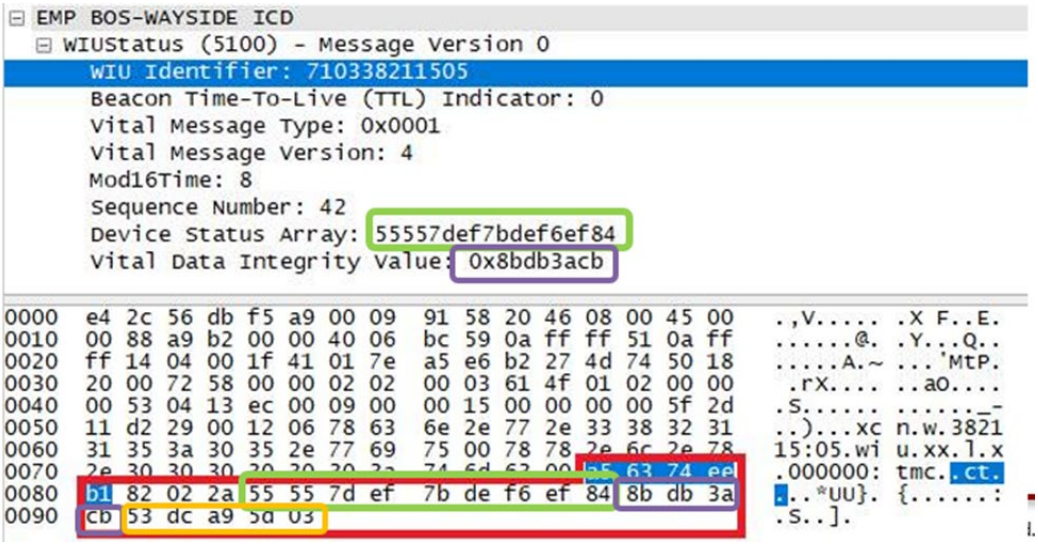


Figure 11. Wayside Status Message Packet Capture

The NIST algorithms were exercised with this configuration on the Intel laptop and the Pi testbeds. Results are shown in Figure 12 and Figure 13. Note that Romulus failed to fully execute on the Pi platform in this configuration.

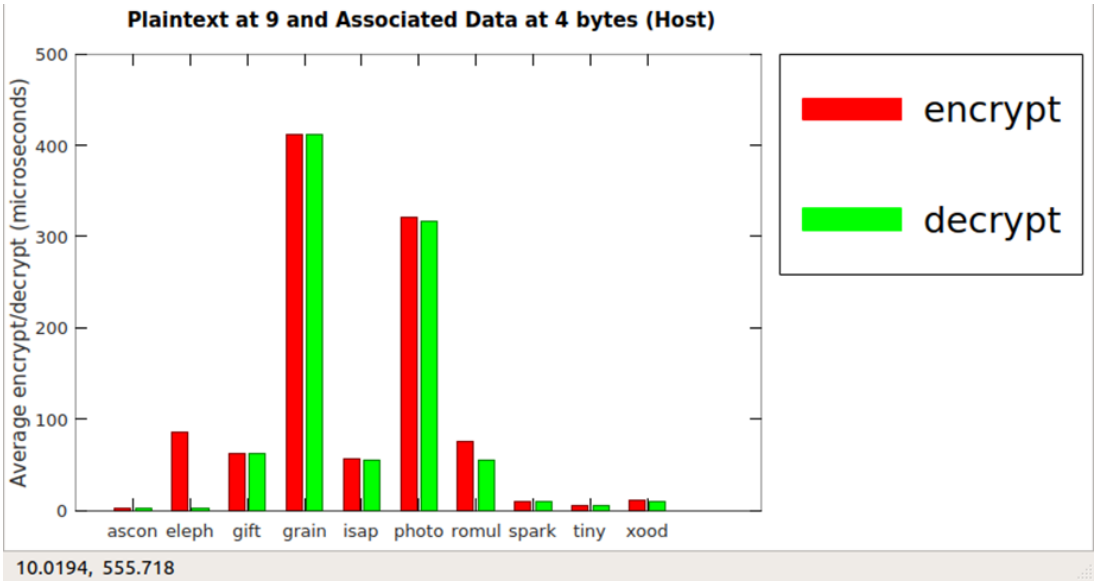


Figure 12. Algorithm Results on Intel Platform

The results show that Ascon, Spark, TinyJAMBU, and Xoodyak are the fastest algorithms; all four algorithms are recommended. Since their application programming interfaces (APIs) are the same and their performance is similar, using all four should be considered, though some may be eliminated from the NIST recommendations in the future.

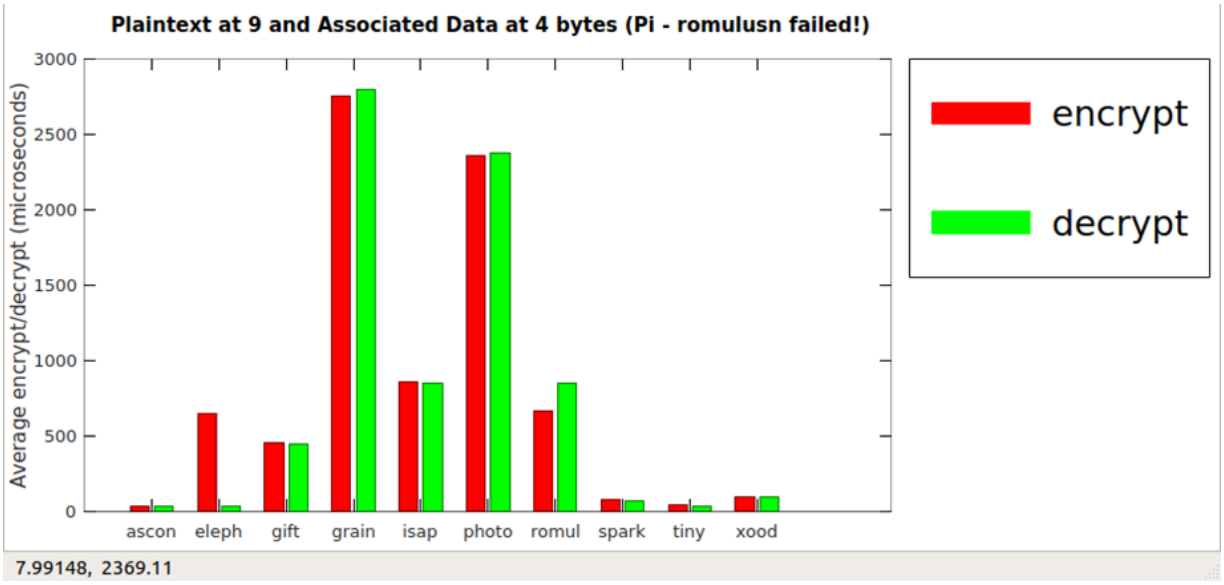


Figure 13. Algorithm Results on Pi Platform

3.2 Elliptic Curve Cryptography

The NIST lightweight encryption algorithms assumed symmetric key operations in which each side has the key in place. Changing keys is challenging, especially in the highly distributed and remote locations of PTC end points. Asymmetric or public key encryption approaches may be useful to investigate in the future. This section discusses ECC, which is an efficient public key approach, and ELLI, which is defined in ISO/IEC 29192-4, Amendment 1 [2].

ECC relies on a one-way intractability and uses a unique form of mathematics based on a known equation of an elliptic curve and a starting point on that curve. The basic concept of ECC operations is that a line drawn between two points on a curve intercepts the curve at a third location. This point is reflected across the X-axis to identify a new point on the curve or to the final location on the curve. Both methods are represented in Figure 14, where P is the starting point, ε is a large number, and A is the result of A= εP. ECC may also be illustrated using planar slices of a three-dimensional curve [18].

The Diffie Helman approach is used to establish a shared secret between two parties, and can involve multiplication by two large numbers, or use the ECC mathematical basis. Figure 15 illustrates this exchange between two parties (Alice and Bob). They already know the curve and starting point. They each have private keys, which are the number they multiply by the starting point. The numbers in the illustration are just examples.

During the information handshake, Alice sends her public key (point A) to Bob. Bob uses point A with his private key to identify a new point C. Bob shares both his public key B and point C back to Alice. Alice uses Bob's public key and her private key to verify that point D that matches

point C. In this case, C is the shared secret that can then be used in conjunction with another asymmetric encryption algorithm.

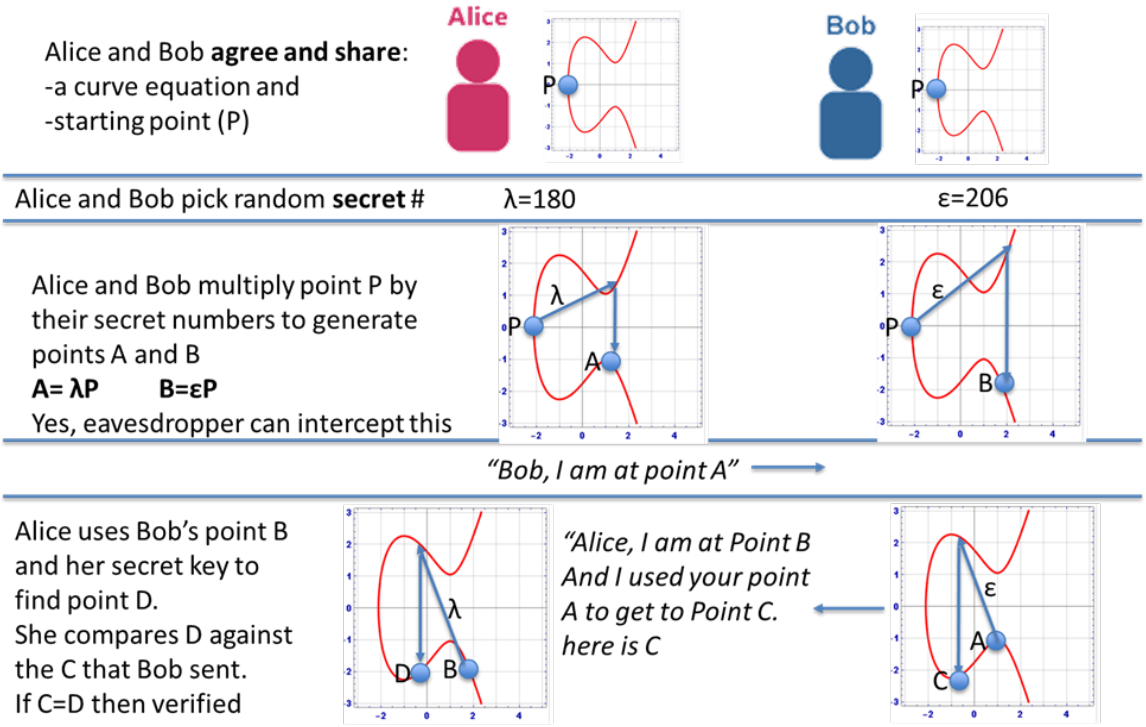


Figure 14. ECC-Based Diffie Hellman Establishment of Shared Secret

One of the metrics used in comparing different asymmetric key approaches is the size of the public key (in bits) required to achieve a level of security (also in bits). Table 5 shows the benefits of ECC over RSA in terms of the size of the public key required to achieve a target security level (in bits).

The benefit to the PTC use case is related to the overhead or extra bandwidth associated with transmitting information across the PTC communications links. With the ECC approach, a smaller amount of information is required to go between two nodes for a given level of security.

Table 4. Comparing ECC to RSA [19]

Security Bits	Symmetric Encryption Algorithm	Minimum Size (bits) of Public Keys	
		RSA	ECC
80	Skipjack	1024	160
112	3DES	2048	224
128	AES-128	3072	256
192	AES-192	7680	384
256	AES-256	15360	512

3.2.1 ELLI Algorithm

The ELLI algorithm is essentially a Diffie Hellman ECC exchange designed for radio frequency identification (RFID) tags with low computational power. The ELLI algorithm also incorporates a signature of the public key to establish some level of trust that an endpoint is genuine. This

involves transmitting a certificate and establishing the public key between the two nodes, in this case the reader and RFID tag. This impacts the overhead across a link and could negate the original benefits of ECC over RSA and the smaller key size needed. Figure 15 shows the ELLI algorithm. Here, A and B are 512 bits long.

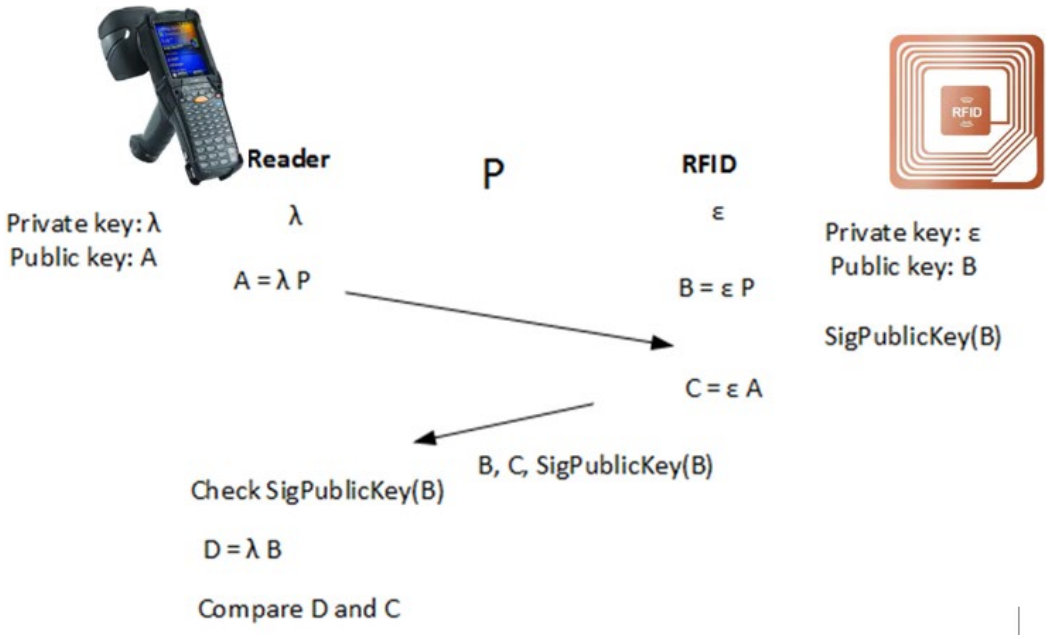


Figure 15. ELLI Algorithm [20]

Figure 16 provides more explicit detail of the ELLI algorithm and handshake of information between two nodes.

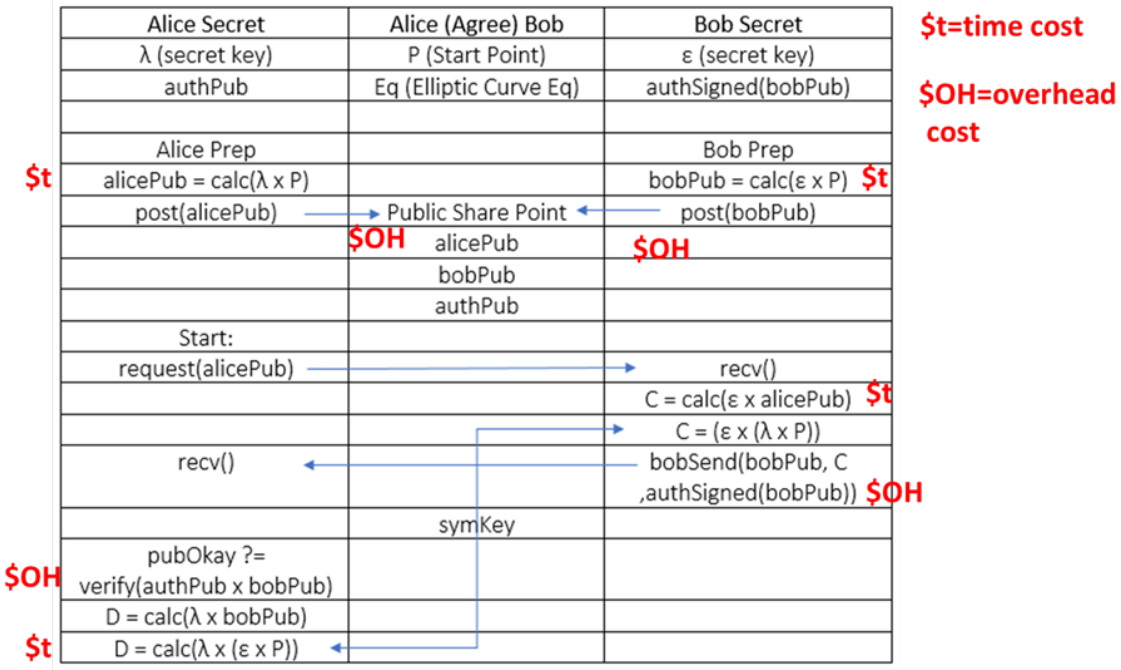


Figure 16. Explicit Steps for ELLI

Two types of costs associated with using the algorithm are considered. One is overhead, or additional information that is transmitted between two locations. The second is computational costs in terms of time related to processing the elliptic mathematics. These are indicated as $OH and $t. This figure is based on reviewing the ELLI documentation and video presented by Buchanan [20].

3.2.2 ELLI Implementation

ANDRO replicated the execution of the python code provided by Buchanan [20] and later implemented the algorithm in C++ using Open Secure Socket Layer (OpenSSL) library. These libraries included ECC mathematic functions and the creation of certificates of public keys. They also attempted to track time stamps to quantify the time cost for performing ECC mathematics, however, they did not complete this work.

3.3 Conclusions and Next Steps

The team successfully exercised the NIST lightweight encryption algorithms and down selected to four recommended algorithms. These are Ascon, Spark, TinyJAMBU, and Xoodyak. They replicated python code of the ELLI algorithm and created their own implementation, leveraging elliptic curve tools within the OpenSSL libraries. Areas for continued work include refining instrumentation of the ELLI algorithm to quantify time costs. A more representative testbed of PTC scenarios would better evaluate algorithms and ELLI. This could take the form of a POC prototype, including PTC message simulators and emulation of representative bandwidth-limited PTC wireless links.

4. Candidate Solutions

There are three categories of use cases for lightweight encryption on field devices in PTC (Figure 17).

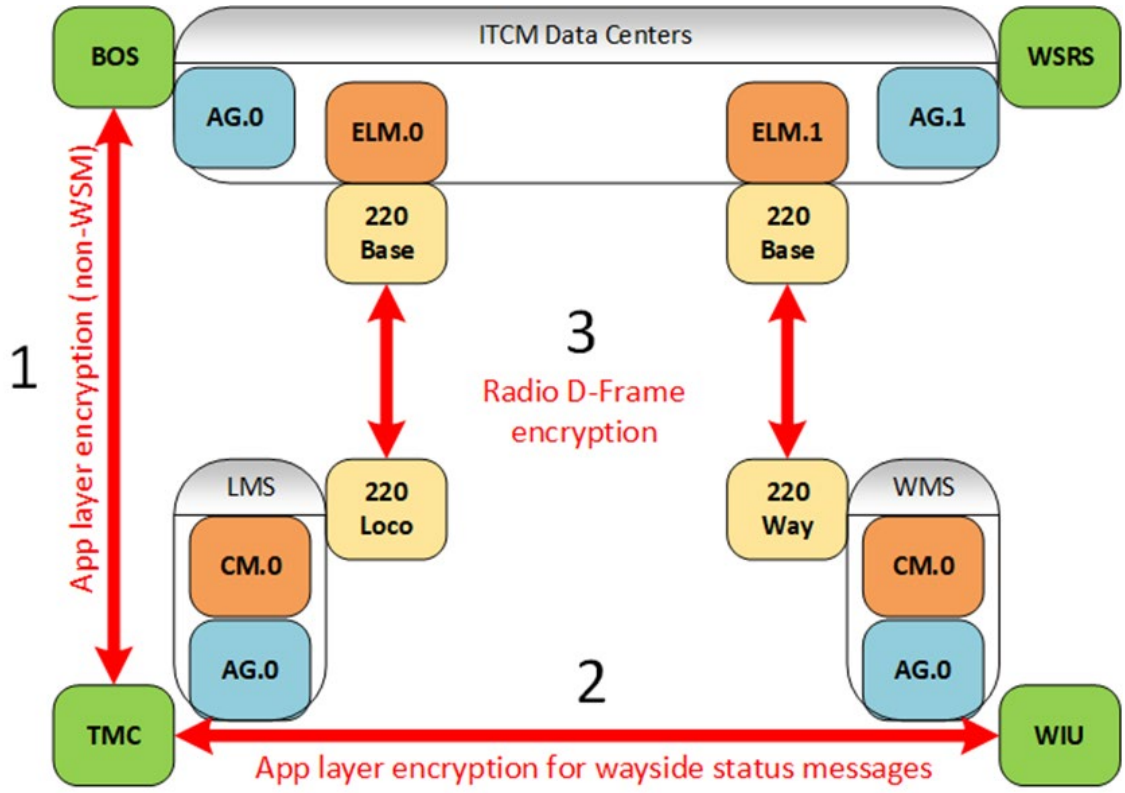


Figure 17. Categories of Candidate Solutions

For each candidate, this section captures:

1. The minimum communication security requirements and the security solution high-level design
2. A strategy for implementation and migration, accounting for interoperability requirements
3. An estimate of the impact of changes to the PTC network requirements
4. Measures to minimize and mitigate the impact of changes
5. Major risks and recommendations for risk mitigation

4.1 Candidate 1: Application-Layer Encryption (Non-WSM)

This section discusses non-WSM application-layer encryption. In this use case, lightweight encryption is applied at the application layer between the remotes and the BO. A typical application on a remote would be an SMA or the TMC on a locomotive, while the corresponding BO application would be an SMG or the Back Office Server (BOS).

4.1.1 Use Case Overview and Benefits

Figure 18 shows application-layer encryption (non-WSM).

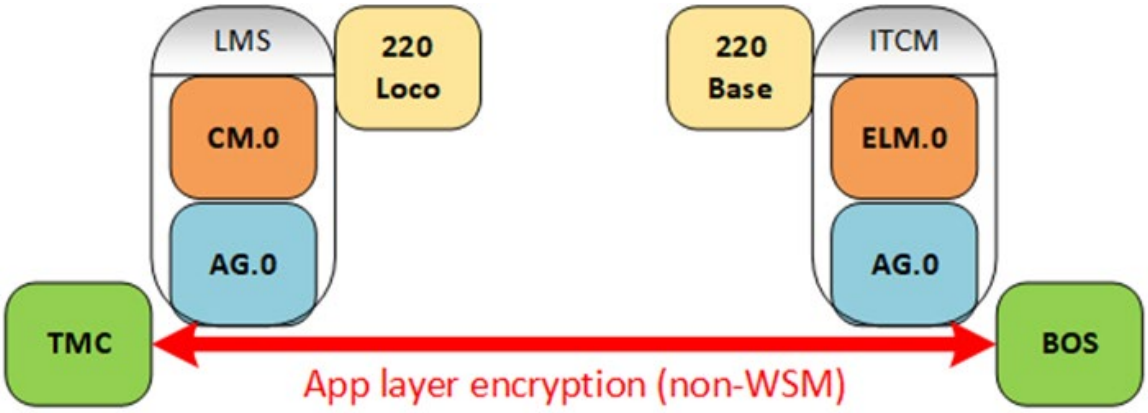


Figure 18. Application-Layer Encryption (Non-WSM)

This use case specifically excludes the WSMs and related peer-to-peer messages because they are a special case in which the messages can transit from remote to remote directly. Security for WSMs is covered specifically under Candidate 2.

This security may be selectively applied by either end application to different message types. It may replace or supplement security for existing PTC messages or secure additional message types to be added to PTC in the future. The messages may be encrypted in whole or in part.

4.1.2 Benefits

Depending on how lightweight encryption is implemented, the potential benefits of for remote applications (excluding WSMs) may include:

1. Reduced computation load for the remote application (TMC, WIU, and SMA) if one of the NIST finalist candidate algorithms is adopted
2. Better secrecy of private keys through reduced key sharing if an asymmetric technology such as ELLI is adopted

This approach benefits from performing cryptographic activities in the application layer (rather than downstream in the messaging or transport layers). Current PTC architecture design houses all application-level security (e.g., authentication and confidentiality) in the edges of the system. The end applications are in the best position to determine which messages (and which portions of those messages) need to be encrypted or signed. Data integrity fields in the EMP header and footer already exist, with sufficient flexibility to accommodate new cryptographic algorithms. These header fields, and EMP message body fields are populated by the applications at the edge of the system.

4.1.3 Requirements and High-Level Design

The following is a discussion of the requirements and the high-level design.

Protocol Requirements

The requirements for application security and data integrity in EMP messages are found in AAR S-9354 [21]. Cryptography for PTC messages may be implemented in three places in an EMP message by a sending application:

1. Within the EMP message body, such as the use of AES-128 Chain Block Cyphering to maintain confidentiality of employee PINs during locomotive initialization.
2. Within the EMP header, an encryption flag (bit number 1 in the flags field) may be set to indicate that the EMP body is encrypted. Note that this is not a service request; the flag is intended to inform the receiver about the encryption status of the payload generated by the sending application.
3. Within the EMP header, a data integrity flag (bits number 3 and 4) may be set to cover the entire EMP message (i.e., all header fields and message body). A corresponding data integrity value is then appended in the EMP footer:
    a. 0 = No data integrity support
    b. 1 = CRC, calculated using the CRC-32 algorithm
    c. 2 = Application-specific data integrity (i.e., HMAC)
    d. 3 = Reserved

In Figure 19, an application-specific security algorithm (in this case, a keyed HMAC signature) is used to calculate a data integrity value in the EMP message body, highlighted in green. If the message body were encrypted, the encryption flag in the header would be set, highlighted in blue. The data integrity field in the header is set, which calculates an integrity value for the EMP footer, both highlighted in red.

New encryption protocols using any arbitrary algorithm may be used in the message body of non-PTC applications if both sending and receiving applications are able to usefully verify or decrypt the message contents. The most significant requirement for non-PTC applications is to avoid overlap with PTC EMP message IDs.

PTC messages also require interoperability. For example, a locomotive initializing on another railroad's track must be able to communicate with the owning railroad's BO systems and the operating railroad's own servers. New encryption protocols for PTC operations are needed at an industry level to allow coordination between railroads and software vendors, ensuring compatibility.

There are three version and type fields in the EMP header relevant to the roll-out of new features, as shown in Table 5.

Table 5. Relevant Version and Type Fields in EMP Header [21]

Header field	Field size	Notes
Protocol version	8 bits	Version of the overall EMP header protocol, defined by each version of the EMP message format specification (i.e., S-9354).
Message type	16 bits	Application message ID, defined by the messaging specifications assigning unique message type IDs (i.e., S-9361); not to be confused with message number, which is an application-level sequence number.
Message version	8 bits	Version of the specific message type; valid versions of each message and its contents are defined by messaging specifications.

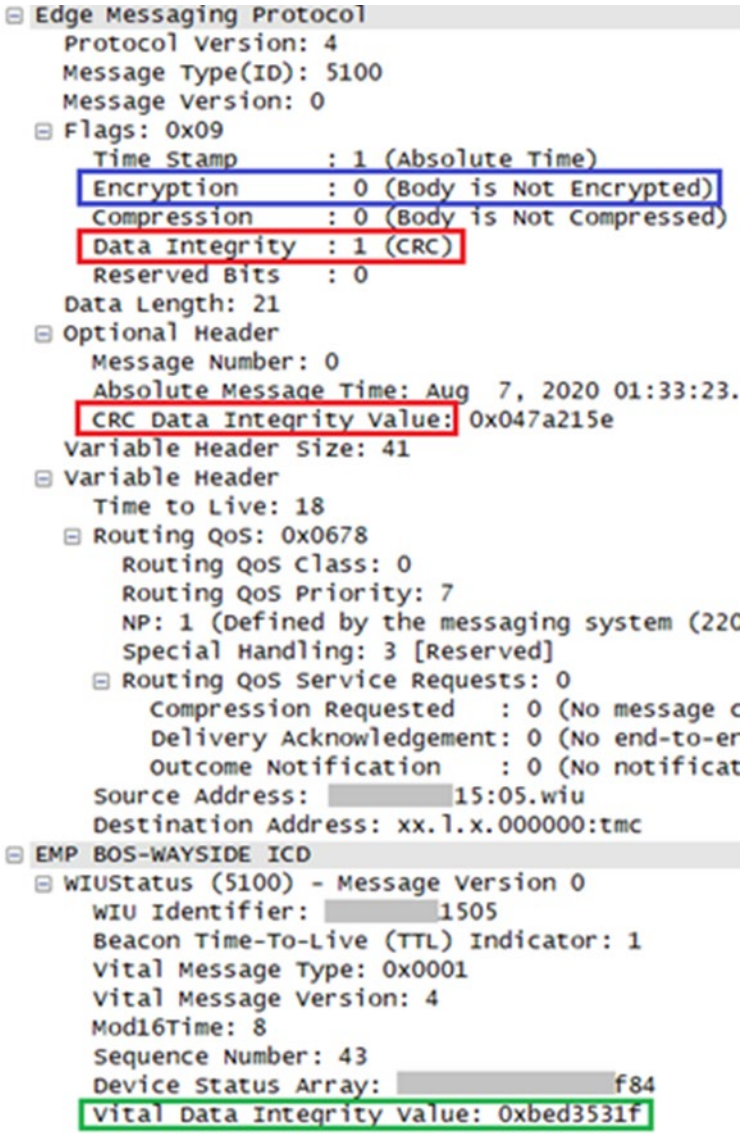


Figure 19. Wayside Status Message Header and Body

If the new security algorithms were introduced to improve the cryptographic strength or performance of existing PTC application messages, the Protocol Version and Message Type fields would be unchanged and the new message format would be defined in a revised standard under an incremented Message Version number. Alternatively, if new functionality is added (e.g., a new pair of request/response messages to exchange keys), the new messages should be assigned new IDs under the Message Type field.

If the qualitative and quantitative changes to the underlying EMP protocol rise to the level of a change in protocol, this large-scale change may be captured by the Protocol Version field. For example, changing the number of bits defined for certain header fields, or defining previously reserved field values, could necessitate incrementing the protocol version. This step must not be overlooked because it affects not only the end applications, but the internal messaging system as well.

Operational Requirements

The storage and exchange of cryptographic material and sensitive material on remotes must be conducted in a secure manner. Symmetric OPKs and asymmetric private keys are required to be encrypted when they are stored or transmitted.

When a remote communicates with a BOS using messages secured by symmetric keys (e.g., the locomotive OPKs used to create dynamic subscriptions in WSRS), there must be a secure mechanism for exchanging these keys. The key exchange service (KES) must be interoperable because the remotes may need to interact with the BOS of another organization. Care must be taken to ensure authenticity and authorization of the requesting party when exchanging keys because, at present, the symmetric OPK is the foundation of PTC security (see Section 2.1).

Replacing symmetric OPKs with lightweight asymmetric cryptography using public/private key pairs (e.g., ELLI) could considerably improve the security of private key material. The private key could be generated locally on the remote, and thus never leave the device. It could also be generated in a data center (to benefit from more powerful random number generators) and securely transmitted to the remote during a one-time installation process. To sign messages for authenticity, a remote would use its own private key, and any receiving party could verify the signature using the remote's public key. To encrypt, a remote could use the BO application's public certificate, which only the BO application could decrypt using its private key. Chains of authenticity could be established using certificate signatures. This approach would require certificate stores and the exchange of certificate revocation lists (CRLs), similarly to current TNU security implementations in ITCM.

The operational cost of switching to asymmetric cryptography for PTC applications is two-fold:

1. Significant discontinuity between the present OPK approach and the new asymmetric approach
2. Increased complexity and computation load on a resource-constrained remote

A hybrid approach combining the best features of symmetric and asymmetric cryptography may help mitigate both costs. For example, a remote running the new system could perform an initial setup handshake using its new asymmetric algorithms, with each side exchanging public certificates to establish a trusted relationship. After the initial setup exchange, a temporary symmetric OPK could be generated and shared using asymmetric encryption, allowing both ends to switch to faster and lighter symmetric encryption for multiple transactions of sensitive data.

The extra overhead of asymmetric operations only occurs at the beginning of the session, making the hybrid approach more attractive for its increased performance. This approach may also provide an easier migration path and interoperability with devices not yet upgraded to the new security algorithms because the system could use pre-shared keys in the event of incompatibility during the handshake process.

4.1.4 Implementation and Migration, Including Interoperability

Implementation of hybrid encryption for PTC applications would require the addition of a shared key negotiation phase for interactions between the remote and the BO. Shared key negotiation generally occurs during an initialization phase, such as at the beginning of a locomotive run. The following outlines its general process:

1. Both applications on either endpoint exchange pre-existing public application certificates, which are then signed on each side by a chain of trust (i.e., a root or issuing certificate).
2. The receiving application on either end verifies its counterpart's certificate against its local copy of the certificate master library, containing the signing certificates of all organizations the endpoint is prepared to trust. If the signature on the application certificate passes, shared key generation is next.
3. One of the endpoints generates a symmetric OPK, typically the application initiating communication (in PTC this is usually the remote). The length and format of this key may be identical to the OPKs currently used by PTC applications for compatibility purposes,. For new applications there may be more freedom in the selection of key size and format.
4. The generating endpoint encrypts the shared secret OPK with its own private asset key and its counterpart's public application certificate, then transmits it to the receiving endpoint.
5. The receiver's private application key is the only key which may be used to decrypt the outer layer of encryption, and this key is never shared. The receiver then uses the sender's public certificate to decrypt the second layer of encryption to retrieve the shared secret OPK, simultaneously verifying the authenticity of the sender application.
6. In response, the receiving application doubly encrypts the shared secret OPK, first with its own private application key, then with the sender's public certificate, and transmits it to the original sender.
7. The original sender decrypts the doubly encrypted shared secret OPK. If it matches the original OPK that it generated, this serves as acknowledgment and acceptance of the proposed OPK.

Migration Strategy and Interoperability

Once a shared secret key is established between the two endpoints, normal authentication or encryption operations between the two sides may proceed, as presently implemented for PTC applications. For the existing TMC, the shared secret OPK may be used in place of the current locomotive OPK for HMAC verification, using the same algorithms for compatibility purposes.

During migration, it is unlikely that all remote devices (e.g., the entire locomotive fleet, or the locomotives of partner companies) will be updated to the new security system simultaneously. This will result in upgraded BO applications communicating with remote devices that do not perform the shared key negotiation steps during initialization. For the duration of the migration, upgraded BO applications should fall back to using pre-shared locomotive OPKs for remotes not yet upgraded. This may be triggered by the lack of response to a certificate exchange request, or an older message version number in one of the initialization messages. The current interoperable KES may then be used to fetch the necessary keys.

Conversely, it is possible during migration for upgraded remotes to communicate with BO applications which have not been upgraded. The most likely scenario for this is in interoperable dispatching, when the locomotive of one company may need to communicate with the BOS systems run by a federated partner railroad. In this scenario, the upgraded locomotive can either:

1. Fall back to using a pre-shared locomotive OPK, generated and installed on the locomotive and BOS using the old security system
2. Generate a shared secret OPK under the new system and deliver it to an upgraded interoperable KES service which serves keys to the BOS

The interoperable KES should be one of the first servers upgraded during the migration because of its importance in bridging functionality during the migration process. The re-use of existing OPK formats and algorithms should ease the migration process to the new hybrid encryption system because it minimizes the software changes on the endpoint applications during normal operations. In effect, the new features primarily consist of an additional certificate exchange and key negotiation step (during locomotive initialization) and key rotation and renewal mechanisms to counter key compromise.

Key Compromise

The lifetime of the generated key should be defined so that it is short enough to limit the usefulness of brute force attacks but long enough to avoid excessive key renewal. Rolling keys too frequently could interfere with operations at a time-sensitive point, in addition to imposing computational and bandwidth overhead.

If a shared secret OPK becomes compromised, the shared key negotiation process may be re-started by either the remote or BO end. As soon as a new OPK is accepted, the older key is deleted and is no longer useful for signature or encryption purposes. In the event of a compromise of an asymmetric application private key, the accompanying public certificate may be added to a CRL so it can no longer be accepted for the purpose of establishing shared secret OPKs. Processes such as simple certificate enrollment protocol (SCEP) can generate and sign a new application public/private key pair. This process should not be fully automated because of the importance of the certificate signature in establishing a chain of trust. Human verification of device authenticity should be part of any certificate enrollment system.

4.1.5 PTC Network Impact

Encryption requires two additional message exchanges for negotiation: two-way certificate exchange, and request-acknowledgment for the OPK. These would impose additional computational and network overhead during locomotive initialization, and during key renewal.

The size of the application certificates would likely be comparable to the X.509 asset certificates for systems management and TNU security (4096 bytes). Shared secret OPKs would retain the same length as the locomotive OPKs they replace (20 bytes).

Asymmetric cryptography on the remote is anticipated to impose more CPU load than the extant symmetric operations. Only using asymmetric algorithms to establish shared secret keys should minimize this new load. The additional CPU load would likely not affect PTC operations, since this step would primarily occur during initialization, when the onboard systems are not busy processing traffic from wayside devices.

Using NIST lightweight algorithms for new apps would likely reduce CPU load, given their optimization for performance in resource-constrained environments.

Migration would also be complicated because shared secret OPK negotiation is not yet practiced. Until the BOS apps are upgraded, Interoperable KES would be heavily utilized to bridge the old and new key systems.

4.1.6 Risks and Mitigations

Table 6 shows the risks for Candidate 1, and their probabilities and mitigations.

Table 6. Candidate 1 Risks and Mitigations

Risk	Probability/Impact	Mitigations
Poor or one-way communications	Medium/medium	• Setting sufficiently long OPK validity time to avoid key renewal during poor communications conditions • Local storage of generated keys in a secure Key Store
No compatible KES	Low/medium	• Prioritize implementation of Interoperable KES, together with related functionality in BO dispatching servers.
Unintended security failures in rollout	Medium/high	• Extensive testing, including trials/pilots • For non-vital new applications, run the system in a non-enforcing mode until issues are resolved.
Unacceptable latency for setup process	Medium/medium	• Optimization of changes to minimize latency.

4.2 Candidate 2: Application Layer Encryption for Wayside Status

In this use case, lightweight encryption is applied to the set of peer-to-peer messages involved with the exchange of WIU status between waysides and nearby locomotives. Lightweight cryptography may be used to either augment or replace existing security mechanisms (HMACs derived and validated using OPKs).

4.2.1 Use Case Overview and Benefits

Figure 20 is an overview and discussion of use-case benefits and challenges.

Overview

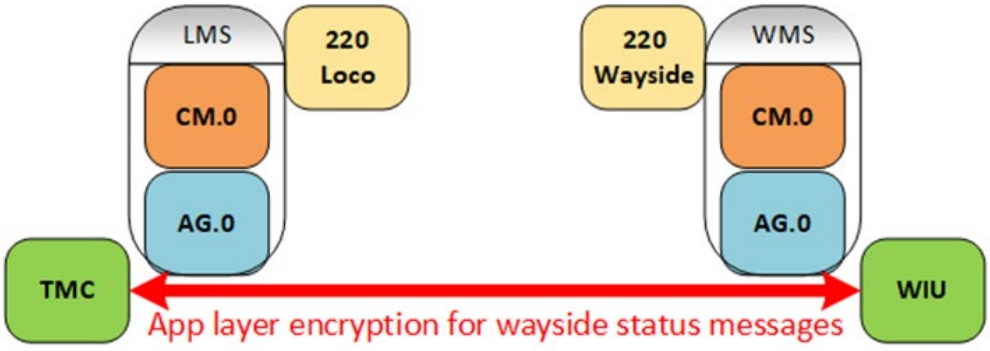


Figure 20. Application-Layer Encryption for Wayside Status

At present, these peer-to-peer messages consist of the following request/response pairs:

1. Beacon On (EMP message ID 5200)
2. WIU Status Messages (5100)
3. Get WIU Status (5201).

4. Get WIU Status Response (5101).

The use of lightweight cryptography to improve the security of wayside status messages can be grouped into three general approaches:

1. **Augment or Replace HMAC with Another Symmetric Algorithm:** The new lightweight algorithm (e.g., a NIST finalist candidate) may be used to sign or encrypt portions of the WSM payload. The new algorithm may improve efficiency, but this change introduces significant compatibility and interoperability issues. Moreover, because the new algorithm is still symmetric, this does not address the problem of widely shared secret keys.
2. **Augment or Replace HMAC with an Asymmetric Algorithm:** The new algorithm (e.g., ELLI) may be used to sign or encrypt portions of the WSM payload with a private key, while the receiving locomotive may verify or decrypt using the wayside's public certificate. Certificate signatures may be used to verify chains of trust between devices. This approach addresses the problem of widely shared secret keys since the wayside's private key never leaves the device. Unfortunately, this approach is prohibitively resource-intensive, both in terms of TMC computational load and 220-MHz radio bandwidth. It also has the same compatibility and interoperability issues as the previous approach.
3. **Use Hybrid Encryption to Generate and Share Wayside OPKs:** A lightweight asymmetric algorithm (e.g., ELLI) would be used as part of a hybrid encryption protocol to securely generate and distribute symmetric wayside OPKs compatible with the existing HMAC algorithm. Key sharing would use existing Interoperable KES and MDM infrastructure, potentially enhanced with peer-to-peer key exchange between locomotives and waysides in the field.

As with application-layer encryption for non-WSM traffic, the recommendation for securing wayside status messages is to pursue the hybrid encryption model, thereby obtaining the benefits of both the symmetric and asymmetric algorithms.

Although hybrid encryption could be used to distribute symmetric keys for wayside status message encryption (rather than just for HMAC verification), this approach is not recommended. Encrypting the entire WSM body would break messaging and radio features such as message filtering and base radio whitelisting. Even if only portions of the WSM body were encrypted, this could break compatibility and interoperability in the migration process. All locomotives and WSRS systems across interoperating companies would need to be upgraded to handle WSM decryption first, before turning on the encryption feature on any waysides. Consequently, encryption of the WSM body, in whole or in part, is not recommended.

Benefits

The hybrid encryption approach poses the least issues with compatibility and interoperability, and a straightforward migration path. It also addresses the issues with widely-shared, long-lived secret keys, since the lifetime of the generated key could be short enough to limit the usefulness of brute force attacks.

This approach has several advantages relating to performing cryptographic activities in the application layer (as opposed to further downstream in the messaging or transport layers). The

present design for PTC architecture calls for all application-level security (e.g., authentication and confidentiality) to be housed in the edges of the system. The end applications are in the best position to determine which messages (and which portions of those messages) need to be encrypted or signed. Data integrity fields in the EMP header and footer already exist, with sufficient flexibility to accommodate new cryptographic algorithms. These header fields and EMP message body fields (e.g., the vital data integrity value) are populated by the applications at the edge of the system. Moreover, most PTC security infrastructure (e.g. the interoperable KES) is currently implemented at the application layer of the system.

Challenges

Wayside status messages have several characteristics which make them a unique challenge for PTC security, including:

- **Vitality:** The WIU and TMC are the prototypical vital applications of the PTC system, and the WSM is especially important for safe operations. Problems with decrypting a WSM, or a delay in decoding such messages, can result in an enforcement, in which the PTC on-board system triggers full emergency braking force on the train. In some cases, false enforcements have resulted in damage to the track or the train. Any alteration to the security mechanisms for wayside status must be mindful of potentially extreme impacts. Note that ensuring the integrity of WSMs is a vital function that must be performed by vital applications (i.e., the WIU and the TMC).
- **Interoperability:** Changes to security must preserve interoperability, i.e., the ability of any train to communicate with the wayside equipment of any other company. Where software or protocols need to be upgraded, backwards compatibility must be maintained with equipment run by other companies with a different upgrade schedule. When keys need to be exchanged between endpoints, the mechanisms must also work between organizations.
- **Message size and volume:** Most messages flowing through the communication system are WSMs, especially for bandwidth-limited 220-MHz radio transports. Aspects of the radio system (e.g., the 30 milliseconds fixed-frame broadcast slots) have been optimized for the present size of WSMs to maximize capacity and performance. Any significant increase in message size for WSMs might be difficult to accommodate with existing radio configurations and hardware.
- **Computation load:** Similarly, the large number of WSMs received by a locomotive in wayside-dense locations (e.g., Chicago) could impose a large computation load penalty if the new security mechanisms involve additional CPU resources to verify or decode.
- **Time sensitivity:** The message validity Time to Live (TTL) on a WSM and on a Get WIU Status Response message is necessarily short, to avoid locomotives reacting to an out-of-date wayside status. As a result, the wayside to locomotive path is subject to the least tolerance for delay in the PTC system.
- **Peer-to-peer:** Broadcast messages between the locomotive and wayside may be sent and received peer-to-peer, without the BO or the base radio network. Any security system implemented for the wayside-to-locomotive path must consider that communication with the BOS may not be possible when the locomotive is communicating with the wayside.

Shared keys or other cryptographic material must be fetched and stored securely and locally ahead of time. Similarly, the system must account for asynchronous or one-way communication between the locomotive and the wayside.

- **Broadcast:** When a wayside sends its status over 220-MHz radio, the broadcasts may be picked up and acted upon by any locomotives within communication range. Peer-to-peer wayside status messages are not addressed to a particular locomotive receiver.
- **Filtering:** To reduce excessive duplication of WSMs in the radio network, a whitelisting feature was introduced in the base radios. Similarly, a WSM filtering feature was added to the locomotive messaging system to reduce delivery of unneeded WSMs. Both features utilize the WIU Advanced Train Control System ID field in the EMP message body for filtering. Changes to security should avoid encrypting this portion of the payload to avoid breaking these existing features. Although the research team supports filtering, changes to other PTC components are required before this can be deployed.

4.2.2 Requirements and High-Level Design

The following is a discussion of the requirements and high-level design for Candidate 2.

Performance Requirements

Current locomotive radio and messaging systems are rated to handle the maximum theoretical load for dense urban areas (DUAs), approximately 120 WSMs per second. Any additional CPU load imposed by a new security algorithm on the TMC must not jeopardize the ability of the onboard application to process this number of WSMs in real time.

Similarly, the WSRS must handle several thousands of inbound and forwarded WSMs per second. Any change to wayside security which impacts the WSRS must be assessed for the overall delay it may add to forwarded messages.

Bandwidth Requirements

The portions of the 220-MHz radio network which handle peer-to-peer communication between locomotives and waysides are:

- **The common channel** is a single 25-kHz channel used for Beacon On, Get WIU Status, and Get WIU Status Response messages. The common channel uses carrier sense multiple access/certificate authority to avoid collisions by multiple transmitters.
- **The F-frame slots** are multiple 30-millisecond slots on multiple channels, allocated only for WSMs. The F-frame slots use slot and frequency planning to avoid collisions by multiple transmitters.

Both the common channel and the F-frame on a wayside radio use differential quadrature phase shift keying half-rate modulation, resulting in a transmission speed of 16 kbps. This data rate, in combination with the 30-millisecond slot size, means each slot can accommodate slightly less than 500 bytes of data. The typical over-the-air size of a WSM is under 50 bytes, however, large security materials (e.g., X.509 certificates, at 4096 bytes), cannot be routinely transmitted over either the F-frame or the common channel.

Operational Requirements

The storage and exchange of cryptographic material and sensitive material on remotes must be conducted in a secure manner. Symmetric OPKs and asymmetric private keys must be encrypted when they are stored or transmitted. The symmetric wayside OPK is presently the foundation of PTC security between remotes. If an attacker acquires the OPK for a wayside, they could subvert the authenticity checks in the TMC vital application and intentionally send invalid messages to the locomotive. A hybrid approach combining the best features of symmetric and asymmetric cryptography may help address the issues with long-lived widely shared wayside OPKs, while mitigating the costs of a purely asymmetric encryption approach.

4.2.3 Implementation and Migration including Interoperability

Implementation of a hybrid encryption approach for securing wayside status messages would require the following changes:

- **Wayside and Interoperable KES Enhancements:** Upgraded waysides can use hybrid encryption to generate wayside OPKs and securely share them with the BO KES application. Conversely, it may be the BO application which generates the OPKs and then shares them with the wayside. To avoid issues with key expiration or rotation during a run, two OPKs must be generated and shared: current and pending. During normal key rotation, the pending key becomes the new current key, and a new pending key is generated. Note that current implementations of interoperable KES already have access to copies of wayside OPKs; the enhancement lies in the use of hybrid encryption to handle key rotation.
- **Subdiv File Enhancements (Current Locomotive Distribution Method):** In the current on-board TMC application, wayside OPKs are stored in the subdiv files, which are a locomotive's map of the track. Updated subdiv files are securely downloaded and stored by the TMC during the initialization process at the beginning of a PTC run. Mechanisms may need to be developed to copy new OPKs from the interoperable KES into the subdiv files downloaded by locomotives preparing to run through subdivisions with the upgraded waysides.
- **Interoperable KES to Locomotive (Probable Future Distribution Method):** One likely, future enhancement to PTC security is the separation of wayside OPKs from the subdiv file. In this scenario, hybrid encryption could be used for locomotives to establish a secure connection to the KES, and then fetch the necessary wayside OPKs that were previously shared by the waysides with the KES.
- **Peer-to-Peer Key Sharing (Optional Enhancement):** If the KES-based key distribution system fails, locomotives could possibly request OPKs from waysides. This feature would require two additional messages per OPK exchanged over the 220-MHz radio common channel:
    - Locomotive request for wayside certificate and OPK, with a copy of the locomotive certificate
    - Wayside response containing the wayside certificate, with the wayside's current and pending OPKs, encrypted first with the wayside's private key, then with locomotive's public certificate.

Peer-to-peer key sharing would impose a requirement for locomotives and waysides to maintain certificate stores and CRLs for establishing chains of trust, since both the locomotive and the wayside would need to determine the authenticity of the certificate signatures received in the above exchange. For ordinary PTC operations the KES-based approach should be sufficient to keep all parties up-to-date for the duration of a locomotive run. This is why the more complex peer-to-peer key sharing enhancement is marked as optional.

Migration Strategy and Interoperability

Once the wayside OPKs are shared between the wayside and locomotives running through the relevant subdivision, wayside status messages can be authenticated as presently implemented for the TMC application.

In the current implementation of PTC, the on-board systems' subdiv files maintain two OPKs per wayside, which minimizes issues with key rotation during a run. This mechanism should be preserved during the migration to a hybrid encryption security system to maintain this flexibility.

During migration, it is unlikely that all waysides will be updated to the new security system simultaneously. Under the KES-based approach, the interoperable KES would revert to obtaining wayside OPKs from extant key repositories or distribution mechanisms if they are not received directly from the wayside using hybrid encryption.

As long as subdiv files are kept up-to-date with the latest keys obtained by the KES, the MDM process for updating subdiv files on locomotives at the beginning of PTC initialization should work universally on existing and upgraded locomotives. If or when a migration to a direct KES-based key distribution process occurs, is the authors recommend retaining the functionality of the old subdiv-file mechanism during the transition, as a fallback for locomotives which have not yet been upgraded.

Key Compromise

As previously discussed for locomotive OPKs, the lifetime of wayside OPKs should be short enough to limit the effects of brute force attacks and long enough to avoid excessive key renewal.

In the event of a wayside OPK becoming compromised, the OPK negotiation process may be re-started by either the wayside or the KES. Once a new OPK is accepted, the current OPK is deleted and the pending OPK is promoted to its position, while the newly-generated OPK is stored in the newly-available pending slot.

In the event of an asymmetric application private key compromise, the accompanying public certificate may be added to a CRL, so it will no longer be acceptable for establishing shared secret OPKs. Processes such as SCEP may be used to generate and sign a new application public/private key pair.

4.2.4 PTC Network Impact

Hybrid encryption requires two additional message exchanges for negotiation: two-way certificate exchange, and request-acknowledgment for the wayside OPK. This would impose additional computational and network overhead on the wayside any time key renewal is conducted.

The application certificates would be comparable in size to the X.509 asset certificates used for systems management and TNU security (4096 bytes). Shared secret OPKs would retain the same length as the wayside OPKs they replace (20 bytes).

Asymmetric cryptography on the remote is anticipated to impose more Central Processing Unit (CPU) load than the current symmetric operations. Using asymmetric algorithms only to establish a shared secret key should minimize this new load. Use of lightweight asymmetric algorithms (e.g., ELLI) may reduce the computational load of signature encryption operations, but the underlying size of the keys and certificates, and their impact on network bandwidth, would be about the same as traditional X.509 certificates.

4.2.5 Risks and Mitigations

Table 8 shows risks, probabilities, and mitigations for Candidate 2.

Table 7. Candidate 2 Risks and Mitigations

Risk	Probability/Impact	Mitigations
Poor or one-way communications	Medium/low	Setting the OPK validity time sufficiently long to avoid key renewal during poor communications conditions.
No compatible KES	Low/medium	Prioritize implementation of Interoperable KES, together with related functionality subdiv file update mechanisms.
Unintended security failures in rollout	Medium/high	Extensive testing, including trials/pilots
Unacceptable latency for setup process	Medium/medium	Optimization of changes to minimize latency

4.3 Candidate 3: Radio D-Frame Encryption

In this use case, lightweight encryption would be applied at the radio transport layer to encrypt non-peer-to-peer messages. This could be applied between the ITCM components which control the transports (the CM and the ELM), or between the radios themselves. Given the greater resources and flexibility of the messaging system, it is likely encryption/decryption would be better performed within ITCM rather than the radios. The existing ITCM components already have functionality that can be enhanced to support this new functionality.

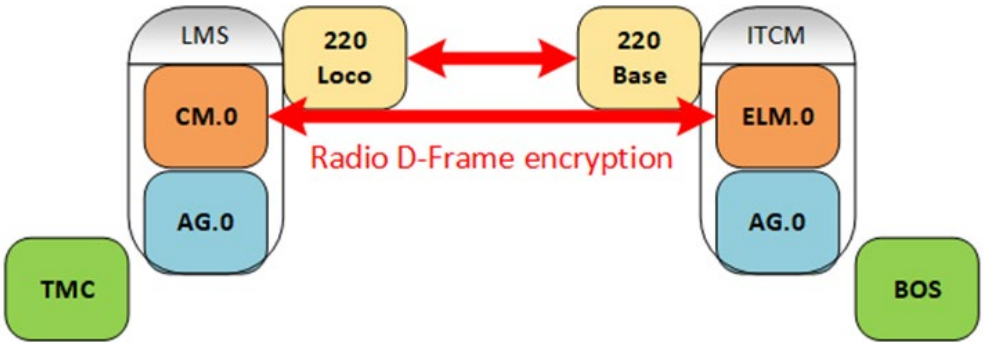


Figure 21. Radio D-frame Encryption

The application of encryption inside the communication infrastructure may mitigate the need for applications to perform encryption to conceal sensitive data. However due to the multiple transports involved in data transmission, this would have to be applied to other transports to

eliminate the other areas where messages could be exposed in transit. Use Case Overview and Benefits

Figure 21 is an overview and discussion of use-case benefits.

4.3.1 Requirements and High-Level Design

Protocol Requirements

To implement Radio D-Frame encryption, the following changes would be required:

1. The ITCnet protocol used between radios would need to be updated to support certificate exchange during connection setup.
2. The ELP protocol would require an update to support the additional messages exchanged for this functionality.
3. The connection manager would have to support the new ELP version.
4. The external link manager would have to support the new ELP version.

High-Level Design

The basic design of TNU security used in the Remote Back Office Exchange (RBX) protocol can be extended to add support for a key exchange required for encryption. RBX is a proprietary MCC protocol used between the CMs on the remotes and the IP-ELM in the BO.

Radio D-Frame encryption would be performed as follows:

1. The remote must be a secured ITCSM asset and support ITCM TNU security.
2. After an available connection is received from the radio, the highest-level ELP version is checked by each side. If the version supports encryption, the CM builds the same TNU registration as in RBX and send it to the ELM. If the version does not support encryption, the connection will be established using the existing, unencrypted ELP version.
3. The ELM will validate the registration message.
4. If the validation succeeds, the ELM will create a symmetrical encryption key, encrypt it using the public key from the registration message, add the key to a new ELP registration response message, and return it to the CM.
5. The ELM creates the transport queue for the remote and sends the TNU to the route publisher to broadcast to all MRs.
6. Each side then encrypts/decrypts the messages as needed using the shared encryption key. The NIST lightweight encryption algorithms could be used. The choice of key length and specific encryption algorithm is deferred to detailed design time.
7. Failures in the encryption processing on either side result in a system error being sent to the other party and the registration process being performed again. Multiple failures on the same connection result in the connection being terminated.

This approach does not provide mutual checks of identity; the BO validates the identity of the remote, but the remote does not validate the identity of the BO. If that identity check were needed, the public keys of all BOs would have to be distributed to the remotes and managed

appropriately for updates. The registration response would be updated to be signed with the BO private key and validated by the remote when it is received. This would not add significantly to the message overhead, but the management of the BO keys on the remote could be substantial because it is an interoperable transport.

Implementation and Migration Including Interoperability

The existence of an ELP negotiation strategy greatly simplifies the migration to the new encrypted connections because the migration simply depends on rolling out the updates. That rollout can be done in any order, though the BO is usually done first, then the system updates naturally. Management of the BO keys is critical because the system must reliably and automatically support syncing of the remotes and BOs or errors will make this transport unusable.

The encryption key has an advertised TTL in the registration response and the remote is expected to re-register within a reasonable period after TTL expires or face a system error. The only additional key management work is the distribution of BO keys if identity verification is required on the remote side. ITCSM provides a 10053 message to push updates of security artifacts to assets; this could be used to distribute those keys. Alternatively, in a pull strategy, the CM could request public keys from the BO as a part of initialization. Each BO already has a certificate store with the needed keys of the whole system.

The existing validation performed by the BO checks against a set of certificate revocation lists from all the federated roads so a compromised remote can be locked out if detected. Failures in the encryption processing could also trigger lock outs of remotes if desired.

4.3.2 PTC Network Impact

The impact to the 220-MHz network is two new messages over the air.

1. The registration message is a duplicate of the RBX message.
2. The registration response is new, but would still contain the encrypted session key, the TTL, the BO area ID, and an optional signature.

These messages do not have to be EMP messages, which would save significant overhead. The ELP header would likely be updated with additional metadata for encryption (e.g., an Enum indicating the encryption method, depending on the characteristics of the message).

The BO public key updates are infrequent and do not represent any impact to the network. CPU impacts depend on the final choice for the algorithm and key size. The latency added to the connection setup process could produce dead spots at inconvenient times. The impact to the size of the messages after encryption is only a few bytes per message.

4.3.3 Risks and Mitigations

Table 9 shows risks and mitigations.

Table 8. Candidate 3 Risks and Mitigations

Risk	Probability/Impact	Mitigations
TNU security not available on remote	Dependent on deployment timeline/high	None, TNU security is a pre-requisite.
Unintended Security failures in rollout	Medium/high	Extensive testing including trials/pilots. running the system in a non-enforcing mode until all issues are resolved
Unacceptable latency for setup process	Medium/high	Optimization of changes to minimize latency; fallback to previous ELP version during transition

4.4 Alignment with Roadmap

Changes to the PTC communication system necessitate consideration to the industry roadmap for ITC developed by AAR's TCCO and the railroads, as well as MCC's communications roadmap.

The core of the TCCO roadmap is moving towards QMB and ATO as the decade progresses, culminating in highly-autonomous operation. The goal of these major initiatives is to improve operational efficiency, thereby reducing costs. These initiatives are supported and supplemented by many other initiatives aimed at enabling QMB/ATO and/or improving efficiency. Examples include the Next Generation End of Train/Head-of-Train Communications Subsystem supporting Positive Train Location, Interoperable Energy Management Systems, and Next Generation Track Circuits to name a few.

Most initiatives on the TCCO roadmap impact the PTC communications system by increasing the load. Additional loads were modeled to forecast the impact on the PTC communication system as these initiatives progress. The analysis shows that for the busiest bases in Chicago, there is insufficient capacity on the extant 220-MHz band to meet all future needs. Therefore, it is critical that changes introduced to support new applications or to improve security are optimized to minimize the impact.

In addition to network loading, processor loading must also be considered. Many railroad products have a minimum life expectancy of 20 years, including PTC products. As a result, the current generations of wayside and on-board equipment are expected to be in service until at least 2030, and potentially much longer. Changes should introduce minimal additional processing load on aging and constrained platforms.

Some examples relating to network and processing capacity are as follows:

- MCC enhanced WSM filtering so the CM within ITCM can drop WSMs that are not of interest to the specific locomotive. This could significantly reduce the processing load on messaging as well as processing and network loads on the OBC.
- Researchers have previously optimized the ITCM software to maximize throughput on remotes, and is undertaking a more significant optimization exercise. This is expected to dramatically reduce processor loading for a given throughput.
- Development is underway on the next generation 220-MHz PTC radio, including several features to address processing capability and network loads. Examples include:
    - Utilizing a powerful multicore System-On-a-Chip for processing
    - Supporting full-rate transmissions from wayside radios, which will free up significant network capacity compared to the existing half-rate radios

- Supporting adaptive coding and modulation so higher bit rates can be achieved, where conditions permit
- Supporting adaptive power control to minimize interference
- Supporting wider bandwidth channels where beneficial

Improving confidentiality through encryption will inevitably result in some penalty to processing load and network utilization. The goal has been to minimize the penalty and compensate for it by choosing efficient algorithms (i.e., NIST and ELLI) and to recommend compensating measures. This best supports the TCCO roadmap, including new applications requiring additional network capacity. Specifically, the authors recommend:

1. Completing the work necessary to fully deploy TNU security, which is a prerequisite for Radio D-Frame encryption, if adopted
2. Considering lightweight cryptographic algorithms backed by open standards wherever possible for future applications
3. Completing the work necessary to deploy WSM filtering
4. Adopting optimized versions of ITCM software as they are released
5. Adopting the next generation 220-MHz PTC radio and the features above as soon as practical for dense urban areas, where the 220 MHz links and current generation radios are heavily loaded
6. Maximizing use of frequency bands other than 220 MHz (including for ITCM), where possible

5. Final Summary and Recommendations

This section presents the project summary and recommendations.

5.1 Phases 1 to 3

The recommendations in this report are based upon work performed in Phases 1–3 of the project, including TAG input.

During Phase 1, the MCC team surveyed potential solutions from NIST and several different industry sectors. They also identified requirements, constraints, and best practices relevant to PTC communications and any candidate solutions. The AAR S-9010 Data Protection Standard [1] (released in 2019) was a key input, outlining how railroad data should be classified. It also specified privacy, integrity, and authentication protection mechanisms and requirements to apply them based on the classifications.

The research team performed an analysis of key interfaces with input from S-9010 and the TAG. They concluded that most messages exchanged over the PTC communications network meet the protection requirements of S9010, and exceptions were being addressed.

The TAG discussed the possibility of encrypting WSMs, as WSMs report the states of signals and switches in an unencrypted form. However, given that the states can easily be observed and considering the possibility that data from multiple WIUs could be aggregated, the TAG concluded that WSMs meet the requirements of S-9010 and do not need to be encrypted.

Phase 2 continued the evaluation of candidate solutions, focusing on implementation, migration, and deployment challenges. Guidance from the Phase 2 report has been considered in this report and will also prove valuable if railroads decide to implement additional measures to improve confidentiality.

A key recommendation during Phase 2 was that the project should consider widely-available algorithms with recognized open standards that have validated libraries. This resulted in a project decision to focus on the NIST “Final 10” Lightweight Encryption Algorithms and ELLI (covered by ISO/IEC 29192-4, Amendment 1 [2]). ANDRO was formally engaged during Phase 2 to help MCC with the assessment of these algorithms.

Phase 3 identified knowledge gaps and areas for further research outside of the project.

5.2 Common Themes

The following were common themes throughout the TAG meetings:

1. S-9010 requires protection mechanisms to be implemented at the application layer, and having that application at lower layers (e.g., the transport layer) does not alleviate this requirement.
2. Location information, combined with commodity information, or even the shipper’s name, requires confidentiality.
3. The ITC committee has not identified requirements for confidentiality other than crew login.

4. Confidentiality without identity assurance is not valuable. Information must be sent to a trusted identity from a trusted identity, otherwise the content cannot be trusted, even if encrypted.

5.3 Recommendations Based on the NIST Lightweight Cryptography Project

The NIST lightweight Cryptography project [13] identified 10 finalist AEAD algorithms, detailed in RFC 5116 [13]. ANDRO successfully exercised the NIST lightweight encryption algorithms and selected four recommended algorithms, based on potential PTC communications use cases and expected performance on constrained platforms.

The recommended algorithms are Ascon, Spark, TinyJAMBU, and Xoodyak. All four exhibited similar performance and can be executed using the same API. NIST is expected to complete its final round of the standardization process in 2022. The research team believes that at least one of these algorithms will likely be documented in a future NIST standard.

The ELLI algorithm for asymmetric encryption was also evaluated. Although a more thorough investigation is needed, the authors believe that ELLI has several potential applications for PTC communications, including improving security for OPKs.

5.4 Recommended Applications of Lightweight Encryption

The project has identified three areas in which lightweight encryption can improve confidentiality for PTC communications. In order of preference, these are as follows:

1. **Application Layer Encryption (Non-WSM):** Lightweight encryption is applied at the application layer between the remotes and the BO.
    a. This would allow additional application-level messages to be encrypted, if desired or required. The use of NIST algorithms would minimize the additional processing load on performance-constrained remotes, compared to more traditional algorithms.
    b. This would also satisfy the S-9010 standard for application layer encryption(although S-9010 would need to be updated to accommodate new algorithms.
    c. The EMP message structure already lends itself to the addition of new encryption methods through the Message Type and Message Version fields. This also makes migration straightforward because the potential required encryption can be determined from these fields.
    d. There is also potential to improve the security of private key material by replacing symmetric OPKs with asymmetric cryptography. Using ELLI for this could also minimize loading on remotes. During migration, applications would fall back to the current pre-shared locomotive OPKs as necessary. This would need to be managed by an Interoperable KES.
    e. Adoption of application layer encryption (as described above) would result in additional computational and network overhead during locomotive initialization and when key renewal is conducted. The additional loading should not be a significant issue.

2. **Application Layer Encryption for Wayside Status:** Lightweight encryption is applied to peer-to-peer messages involved with the exchange of WSMs between waysides and nearby locomotives.
    a. As discussed above, the consensus of the TAG is that encryption is not currently required for WSMs.
    b. A hybrid approach could be used, in which a lightweight asymmetric algorithm (e.g., ELLI) would be used to securely generate and distribute symmetric wayside OPKs compatible with the existing HMAC algorithm.
    c. If the position changes regarding the need for encryption for WSMs, the hybrid approach could be extended so that symmetric encryption using NIST algorithms is applied to WSM sensitive fields, improving confidentiality. Other fields (e.g., WIU IDs) would remain in the clear (for example, to support message filtering). WSM encryption is not recommended because it will likely break compatibility and interoperability, and result in significant additional loading for the TMC, even if NIST lightweight encryption algorithms were adopted.
    d. A hybrid encryption approach that excludes encryption of WSM status fields would provide a straight-forward migration path. This would require changes to the KES, subdiv files, WIUs, and TMCs. The KES would support interoperability, managing fall back to the existing OPK scheme for assets that are not yet upgraded.
    e. The use of ELLI would minimize the associated processing load where traditional algorithms (e.g., RSA) would be impractical. The processing load would likely be higher than that required to process HMACs, so optimization may be required as above.
    f. Based on loading of on-board systems observed in DUAs, optimizations will be required before any significant additional loading can be introduced (e.g., the adoption of WSM filtering and optimized remote ITCM software).
    g. Additional loading of wayside systems is unlikely to be an issue, but this needs to be verified.
3. **Radio D-Frame Encryption:** Lightweight encryption is applied at the radio transport layer to encrypt non-peer-to-peer messages.
    a. Symmetric encryption using NIST algorithms could be implemented to improve confidentiality of all Dynamic Time Division Multiple Access traffic transferred in the D-Frame of ITCnet.
    b. To avoid increasing loading on first generation PTC radios, this would be applied between the ITCM components interfacing to the radios, rather than by the radios themselves.
    c. This would require changes to the ITCnet protocol used between radios to support certificate exchange during connection setup, as well as changes to the ELP protocol.
    d. As with application-layer encryption for wayside status, optimizations will be required before any significant additional loading of on-board systems can be introduced.

6. Conclusion

Based on the work MCC and ANDRO performed during the project and the TAG input received, this project identified several areas where confidentiality could be improved for PTC communications. Based on TAG input and S-9010, there are no current requirements to do so, therefore the options that follow are provided for consideration if requirements were to change.

Applying symmetric encryption using NIST lightweight algorithms to non-WSM application messages would improve confidentiality while minimizing the additional load on end points. Existing protocol features enable migration in a straightforward manner. The Ascon, Spark, TinyJAMBU, and Xoodyak algorithms are recommended for consideration, although this list may be revised based on NIST standardization efforts. These algorithms could be used to replace existing algorithms or reserved for messages not currently encrypted.

There is potential to improve the security of private key material by using a hybrid approach to distribute OPKs. This would use lightweight asymmetric cryptography such as ELLI, possibly together with NIST algorithms, for lightweight symmetric encryption. During migration, applications would fall back to the current pre-shared locomotive OPKs as necessary. This would need to be managed by an Interoperable KES. The value of migration would have to be assessed against the risks associated with widely shared keys and the associated mitigations.

NIST algorithms could also be used to encrypt sensitive fields within WSM payloads at the application layer if this became a requirement. Encryption would likely only be practical if coupled with the adoption of WSM filtering and/or optimized remote ITCM software to counterbalance the added processing load.

Encryption of the Radio D-Frame using NIST algorithms would be equally problematic for loading. This would likely require encryption/decryption to be performed within ITCM, rather than the radios. This is because there is a greater possibility of counterbalancing the added load through optimizations (as above) when compared to the radio.

References

[1] Association of American Railroads, "S-9010 Data Protection Standard, V1.0," 2019.

[2] International Organization for Standardization, "ISO/IEC 29192-4," vol. 1, amendment 1, June 1, 2016.

[3] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. New York, NY, USA: Wiley & Sons, 1996.

[4] National Institute of Standards and Technology, "NIST Lightweight Cryptography Finalists," (accessed Nov. 2, 2021.)

[5] V. C. Hu, "NISTIR 8360 Machine Learning for Access Control and Policy Verification," National Institute of Standards and Technology, 2021.

[6] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "NIST Special Publication 800-207 Zero Trust Architecture," National Institute of Standards and Technology, 2020.

[7] J. Kindervag, "John Kindervag: 'The Hallmark of Zero Trust Is Simplicity'," *The Wall Street Journal*, April 15, 2021.

[8] National Security Agency, "Embracing a Zero Trust Model," 2021.

[9] National Institute of Standards and Technology, "Special Publication 800-63," 2017.

[10] The Open Worldwide Application Security Project. "M9: Improper Session Handling."

[11] ANDRO Computational Solutions, "Lightweight Encryption Implementation and Testing Option Final Report, Rev 1.0," 2022.

[12] K. McKay, L. Bassham, M. Turan, and N. Mouha, "NIST IR 8114: Report on Lightweight Cryptography," 2017.

[13] D. McGrew. "RFC 5116 - An Interface and Algorithms for Authenticated Encryption," *Internet Engineering Taskforce* (accessed Feb. 9, 2022.)

[14] National Institute of Standards and Technology, "Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process," 2018.

[15] M. Kerrisk. "perf_event_open(2) — Linux manual page," *man7.org* (accessed Oct. 10, 2021).

[16] Ç. Çalık, M. Hasan, and J. Kang, "Benchmarking Round 2 Candidates," presented at the NIST Lightweight Cryptography Workshop 2020, Virtual, Oct. 19-21, 2020.

[17] M. S. Turan, et al., "NISTIR 8369: Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process," 2021.

[18] M. Hughes. "How Elliptic Curve Cryptography Works," June 26, 2019.

[19] D. McCreary. "Comparing RSA and ECC Encryption," *NetBurner*, Mar. 23, 2020.

[20] B. Buchanan. "Public-Key Crypto and RFID Tags," *Medium*, Mar. 28, 2020.

[21] Association of American Railroads, "S-9354, Edge Message Protocol Specification," 2012.

[22] National Institute of Standards and Technology. "NIST Lightweight Cryptography Project," (accessed Feb. 9, 2022).

Abbreviations and Acronyms

ACRONYMS	EXPLANATION
AAR	Association of American Railroads
AEAD	Authenticated Encryption with Associated Data
AMQP	Advanced Message Queuing Protocol
ATO	Automated Train Operations
BO	Back Office
CM	Connection Manager
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSMA	Carrier Sense Multiple Access
CT	Cipher Text
DUA	Dense Urban Areas
ECC	Elliptic Curve Cryptography
ELLI	Elliptic Light
ELM	External Link Manager
ELP	External Link Protocol
EMP	Edge Message Protocol
FPE	Format Preserving Encryption
HMAC	Hash-based Message Authentication Code
IAL	Identity Assurance Levels
I-ETMS	Interoperable Electronic Train Management System
IoT	Internet of Things
IP	Internet Protocol
IP-ELM	IP External Link Manager
ITC	Interoperable Train Control
ITCM	Interoperable Train Control Messaging
ITCSM	Interoperable Train Control Systems Management
KES	Key Exchange Service
MCC	Meteorcomm, LLC
MDM	Mobile Device Management
MR	Message Router

MRG	Messaging Real-time Grid
NIST	National Institute of Standards and Technology
OBC	On-Board Computer
OPK	Operational Private Key
POC	Proof of Concept
PT	Plain Text
PTC	Positive Train Control
QMB	Quasi-Moving Block
RBX	Remote Back Office Exchange
RF	Radio Frequency
SCEP	Simple Certificate Enrollment Protocol
SMA	Systems Management Agent
SMG	Systems Management Gateway
SSL	Secure Socket Layer
TAG	Technical Advisory Group
TCCO	Train Control, Communications, and Operations
TMC	Train Management Computer
TTL	Time to Live
VPN	Virtual Private Network
WIU	Wayside Interface Unit
WSM	Wayside Status Message
WSRS	Wayside Status Repeater Service