SANDIA REPORT
SAND2022-11359
Printed August 2022

Sandia National Laboratories

Security Evaluation of Smart Cards and Secure Tokens: Benefits and Drawbacks for Reducing Supply Chain Risks of Nuclear Power Plants

Benjamin Karch, Michael Rowland

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

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

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

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

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

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

ABSTRACT

The supply chain attack pathway is being increasingly used by adversaries to bypass security controls and gain unauthorized access to sensitive networks and equipment (e.g., Critical Digital Assets). Cyber-attacks targeting supply chain generally aim to compromise the environments, products, or services of vendors and suppliers to inject, add, or substitute authentic software and hardware with malicious elements. These malicious elements are deemed to be authentic as they arise from the vendor or supplier (i.e., the supply chain). This research aims to leverage findings and assumptions made from the previous report to determine the security benefits and drawbacks of a smart card-based hardware root of trust.

Smart cards can provide devices inside Nuclear Power Plants (NPP) with a secure environment to store keys in and perform sensitive operations such as digital signature generation. These abilities can be leveraged to increase supply chain cybersecurity by autonomously providing NPP Licensees with reports on device integrity, authenticity and measurements of executable and non-executable data.

ACKNOWLEDGEMENTS

The authors acknowledge Dr. Christopher Lamb (SNL) and Dr. Shannon Eggers (INL) for their review and comments on this report.

This research was funded by the U.S. Department of Energy Office of Nuclear Energy Cybersecurity research and development program under milestone M3CT-22SN1105013.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND TERMS

Acronym/Term	Definition
APDU	Application Protocol Data Unit
CDA	Critical Digital Asset
CMV	Configuration Manager and Verifier
ECDSA	Elliptic Curve Digital Signature Algorithm
ENISA	European Union Agency for Cybersecurity
FAT	Factory Acceptance Testing
FPGA	Field Programmable Gate Array
HROT	Hardware Root of Trust
I&C	Instrumentation and Control
IC	Integrated Circuit
ICT	Information and Communications Technology
IP	Intellectual Property
IT	Information Technology
JCOP	Java Card Open Platform
NPP	Nuclear Power Plant
OEM	Original Equipment Manufacturer
OT	Operational Technology
PCI	Peripheral Component Interconnect
PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
ROM	Read Only Memory
RSA	Rivest Shamir Adleman
SCAS	Supply Chain Attack Surface
SIEM	Security Information and Event Management
SOC	Security Operations Center
TBB	Trusted Boot Block

This page left blank

1. INTRODUCTION

A European Union Agency for Cybersecurity (ENISA) report titled "ENISA Threat Landscape for Supply Chain attacks" noted that "supply chain attacks increased in number and sophistication in the year 2020 and this trend is continuing in 2021, posing an increasing risk for organizations. It is estimated that there will be four times more supply chain attacks in 2021 than in 2020." [1].

The ENISA report lists several publicly disclosed supply chain attacks for which hardware roots of trust are expected to provide varied levels of protection. For example, consider counterfeit of a Hardware Wallet for cryptocurrency. Attackers have demonstrated a capability to provide consumers with counterfeit hardware (USB) based cryptocurrency wallets. Upon insertion of the user's wallet into a computer by the user, the private keys are then exfiltrated back to the attackers. A hardware root of trust (HROT) would be able to prevent this attack because the counterfeit devices would not have an embedded root of trust capable of providing signed and verifiable information by a trusted public/private key pair.

However, the level of protection for a SolarWinds Orion type attack is not direct. For example, a HROT would not defend against attackers that were able to provide software updates that included malware and originated from and were digitally signed by the software vendor themselves. Some protection may be provided by the HROT to securely report device configuration and state information. This is limited by whether an attacker can sign the changes with a trusted vendor's private key. In this case, a malicious backdoor could be trusted and allowed to change the configuration or state of the device, with the operator unaware of these malicious changes. It should be noted, though that if the attackers leverage the malicious back door to change other executable code or configuration data (such as logic data) to no longer match the signed/trusted/expected version, the operator can detect this second stage of the attack.

This research seeks to evaluate the protections provided by tamper-resistant smart cards and determine whether they are a suitable candidate to provide a HROT within a digital instrumentation and control (I&C) device or platform operated or relied upon by Nuclear Power Plant (NPP) licensees (i.e., critical digital asset (CDA)). This effort leverages the previous report, "A Review of Technologies that can Provide a 'Root of Trust' for Operational Technologies" [2], combined with a reporting mechanism to uncover malicious changes made to the device after the design phase. [2] finds that smart cards are particularly well-posed to act as an HROT that provides supply chain protections, as they can be bound to the device at an early stage in the supply chain. Solutions such as the Trusted Platform Module (TPM) [3] rely on personalization of the device to a user. Digital I&C generally operate without a user present, which is a core assumption / requirement of the TPM, and the process of taking ownership cannot occur until the I&C device is present at the NPP.

The evaluation and analysis of smart cards detailed within this report focuses on a single root of trust implementation (i.e., secure element to store cryptographic secrets and sensitive information) of CDAs in an NPP. This implementation is a prototype meant to provide insights into the applicability of smart cards as a HROT in the context of NPPs, allow for performance analysis, and analysis of preliminary attacks. This paper is not intended to serve as a blueprint for a final implementation or as an absolute catalog of smart card features and capabilities. A "root of trust" is a tamper-resistant element in a digital system that has a high level of protections, making it impossible or extremely unlikely for its secrets to be extracted. This justifies the assumption that secure elements/smart cards can be depended on as the root of all trusted operations. The core protection of trust provided by the root of trust is the secure (i.e., tamper resistant) storage of a private key (i.e., a secret that is not shared) which can be used to manage other generated keys and

sign data to be sent out of the secure element. The root of trust protection involves asymmetric cryptographic mechanisms that involves a public key which is shared widely and openly and a private key generated on the root of trust and never leaves the root of trust. Since the private key is only available to the root of trust, any communications can be signed using this key to provide non-repudiation and significantly strengthen confidence in authentication. Asymmetric cryptographic mechanisms can provide for protection of confidentiality (encrypt via public key; decrypt via private key) as well as provide integrity (via digital signature standards/algorithms) as the root of trust is the only component that has access to the unique private key. Secure and trusted communications between devices using public key cryptography requires the implementation of a Public Key Infrastructure (PKI). A PKI is used to exchange information about the authenticity and validity of keys, relying on a Certificate Authority (CA) which ensures all members of the system have access to updated and trusted information regarding public keys and their associated entities' validity. The development of a PKI is not within the scope of this project, and it is assumed that information held at the CMV is trustworthy and not subject to attack / tampering.

The three major benefits of smart cards are:

(i) Plug and play functionality.

(ii) Reduce operator workload.

(iii) Device authenticity tools/technology.

Plug and play functionality is established by generating the private/public key on the card at the smart card manufacturer providing the Original Equipment Manufacturer (OEM)/Integrator with a smart card integrated circuit (IC) loaded with a private and public key. The improvements in the smart card technical performance over the last few years have allowed for these key pairs to be generated by the card. This substantively increases security as the private key never leaves the secure element (i.e., card). Additionally, this reduces the effort to 'personalize' or preload the data with platform information, thus enhancing its plug and play functionality.

In this way, the private key is held only by the HROT, and unavailable / unobtainable by any other entity. The smart card app then implements a function where you can request its public key (see appendix for the Application Programming Data Unit (APDU)). The smart card manufacturer can then sign this public key as a certificate and provide it to the OEM along with the smart card. When the OEM installs the smart card into a particular device (this is the hardware integration phase) the OEM creates a mapping of the device to the public key (the public key is verified with the smart card manufacturer's certificate authority). This mapping can then be signed by the OEM increasing security and trustworthiness. A trusted smart card will have a verified public key by the OEM's certificate authority. These certificates are then verified at the NPP location/endpoint; this establishes a one-to-one mapping with the system and the smart card via the public/private key. There is no need for further configuration at the endpoint, everything just needs to be verified, and that can be done autonomously by the CMV. Note that the current implementation assumes a trusted list of public keys on the CMV.

Reduction of NPP staff workload is accomplished by not requiring the operator to conduct an in-depth analysis for hardware and software validation. The device's report of its firmware is trusted because it is signed by the private key of the root of trust (see Appendix A). This provides evidence (i.e., cryptographic proof) that the device is giving an authentic report as to the device's state. This reduces the effort on the operator to only verify that there are no signs of hardware manipulation on

the device. This reporting mechanism can be leveraged for by subsequent tools and interfaced technologies (e.g., Cyber Security Operations Center (SOC)).

Additionally, leveraging the one-to-one mapping of devices and smart cards, the most credible attack that could bypass the protections is one where a legitimate device has had its smart card unfused and removed from the legitimate device and placed into a counterfeit. If the smart card is securely embedded into the hardware (i.e., placed under a layer of silicone on the board), the potential for an attacker to be able to successfully remove it without destroying it decreases, and if they do, the product received should exhibit clear evidence of tampering such as soldering that does not appear to have been done by the manufacturer. This attack would be limited to adversaries with significant resources, capabilities, and time, significantly lowering the risk due to eliminating many potential adversaries. Additionally, at a worst-case scenario there can only be one counterfeit per genuine device, the genuine device is destroyed in the process.

The HROT tools/technology have plug and play functionality and the configuration reporting mechanism, the NPP staff can apply these tools to verify the authenticity of the device, at boot time and other discretionary high impact operations. These operations can trigger the secure element/root of trust to provide information about authenticity of the logic files that are received and executed. The only limiting factor is the length of time for the digital signature to be generated. The trust anchor could also provide secure reporting back to Cyber SOC or Security Information and Event Management (SIEM) system or to provide identity management in a zero-trust network architecture

The purpose of this research effort was to verify that smart cards will improve supply chain cybersecurity for NPPs. A long-term goal is to provide off the shelf security and security-enabling features from the OEM to Integrator to NPP Licensee. This report details the evaluation of the smart cards and supports the assumptions with respect to the level of protection provided against supply chain attacks by roots of trust. However, roots of trust are not a perfect solution. This report also identifies challenges that require further research to establish that the smart card can provide major increases to security for a large portion of the supply chain.

This page left blank

2. BACKGROUND

The recent advances in the performance and capabilities of these HROT technologies in recent years has increased their potential applications to reduce or mitigate exposure of the supply chain attack pathway. The focus of the initial report [2] was to being on providing an analysis of the benefits and disadvantages of smart cards, secure tokens, and elements to provide root of trust. This report follows the initial report [2] and aims to provide evidence that these roots of trust can increase the technical capability of equipment and networks to authenticate changes to software and configuration thereby increasing resilience to some supply chain attacks, such as those related to logistics and Information and Communication Technology (ICT) channels, but not development environment attacks.

The supply chain is the network by which a product or service moves from supplier(s) to the acquirer. Figure 1 below illustrates these relationships for an NPP supply chain.

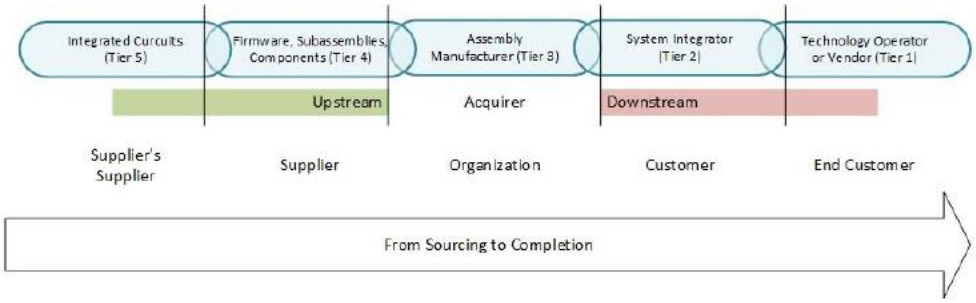

Figure 1 Supply Chain Relationships [4]

Relevant entities establish supply chain relationships with vendors, contractors, and suppliers for a variety of reasons such as focusing resources on core functions, acquiring capabilities that the relevant entity needs but does not possess, acquiring a utility or basic service that is commonly available, enabling work from remote locations and acquiring new or replacement systems which perform functions related to nuclear safety or security [4].

2.1. Supply Chain Attack Surface

The nuclear I&C supply chain attack surface defined in [5] and depicted in Figure 2, defines stages and attacks within the supply chain.

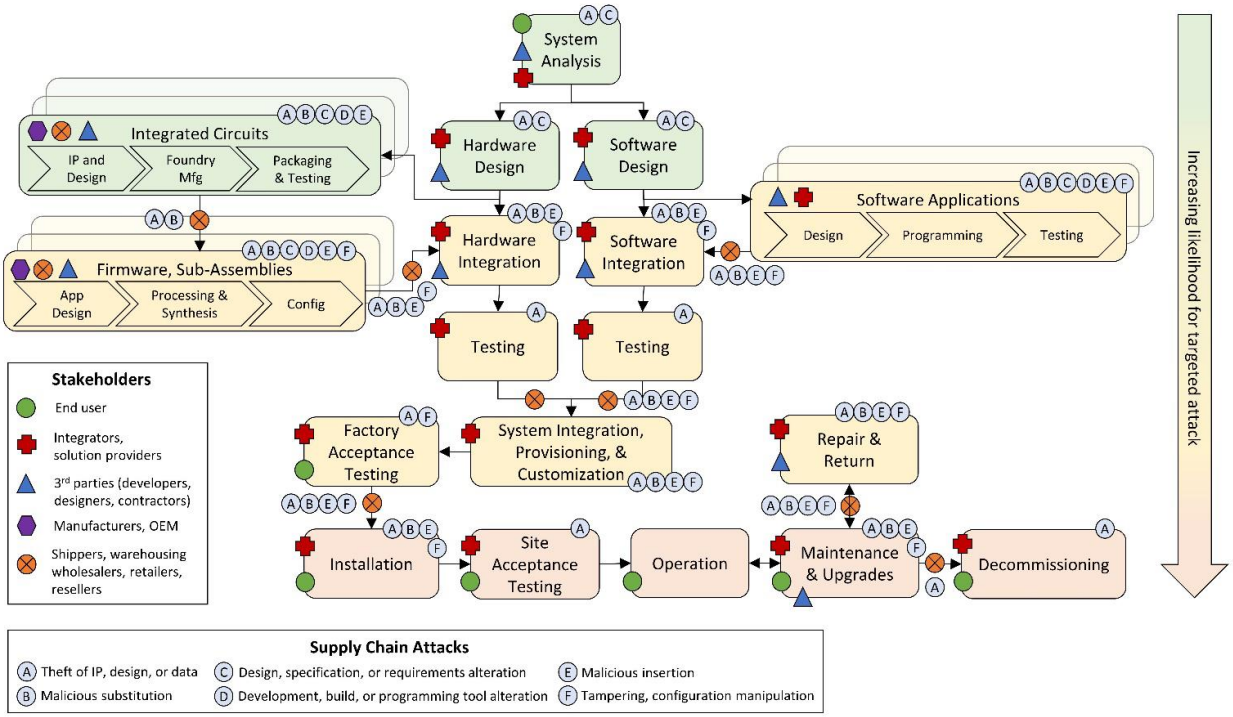

Figure 2 Supply Chain Attack Surface [5]

The Supply Chain Attack Surface (SCAS) figure above provides a dashboard from which to evaluate the types of protections (i.e., what types of attacks are reduced or mitigated) and the level of protection (i.e., degree to which the attack is reduced or mitigated). The Supply Chain Attack Surface and attack types are key to evaluating the protections provided by roots of trusts/secure elements.

This report evaluates a potential implementation and how protection may be targeted for protection by the smart card from supply chain attacks through the system lifecycle. An example supply chain attack is a malicious insertion, in which a part of the devices code or a physical component of the device is replaced with a malicious version, or a malicious hardware or software component is added to the device. Configuration reports are signed and sent out from the smart card should be able to capture malicious insertions that may happen at any point after the device OEM develops and signs the firmware. After these reports, measurements taken by the device can be compared with the known good firmware to validate authenticity and integrity.

Leveraging the SCAS, the roots of trust must be initialized by the smart card manufacturer and integrated/installed within the Operational Technology (OT) system or environment, likely by the OEM or integrator. One type of protection such as authentication of hardware components can provide key insight into the supply chain security of a device. [6] depicts the supply chain for NPP devices in Figure 3.

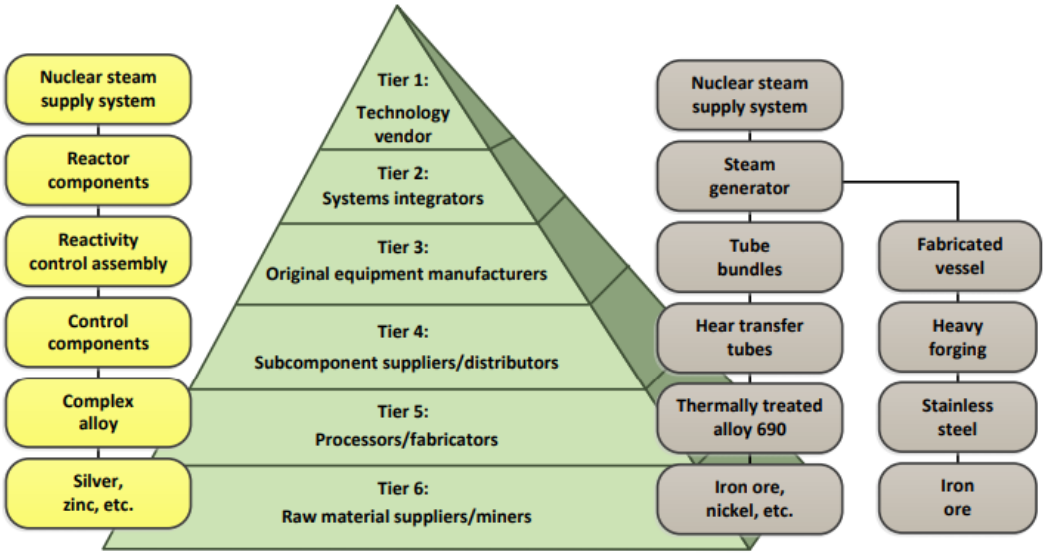

Figure 3 Supply Chain for NPP Devices [6]

Figure 3, which depicts the supply chain for NPP devices, provides a key installation/establishment point for root of trust. This report postulates that root of trust implementation to protect the Nuclear Supply Chain is most likely to occur at the System Integrators (Tier 2) and OEM level (Tier 3). Wider protections that extend beyond the Nuclear Supply Chain may further extend root of trust protection to its subcomponents are Tier 4. Some microprocessors provide for a firmware root of trust, but these do not offer many of the tamper resistant features of smart cards and other security elements. The protection value of the HROT is expected to be based upon the tier of implementation. For example, the lower the tier of HROT implementation, the expectation is that the probability of a successful attack will decrease and the confidence in the received device and its subcomponents is increased.

Following from this analysis, this report relies upon the fundamental assumption that protections provided by the proposed HROT are only applicable after the design phase. Protections are offered only after design because there must be a point at which the expected state of the Programmable Logic Controller (PLC) is reported by the manufacturer to be later verified during boot and reporting back to the CMV. In other words, the operator can detect devices which do not meet the specifications provided by the supplier but detecting a flawed product or malicious design is not likely or feasible by the device itself or its end user. These key considerations are investigated in this report.

This page left blank

3. RESEARCH METHODOLOGY

The previous report [2] established a theoretical level of confidence in attack protections against various attacks. These findings, summarized in Table 1, are used to develop test scenarios to verify that the protections can be implemented using a smart card based HROT. The test results and analysis provide evidence supporting the hypothesis that an integrated HROT significantly lowers the probability (or makes impossible) certain types of attack that may occur in the supply chain for a digital I&C device.

Table 1 Summary of Protection (Adapted from [5])

Attack Type	Description	Root of Trust Protection Confidence	Protection Method
Theft of IP, design, or data	Unauthorized disclosure of information from a stakeholder who has a trust relationship with the end target, enabling future attacks and/or causing economic loss. This may include but is not limited to intellectual property (IP), design information, operational / configuration data, or stored secrets (i.e., private key, digital certificates).	Low	Full disk encryption using stored keys on the secure element will help to prevent unauthorized access to device information. A supplier may implement access control and auditing to prevent unauthorized access while a device is in development / manufacturing stages. A secure element will not prevent an authorized user from maliciously leveraging their authorized access.
Malicious substitution	Complete replacement of digital technology, including hardware, firmware, and/or software. Hardware clones or counterfeits may not impact all end users depending on the distribution, whereas a substituted software package may compromise all end users even if only a few were targeted.	High	A root of trust will be able to provide the operator with real time trusted information on the devices hardware, firmware, and software, making unauthorized substitutions detectable. A complete hardware swap will not report correctly or at all without the secure element, as any report will not be able to be signed with a key which may be confirmed through a certificate issued by the manufacturer.

Attack Type	Description	Root of Trust Protection Confidence	Protection Method
Design, specification, or requirements alteration	Unauthorized modification of design, specifications, or requirements that compromises the design stages and results in the purposeful inclusion of latent design deficiencies (e.g., requirements that result in vulnerabilities) or built-in backdoors.	Low	Like theft of IP, design or data, a supplier may implement an access control or auditing system that utilizes root of trust technologies, but an authorized change made during design stages will lead to a trusted but insecure state at the customer.
Development, build, or programming tool alteration	Unauthorized modification of the development environment, including platform, build and programming tools, with the intent to corrupt the device under development.	Medium	Code should be signed by the supplier, and then verified on the end device. Alterations that attempt to alter code after it has been signed will be detected. Though, a supplier with bad security practices may improperly implement signing procedures and sign code after it has been manipulated. Additionally, any tools / programs (e.g., a compiler) used to assist in development can also be signed and verified by the supplier.
Malicious insertion	Addition or modification of information, code, or functionality directly into a device to cause malicious intent, such as impairing or altering device operation or function.	Medium to High	Similar to malicious substitutions, a malicious insertion during the logistics and ICT transfer to the customer or during operation would be detected by system. Authorized and approved malicious insertions directly from the developer or OEM would be part of the approved component.

Attack Type	Description	Root of Trust Protection Confidence	Protection Method
Tampering, configuration manipulation	Unauthorized alteration or fabrication of configuration, non-executable data, or sending of unauthorized commands with the goal of impacting device operation or function.	Medium to High	Changes to non-executable data that is measured and reported will be detected, though, it is likely not feasible to measure all data on the device. Properly defining which files and data is important to verify will allow for this information to be measured and reported. A root of trust also allows for the device to send commands securely (encrypted and authenticated) to other devices in the system.

3.1. Environment

A representative environment was established using single-board computers (i.e., Raspberry Pi 4) that are often used in IoT implementations, coupled with a commercially available and programmable smart card. This environment, depicted in Figure 4 and detailed in Appendix A, represents a PLC, a HROT, and a Configuration Manager and Verifier (CMV) which are programmed to interact with each other similarly to how a smart phone, its Subscriber Identity Module (SIM), and a cellular tower confirm a smart phones identity and configuration.

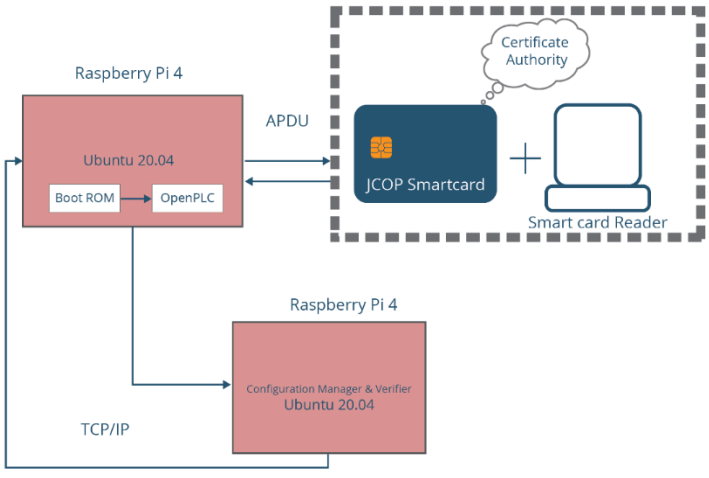

Figure 4 Environment Architecture

The environment consists of the following major components:

1. The "PLC" – a Raspberry Pi 4 with an Ubuntu 20.04 Long Time Support (LTS) Advanced Reduced Instruction Set Computer Machine (ARM) version running a custom Boot ROM and OpenPLC[1];
2. Configuration Manager and Verifier – a second Raspberry Pi 4 with an Ubuntu 20.04 LTS ARM version; and
3. The HROT – Implemented using a Java Card Open Platform (JCOP) smart card.

The HROT is integrated with the PLC via a USB card reader. OpenPLC is implemented as a service that begins on the startup of the PLC. The startup script for OpenPLC serves the role of a Boot Read Only Memory (ROM) for the PLC, as it is the first executed user space file for commencing PLC related functions. The CMV receives reports from the PLC and makes determinations about the PLC's state. The CMV (i) verifies the signature and message data, (ii) maintains the records of state information and allowed configurations, and (iii) tracks reported nonces. There are two possible outcomes from the CMV:

1. The CMV receives information from the PLC that indicates the PLC is outside of its expected state or that it has been altered or replaced; it detects this variation and raises an alarm condition. This is an abnormal condition but may be the result of an authorized change to the PLC, such as maintenance personnel changing PLC firmware to a non-malicious version that has not been authenticated yet.
2. The CMV receives information from the PLC that indicates the PLC is within its specification, this information could be later repeated back to the PLC (if it is determined to be misconfigured) so that it can revert to the state requested by the CMV. This would be the normal/expected condition. Note that autonomous reconfiguration is not implemented, but a potential subject of future work.

These binary outcomes are important as it demonstrates how a device (in this case a PLC) may be protected by the HROT, and it also provides a third party a means by which to verify the information from that device. Additionally, the means for a third party to verify the information can be further evaluated to determine the potential for this party to detect attacks. The CMV is a custom program that provides a determination as to whether the device is a trusted implementation.

The CMV serves essentially the same role as the Policy Engine within a Zero Trust Architecture network, defined in NIST Special Publication 800-207 [7]. In other words, the CMV can receive authentication information from each device before they are allowed to interact with other devices or can revoke devices' access if an attack or misconfiguration is detected. The use of open-source software allows for these programs to be edited to integrate the HROT during various operations. Additionally, the software can be edited and recompiled to simulate an attack on the device's firmware.

[1] an open-source software-based product implemented as a service on Linux.

3.1.1. Startup Execution Flow

The execution flow for Boot ROM for the PLC (depicted in Figure 5) is modeled after a measured boot, where the measurements are signed by the smart card. The boot process[2] involves the following:

1. The boot process for OpenPLC is initiated
2. 'startup.sh' is loaded and executed
3. The PLC firmware (webserver.py) is measured using SHA-1
4. A pseudorandom nonce is generated
5. The nonce and hash are passed to the HROT to be signed
6. The nonce, hash, and signature are passed to the CMV
7. A list of hardware components on the PLC is taken using standard Linux libraries
8. Steps 4-6 are repeated with the hardware measurement

The 'start_openplc.sh' acts as the Trusted Boot Block (TBB), which is used in the trusted boot procedure defined by the Trusted Computing Group [8]. The TBB is not measured by any external code and is the first execution of the trusted boot chain. A core assumption of this trusted boot chain is that the TBB cannot be altered or tampered with without destroying the PLC.

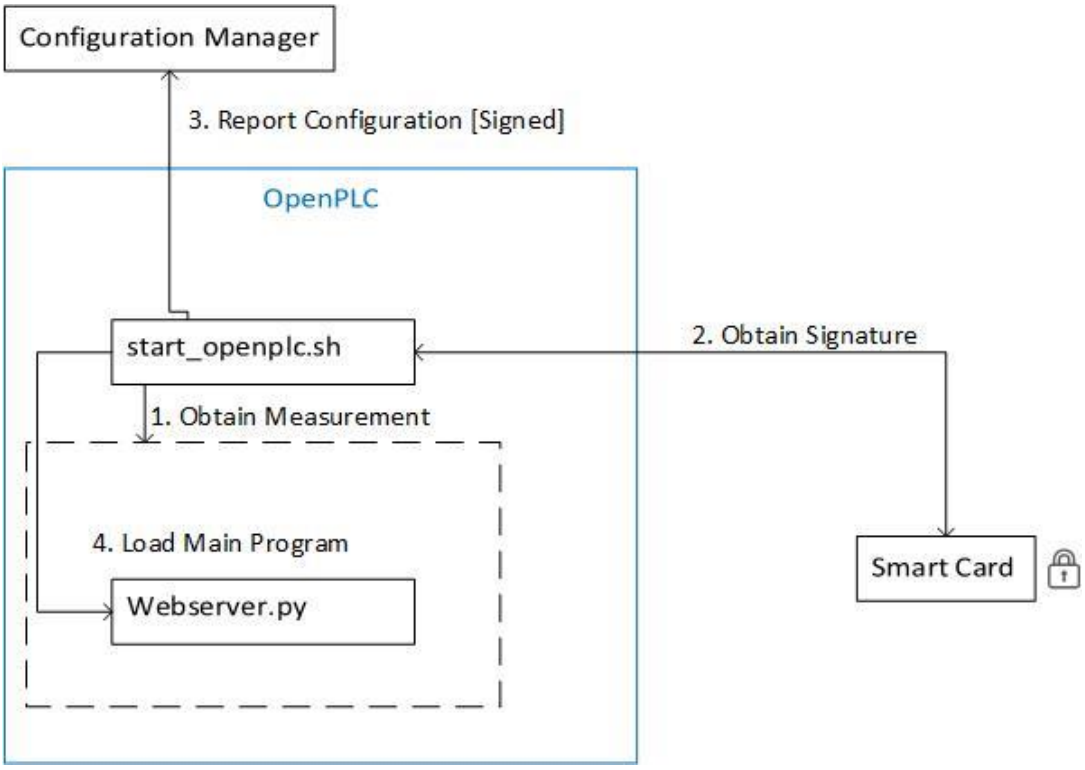

Figure 5 Boot Attestation Procedure

[2] The boot process is a notional example used in testing and not analyzed for security weaknesses or vulnerabilities

The PLC requests for the HROT sign reports to the CMV based on hardware configuration and non-executable data. After the initial startup the PLC will report hardware configuration data in the form of a list of devices connected via USB. Additionally, whenever new logic files are uploaded to the PLC, a report is returned to the CMV, which can then confirm that the logic files match what was supposed to be sent by the engineering workstation.

A similar process is followed when new configurations (PLC logic) are uploaded to the PLC. The PLC is assumed to be in a "trusted" stat and follows the above boot procedure. During a PLC logic change, the logic is measured and reported to the CMV, along with the tools used by OpenPLC to convert the logic into executable code, before compiling and loading the logic for execution.

3.2. Test Case Summary

This security evaluation leverages test cases to determine the viability of the smart card based HROT. These tests were chosen to determine the level of protection provided by the HROT. Level of protection is based on the whether the HROT, when integrated to a device as described above, can provide detection of certain supply chain attacks after the design phase (i.e., the expected phase that would result in the integration of the HROT). These tests also evaluate whether this protection is provided in a timely manner.

A final implementation would integrate the smart card into the boot process of the device as well as during runtime, to ensure that device states are within an accepted range. If the smart card's computing of cryptographic primitives is not timely, it could cause unacceptable delays in the system while waiting for a response. This is unacceptable for devices such as PLCs in NPPs as they are required to perform significant actions within strict time intervals.

Tests are based on postulated scenarios that could occur during supply chain attacks (detailed and defined in [5]). Specific procedures and implementations for tests are presented in Appendix B.

Observations and findings support the initial hypotheses on the level of protections offered by a HROT after the design stage (see Table 1 above). However, instantiating a representative environment and conducting test revealed that there are additional challenges associated with overall implementation, including formalizing the cryptographic primitives used in reporting, ensuring chosen devices support those primitives, establishing acceptable performance ranges for cryptographic operations, and establishing a standardization for reports sent to the CMV.

3.2.1. Baseline

A baseline test was executed to confirm that the instantiated environment allows the CMV to detect a true negative result from the PLC's HROT-enabled report. In other words, the baseline is a test of all the components in working order, without an attack present on the system, to confirm that the CMV receives the information and trusts it without issuing an alert. The CMV receives JavaScript Object Notation (JSON) formatted data that includes the fields listed below in Table 2. The baseline test was successful, proving that the environment is instantiated properly and ready for further testing.

Table 2 Configuration Report Field Summary

Field	Description
Action	This field denotes the process the PLC is executing which resulted in the report. For example, 'start' indicates the PLC's boot procedure, and

Field	Description
	therefore the CMV should expect the hash of the 'webserver.py' program file.
Nonce	This is a number used only once, to ensure the CMV can detect recycled messages and that received reports are "fresh".
Hash	The measurement(s) of the files or data pertinent to the report. This field is in the format of <filename>:<SHA-1 hash>. Multiple files and their hashes may be included, separated by commas.
Signature	The HROT computed digital signature of the hash field, with the nonce prepended. The reporting procedure on the PLC will Base64 encode this field before it is forwarded to the CMV to ensure that data is in an American Standard Code for Information Interchange (ASCII) readable format.

Additionally, timing measurements for various time sensitive operations were taken. This information is vital when making an informed decision on the viability of these devices for digital I&C devices. PLCs in safety critical operations in nuclear power plants require near real time performance, and a delay of a few seconds is not conducive to this fact. Timing measurements taken are shown below in Table 3. The measurements in Table 3 are the result of sending APDU requests for each of the following commands to the HROT (100 times per command).

Table 3 - Timing Measurements for HROT Commands (Values in Milliseconds)

Command	Description	Minimum	Mean	Standard Deviation	Maximum
Get Random	The JCOP card calls a function for secure random number generation and returns 8 bytes of random data.	32	33.98	.75	35
Get Static	The JCOP card simply returns 8 bytes of the APDU command buffer it has received.	31	32.74	.69	34
Get Signature	The JCOP card computes the signature of the data field in the received command APDU and returns it. This signature consists of 256 bytes.	2,412	2,417.06	25.75	2,673
Get Public Key	The JCOP card returns its stored	71	71.92	.44	73

Command	Description	Minimum	Mean	Standard Deviation	Maximum
	public key, 256 bytes.				

Initially, measuring timing for random number generation appeared to give an unexpectedly high result. However, the time measurement included the total time from command APDU to reception of response APDU. Therefore, there exists some uncertainty as to whether the speed is bottlenecked by the computation of the random number or the communication. An insight to this is given by the request for the same amount of data, but without computation of random numbers. The timing for said operation is nearly the exact same as the random number request, except for an additional three milliseconds in the slowest measured times. This is not the case for the signature command, and the get public command which both return the same amount of data. The get signature command, an example of an asymmetric cryptographic operation takes around 2 seconds longer than communicating a similar amount of precomputed/static data. Random data from the HROT is also preliminarily verified to be sufficiently random; no collisions are detected in a sampling of 3,500 requests for a random 8 bytes. Raw data results of baseline testing are available upon request.

3.2.2. Test 1: Malicious Substitution Attacks

Test 1 investigates whether malicious substitutions (see Table 1 above) are detectable in the established environment. This is achieved through an evaluation of the HROT's ability to provide proof of device authenticity. Reports to the CMV are provided to ensure that when a device reports both an expected firmware measurement and a digital signature that corresponds to that measurement and can be verified with the expected devices public key. The baseline tests affirm that the CMV identifies devices that are in a good state and does not raise an alert, and this test begins to explore potential scenarios in which the device will be in an unexpected / malicious state.

The malicious substitution attacks must make an alteration in at least one of three key places, which are (i) the OpenPLC program (ii) the signature and (iii) the private key used to sign the digest of OpenPLC. A malicious substitution would involve a complete substitution of hardware, firmware, or both on a PLC. A common scenario in which this would occur is the case of a counterfeit PLC. Because a counterfeit PLC will not have access to the PLC OEM's Certificate Authority, the counterfeit must include its own public/private key pair to send reports to the CMV. Alternatively, the device may have its firmware substituted for a malicious version. This will result in one or both OpenPLC measurement and the signature included to be altered. The PLC's boot ROM will hash and receive a signature from the HROT for the firmware to be loaded. The substituted firmware must then either forward the true measurement and matching signature or forward a false measurement which will not match the signature from the HROT.

Any of these substitutions result in the CMV issuing a message that the device is either compromised or illegitimate, this is described in detail with examples in Appendix B. [2] found that the basis of security in the HROT stems from the ability to report authentic and non-repudiated information.

Table 4 Summary of Protections Against Malicious Substitution

Substitution Type	Detectable at CMV	Detection Method
OpenPLC program	True	Reported hash deviates from expectation / list of acceptable hashes.
Signature	True	Mismatch detected between reported hash and included signature.
Private Key	True	Signature verification failure, included signature does not verify to any public key in CMV trusted key list.

3.2.3. Test 2: Tampering, Configuration Manipulation

Test 2 evaluates whether the HROT protects against the tampering, configuration manipulation attacks (see Table 1 above). Protection against these attacks requires the HROT to provide reporting on the device's non-executable information such as logic files and configurations. Protections from these attacks are modeled by the device reporting the hash of logic files received by OpenPLC. This would offer protection against attacks such as the Stuxnet attack [9], which resulted in a compromised engineering workstation loading malicious logic onto a PLC. The engineering workstation can report its copy of the logic to the CMV; to ensure that the logic uploaded matches the hash that the PLC reports, it must provide the malicious copy to the CMV. Upon review at the CMV, it is apparent that the logic is malicious, assuming that the logic is reviewed by a qualified engineer at the CMV.

Test 1 established the system's tolerance to basic attacks like substitution of signatures, whereas Test 2 explores the "replay attack", which an adversary may employ to masquerade as a device in a trusted state. A replay attack consists of sending (i.e., replaying) previously captured traffic to a device with the aim to fool the device into performing an undesired action [10]. For this test, the device will attempt to replay a previous message of a received logic file reported to be non-malicious by the CMV.

The process for executing Test 2 involved editing the reporting procedure used when new logic files are received to replay a previously recorded message with a measurement of a genuine logic file regardless of the logic file truly received at the PLC. Something to note here is that this test assumes that a sophisticated attack has already taken place and gone unnoticed, because a change in this procedure should result in a change to the measured program files reported upon boot.

Reports sent from the PLC to the CMV include a nonce, which should only ever be used once and are included in the signed data. The CMV is responsible for storing these nonces and verifying that messages received do not include a previously logged nonce. The CMV can leverage this fact to verify that any incoming report has a nonce field that has not been recorded, and a new and fresh signature. This assumption is supported by the results of this test. The CMV detected replayed reports and displayed an alert message when old reports are received. Specific examples of the CMV's replay attack detection, as well as details of the HROT's nonce generation process are presented in Section B.2.

3.2.4. Test 3: Malicious Insertion

Test 3 interrogates protections against a malicious insertion attack. A malicious insertion occurs when an addition is made to a part of the device rather than being swapped (i.e., malicious substitution). Tests 1 & 2 establish the level of protection in software-based attacks by providing detail on firmware and other software-based configuration reporting. Test 3 is focused on a hardware-based insertion, which is important because hardware authentication and integrity is integral to supply chain security. Common open-source tools available in Ubuntu and other Linux flavors allow devices to query hardware configuration information such as devices connected to the USB and PCI busses [11, 12]. These tools are used to generate a report, which can be signed by the HROT and sent to the CMV.

Test 3 is limited in applicability because the Operating System and hardware used is representative of a commercially available I&C device, but not a perfect match. It is unclear whether the tools used will be functional in a commercially available PLC. The test aims to determine whether hardware changes such as USB device additions are detected at the CMV.

Test 3 consists of hardware measurements which are signed by the HROT and provided to the CMV to verify that hardware changes have not occurred. The hardware measurements that are reported to the HROT can be verified and are resistant to attack (supported by Test 1 & 2 findings), but the HROT is unable to read directly from the devices bus or the system's kernel, relying upon the device to report these measurements accurately and completely to the HROT. This reliance on the device is a result of the HROT acting as a server rather than a client, making it incapable of creating its own commands to issue on the system bus. This reliance may allow for attacker to add a "silent" subcomponent to the bus capable of reading and writing to the system bus and does not report any configuration information during boot. There would likely be other indications of the substitution and the attacker would need to know the messages reported to the HROT.

Regardless, this test utilizes periodic hardware measurements taken by the PLC and forwarded to the CMV. These hardware measurements include the results of a listing of components found on the PLC's USB busses. The findings from this test show that the CMV can create an alert based on hardware-insertion attacks such as the addition of an unapproved USB device being inserted into the PLC.

This page left blank

4. DISCUSSION

The objectives of the tests and analysis was to provide evidence supporting the hypothesis that an integrated HROT significantly lowers the attack success probability (or makes impossible) for certain types of supply chain cyber-attacks that may occur in the supply chain for a digital I&C device. Also, the HROT provide detection of these attacks in a timely manner.

Timing for operations done on the HROT is very important as many of the digital I&C devices will need to provide real time or near real time performance. Testing performed for the baseline evaluation of the smart card used for this research indicates that operations such as secure random number generation are done in a timely manner. The difference between a request for 8 bytes of random data's max recorded time and the max recorded time for 8 bytes of static data was 3 milliseconds, and both performed the same during their minimum recorded time. This indicates that the time required for random number generation on the card is negligible, but there is certainly room for improvement on the communications speed. It should also be noted that the communications occur over a standard USB card reader. The speed of communications would improve through upgrade of the card reader to a faster interface such as USB-C (i.e., the limiting factor of the communications speed in the experiments was the Bus speed). Additionally, there response times would be lower if the device was connected over a bus such as I2C or SPI where the HROT could be directly interfaced. These are supported by the standardization for APDU communications [13].

Digital signature generation was found to have a less than ideal timing, around two seconds. This would mean that inclusion of a digital signature in operations that need to happen very quickly, such as polling of safety-critical functions is not feasible. However, it is still applicable to operations that are less time sensitive, such as during the boot process. Asymmetric cryptography is much slower than symmetric cryptography [14]. It would be possible to use the HROT's secure processing, storage, and asymmetric cryptographic primitives to instantiate symmetric session keys which will be present on the CMV and the HROT. This would allow for significantly faster cryptographic operations, and those keys can be regenerated periodically to ensure that there are no compromises regarding the system's overall security. For example, Microsoft recommends that Kerberos service tickets (used to grant user access to enterprise IT resources) be regenerated every 600 minutes [15].

Test 1 results demonstrate that the implemented trusted boot attestation provides proof of the PLC's base state (i.e., authenticity). A malicious substitution involves a complete substitution of the device or part of the device. This will result in changes to the report, which the CMV will detect as deviations from the base state and therefore indicators of compromise. A partial substitution will result in changes to the hash of the OpenPLC software or the reported signature, and a complete substitution or counterfeit must use a different private key. This test shows that the CMV can detect either one of these scenarios.

Test 2 provides two important results. The first result is that inclusion of a secure nonce prevents replay attacks in the testing environment. The second result is that the HROT provides the ability to securely generate random data in a timely manner. Combining these two results brings the logical conclusion that it is possible to develop a replay resistant reporting procedure using the HROT to report information that prevents tampering and configuration manipulation attacks.

However, the current test bed design is limited by not handling extended APDU (i.e., larger messages) resulting in insufficient size in the APDU to include both the nonce and the signature. This limits the protection offered by the HROT as the signature cannot be verified because there is an "fresh" nonce prepended to the message. This is a limitation of the current testbed

implementation and not the HROT technology which could leverage the extended APDU to overcome this challenge.

A possible solution could be to implement a secondary command in which after the signature is returned the device sends an additional APDU requesting the last random nonce used for signature generation. Preferably, the implementation would utilize a signature routing such as Elliptic Curve Digital Signature Algorithm (ECDSA) which implements a random nonce by default [16]. It is vital that a final solution ensure that random number generation be secure and used only once, as improper use of random numbers or faulty random number generation can have critical impacts, such as allowing recovery of the private key [17]. Newer versions of Java Card and other smart card Operating Systems support both extended APDU and ECDSA.

Test 3 used tools for hardware measurements that rely on the PLC OS's understanding of its hardware, which is built during the boot process. During boot, the BIOS requests configuration information for subcomponents on the system bus. This leads to an implicit trust between the subcomponents and the PLC, and therefore an implicit trust between the greater OT environment and the PLC's subcomponents. It may be possible that a device that can serve as a client (such as a Field Programmable Gate Array (FPGA)) to perform an analysis of the system bus using timing or power analysis to improve the accuracy and completeness of the information provided to the HROT. In short, Test 3 shows that the HROT provides protections against malicious insertion attacks if there is an accurate way to measure the hardware components of the device.

5. CONCLUSION

This report details the testing environment, tests, and findings related to personalizing a general use smart card for the purpose of a HROT in a digital I&C device. The testing performed indicates that the environment developed for this effort eliminates multiple high consequence supply chain attacks on digital I&C devices for NPPs. By providing verifiable and trusted data on device configuration and state information, attacks which rely on changing those parameters to achieve a malicious end goal are detected by the HROT or forced to become much more sophisticated (e.g., compromise of TBB). This increases adversary time and cost of a developing and deploying a successful attack. Therefore, the risk associated with these attacks would decrease as it would require defeat or bypass of approved cryptographic mechanisms or secure boot mechanisms.

The research has found that assumptions on protection from supply chain attacks, listed in Table 1, are qualitative but can be improved based upon the findings of this report. The HROT demonstrated detection of the evaluated supply chain attacks and should lead to increased confidence of the against both tampering/configuration manipulation attacks and hardware substitution attacks and replay attacks, with less protection provided against malicious insertion attacks.

The HROT protections against malicious substitution attacks were confirmed by Test 1. The PLC with a HROT can respond with a trusted measurement of its firmware and that the signature confirms the HROT and therefore device identity. For an attacker to provide a counterfeit, they must either extract the private key from a genuine HROT or remove the HROT of a genuine PLC and insert into the counterfeit. Both attacks will leave significant evidence of physical tampering, and require significant expertise, time, and financial contributions by an attacker. Test 2 also provides that non-executable information can be reported in a similar manner, protecting against attacks that rely on configuration information rather than manipulation of firmware or other device code.

Specifically, malicious insertion attacks (modeled in Test 3) are not completely protected against by the HROT in the specified environment. This is due to the HROT being unable to actively access the system bus, therefore it must rely on the PLC underlying firmware to provide an accurate listing of hardware components. Sophisticated attackers may be able to spoof as trusted subcomponents to the system BIOS or skirt requests for information completely.

Evidence shows that the smart card can provide the three benefits listed:

- The HROT is plug and play, meaning that there is no need for configuration when the device is received by the operator, and the device OEM could purchase the card with the applet and keys already generated on the card meaning it will simply need to be soldered, and a mapping of the card to the device needs to be signed. This is shown in the test bed environment; the smart card and reader can be transferred to another device with the same startup procedure, and the device will be accepted.
- The HROT reduces operator workload by providing trusted reporting on the device configuration details. This reduces operator workload significantly during Factory Acceptance Testing (FAT). Beyond FAT, the device can leverage the HROT for reporting during reboots, configuration changes, logic updates, and any other event defined to report to a secondary device. Secure reporting on device state information and the elimination of some supply chain attacks allows for an Operator to have increased trust in the device and therefore spend less time and money confirming its security.

- The HROT provides an I&C device with sophisticated authenticity technology. The HROT enables the device's authentication reports to be cryptographically secure and move communications and authentication of NPP devices to a level already common in enterprise IT environments.

Test bed development and test execution resulted in complications that identify areas for further research and development needs. The smart card used as the HROT in this environment does not support cryptographic primitives and other features such as extended APDU that would allow for a more seamless and future-proof implementation. A smart card with a more modern suite of cryptographic support would remove the need for workarounds in this research, such as nonce generation by the PLC; this should take place in a trusted environment, i.e., the HROT. Further research is also required to determine what level of confidence can be gained against a hardware-based malicious insertion attack.

6. FUTURE WORK

Future work also includes the establishment of an agreeable schematic for a shared Public Key Infrastructure (PKI) system among operators and manufacturers. The environment researched in this report does not include certificate authorities or trusted certificates from device manufacturers for device states, HROT public keys, etc. A PKI scheme must be established with some agreement from multiple vendors to establish an autonomous system for trusted devices. Smart card manufacturers may provide the OEM with established cards and a certificate proving that they are from the smart card manufacturer themselves, or OEMs could receive "unfused" cards which they must then load with the proper programs and set to the proper lifecycle to ensure they cannot be tampered with. The OEM should then establish a mapping of devices and smart cards and this information should also be signed so that the operator can verify it. This requires that OEMs maintain a strong security culture and ensure that they do not experience a compromise of their certificate signing procedure. Operators should vet device OEMs accordingly to ensure that risk is low, and that if a compromise were to occur the OEM would report such an incident in a timely and responsible manner.

Overall, a smart card based HROT can provide operators with significant gains in security while also reducing the required workload. Testing conducted for this paper proves that if the device and smart card are correctly initialized and configured at the hardware assembly stage, there is little, or no work required by the operator to establish trust that the device is genuine and correctly / safely configured. The smart card is an established piece of hardware that meets the necessary requirements for providing a trust anchor, but there is more work to be done in establishing the method by which the root of trust is used. For operators to seamlessly integrate trusted devices, there must be a consensus on the PKI used by manufacturers. There must also be an agreement on the specific mechanism for authentication of devices, like that used by the telecommunications industry.

This page left blank

REFERENCES

1. ENISA, *ENISA Threat Landscape for Supply Chain Attacks*. 2021, European Union Agency for Cybersecurity (ENISA).
2. Rowland, M. and B. Karch, *A Review of Technologies that can Provide a 'Root of Trust' for Operational Technologies*. 2022, Sandia National Laboratories.
3. TCG, *TCG Specification Architecture Overview*. 2007, Trusted Computing Group.
4. Rowland, M., et al., *Managing Cyber Security Supply Chain Risks for the Security of Radioactive Materials*, in *International Conference on Safety and Security of Radioactive Sources – Accomplishments and Future Endeavours*. 2022: IAEA Headquarters Vienna, Austria.
5. Eggers, S. and M. Rowland. *Deconstructing the Nuclear Supply Chain Cyber-Attack Surface*. in *Proceedings of the INMM 61st Annual Meeting*. 2020.
6. IAEA, *Procurement Engineering and Supply Chain Guideline in Support of Operation and Maintenance of Nuclear Facilities*.
7. NIST, *Zero Trust Architecture, SP 800-207*. 2020.
8. Tomlinson, A., K. Mayes, and K. Markantonakis, *Introduction to the TPM*. 2007. p. 155-172.
9. Falliere, N., L.O. Murchu, and E. Chien, *W32.Stuxnet Dossier Version 1.3*. 2010, Symantec Security Response.
10. MITRE. *CWE-294: Authentication Bypass by Capture-Replay*. Available from: https://cwe.mitre.org/data/definitions/294.html.
11. *Linux USB Project*. Available from: http://www.linux-usb.org/.
12. Mares, M. *The PCI Utilities*. Available from: https://github.com/pciutils/pciutils.
13. Commission, I.O.f.S.I.E., *ISO/IEC 7816-4*, in *Organization, security and commands for interchange*. 2020.
14. Panda, M., *Performance Analysis of Encryption Algorithms for Security*, in *nternational conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016*. 2016, Sambalpur University.
15. Microsoft. *Maximum Lifetime for Service Ticket*. 2021; Available from: https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/maximum-lifetime-for-service-ticket.
16. *FIPS PUB 186-4*, in *Federal Information Processing Standards Publication*. 2013, NIST. p. 130.
17. Schmid, M., *ECDSA - Application and Implementation Failures*. 2015, UC Santa Barbara.
18. *NXP J3A040 and J2A040 Secure Smart Card Controller Rev. 3 Evaluation Documentation*. 2011.
19. Maletsky, K., *RSA vs. ECC Comparison for Embedded Systems*. 2020: Microchip Technology Inc.
20. Society, T.I., *PKCS #1: RSA Cryptography Specifications Revision 2.0*. 1998.
21. Group, S.-I.C.W., *SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms*. 2020.
22. Alves, T. *OpenPLC*. Available from: https://www.openplcproject.com/.
23. NIST, *Digital Signature Standard (DSS)*. 2013.

This page left blank

APPENDIX A. TEST BED DESIGN

For the testing required by this project a testbed was developed to be representative of a digital I&C with an integrated smart card. The smart card used is a Java Card Open Platform (JCOP) smart card providing the root of trust capabilities that are the subject of the testing. The card is a NXP JCOP J2A040, certified to Common Criteria EAL5 with advanced vulnerability assessments [18]. The card has a total of 40K of total memory. APDUs can be sent to the smart card to request information on memory available. The cards responses indicate 17K of persistent memory, which acts like a hard drive, and 1.3K of transient memory, which acts like typical RAM. The rest of the 40K is likely used to store the card's operating system and the Java Card Virtual Machine. The card's operating system is Java Card 2.2.2, which has since been succeeded by version 3. Newer Java Cards have added more cryptographic primitive support that would allow for schemes that require less storage for keys and less processing time for encryption and signatures. For example, ECC is faster than RSA by a factor of ten [19]. Because the acquired smart card does not support ECC, it uses RSA 2048-bit Chinese Remainder Theorem algorithm [20] to digitally sign information with a static key stored statically within the applet's CAP file. RSA for encryption and digital signatures is considered secure, but only under certain circumstances where padding and formatting attacks are avoided, otherwise it is considered a legacy cryptographic scheme [21]. The applet developed and loaded onto the card implements the necessary features to act as a root of trust. A description of APDU commands that are handled by the card applet are listed in Table A-1.

Table A-5. Root of Trust Applet Commands

Command	INS code	Returned APDU
Get RSA Public Key	0x10	Returns the RSA public key modulus by default, and exponent if P1 is set to 0x01.
RSA Sign	0x11	Digitally signs the data provided using the private key and returns the generated signature.
RSA Verify	0x12	Returns the standard response (0x9000) if the provided signature and data are a match, and an error if not. This process can be accomplished with the public key without access to the card as well.
RSA Cipher	0x13	Generates an encryption of the provided data and returns the encrypted data. Currently encrypts to its own public key. Further work is required to load other public keys to encrypt to.
Generate Random Data	0x14	Generates and returns 8 bytes of securely generated pseudorandom data.

Connected to the smart card is a Raspberry Pi 4, which serves to represent a PLC. This is accomplished by the open-source software OpenPLC [22]. The test bed allows for the hardware to be looped into a larger system, as the OpenPLC program can generate Modbus traffic based on the

uploaded logic files and sensor data. The Raspberry Pi runs Ubuntu 20.04 LTS as an operating system, allowing for easy access to system controls such as services. OpenPLC is installed as a service, executing a startup script upon boot. The startup script has been modified such that before launching the webserver and the rest of the functionality, the hash of the webserver program is signed and sent to the card along with some random number. The results for the command 'lshw' follow the same procedure. The measurements are then passed along to a third party for review, which is called the CMV. Additionally, OpenPLC lets users upload logic (ST) files. The files are then compiled by a tool in OpenPLC. The webserver routine has been modified so uploading logic files now also reports the digests of the logic file received, the compiler tool, and the compiled logic code. This is also combined with a random number acting as a nonce to the configuration CMV.

The CMV is also implemented as a Raspberry Pi 4 running Ubuntu 20.04. The CMV is loaded with software in the form of a python program, which constantly listens for reports from the PLC Raspberry Pi. The CMV outputs received information to the console and performs integrity checks on the reports. Integrity checks consist of signature and nonce verification. The CMV currently implicitly trusts a list of public keys. If a report is received and not signed by any keys in the trusted list, the device must either be experiencing a major fault, or the device is not in an expected configuration. If the device sends a report with a nonce that has been previously recorded it is highly likely that the device is attempting a replay attack. This would indicate that the device has been compromised or is an asset.

APPENDIX B. TEST RESULTS IN-DEPTH

B.1. Test 1 - Malicious Substitution

A simple way for a device to provide confidence in its supply chain security is to provide an attestation of its integrity. This is accomplished through a representation of a cryptographically secure boot attestation. During the boot process, the device takes measurements at each stage (depicted in Figure 6) and reports those measurements back to the CMV. The smart card can be used to prove the authenticity and integrity of the measurements using a digital signature algorithm.

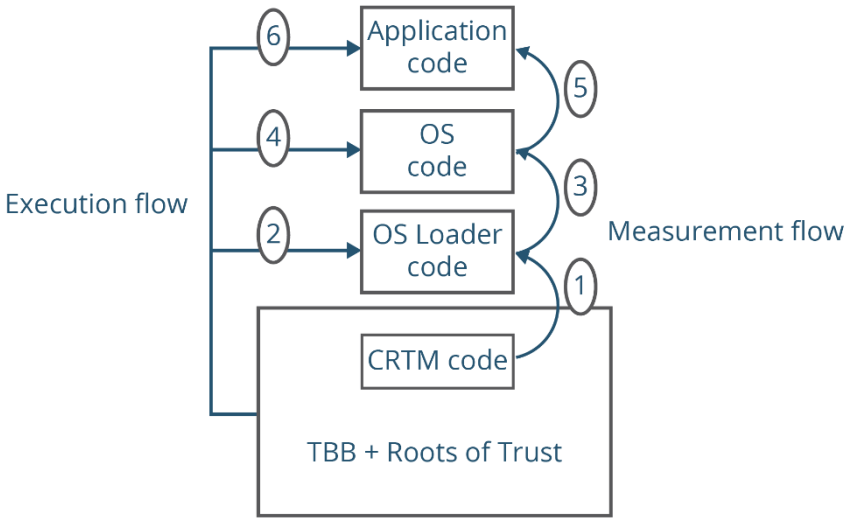

Figure 6 Trusted Boot

Because the smart card is the only place w the private key, you know it genuinely came from the smart card. The attacker is not able to generate a signature without the smart card. We verified that a CMV detects the changes. The OpenPLC software is started by a script with emulates a Trusted Boot Block (TBB) which cannot be changed. This script takes a measurement of the "webserver.py" file which is the main program file for OpenPLC. Additionally, when a new ladder logic file is uploaded to be used, the ladder logic, the compiler, and the compiled logic file are all measured and returned to the CMV. OpenPLC and its startup script are not perfectly representative of a commercial PLC or its boot ROM (Read-Only Memory), but this process shows that it is feasible to use a smart card to provide a cryptographically assured measurement of a device's firmware and configuration data.

The information sent from the PLC can be verified by the CMV to confirm that the signature is verified to be for the included measurement, and that the measurement is signed by the expected smart card. An example of a verifiable and unverifiable measurement is shown in Figure 7 and Figure 8, respectively.

Figure 7 Verified Measurements – No Substitution

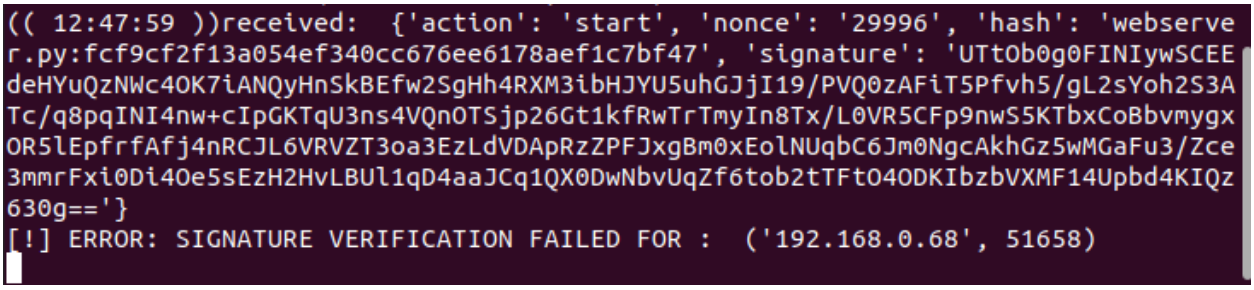

Figure 8 Failed Signature Verification – Substitution of Signature

The CMV can then alert if the device is out of a trusted state or if the message received does not have a signature verified with the correct public key. Something to note here is that the configuration should not simply keep a mapping of devices and their secure element's public key on record to verify measurements. Instead, a Public Key Infrastructure (PKI) must be developed to provide a secure way the OEM can provide a verifiable certificate for each device's public key. This lowers the probability of an attacker inserting an untrusted public key into the CMVs trusted list, and it would require compromising the private key used by the OEM to sign the device certificates.

The addition of a similar routine into a digital I&C device would significantly lower the number of successful attacks that can occur, reducing overall risk to the device and plant from compromise via cyber-attack. Depending on the implementation, verifying firmware and configuration information increases demands on the attacker attempting to introduce any kernel level module onto the device, and will be forced to utilize only the standard functionalities of the device to introduce malicious modules. The device should verify all applications or scripts that are run at boot of the device, which also makes it much less probable for an attacker to achieve persistence on the device. The specifics of the implementation are vital to the success of this countermeasure, though. For example, if only a selection of kernel modules is measured, a new imported module or a compromised unmeasured module could give the attacker access to the device. Any implementation is an improvement over the current solution which is an implicit trust in the device after factory acceptance testing.

B.2. Test 2 - Replay Attack

A common attack to circumvent protections created by reported information is a replay attack, defined be CWE-294 [10]. These attacks are difficult or infeasible to defeat without the use of cryptography, which is provided by the smart card. This number generation is also verified by Common Criteria in most smart cards, and specifically in the JCOP card used for this paper. This random data can be integrated into the digital signature in the form of a nonce. The CMV keeps a record of reported nonces and can easily detect a replay by comparing the current nonce to the previous reports (shown in Figure 9), ensuring that should a device attempt to replay a previously used measurement report, the CMV detects and alerts that the device is potentially compromised

and should be investigated further. Additionally, the CMV performs a check to ensure that a nonce is included in the message. If the nonce is missing or insufficient in size (e.g., only one byte long), the CMV will initiate the same alert.

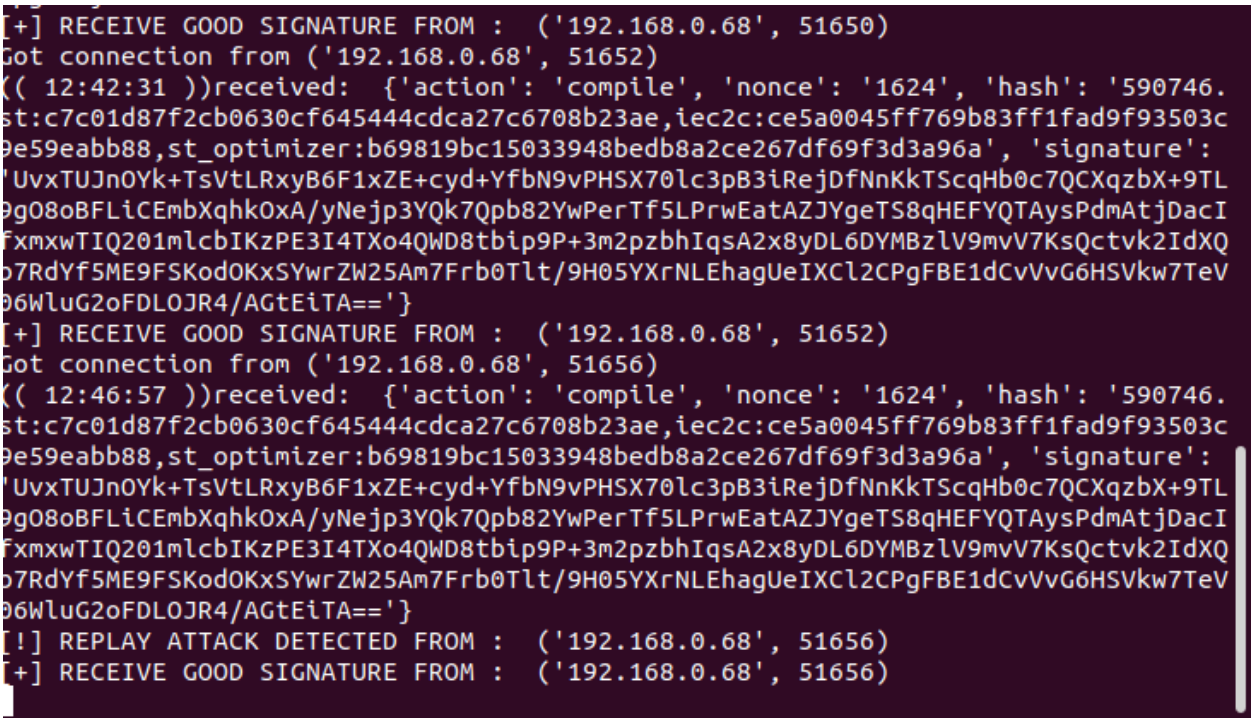

Figure 9 Replay Attack Detection

The representative architecture developed for this paper makes use of a nonce to aid in prevention of the replay attacks. The nonce is incorporated into the measurement and the signature is reported to the CMV. However, because of the limitation in length of APDU responses, the nonce is not able to be included with the signature in the response in plaintext, meaning that there would need to be another command to obtain the nonce used. The APDU can respond with 256 bytes of data, and the signature is also 256 bytes. This means that at no point can the HROT used in this research respond with both data and its signature.

For now, the nonce is generated by the host device and passed to the smart card, which is not ideal. One solution would be to use extended APDU, allowing for the card to return a signature for both the nonce and the measurement. This solution is shown below in Figure 10. This provides authenticity that the nonce used was indeed generated by the smart card, and the signature can then be verified with the reported nonce. Another solution is to use a digital signature algorithm such as ECDSA, which includes a random number by default [23]. ECDSA also provides a faster signature [19]. The smart card used for this effort does not support extended APDU or ECDSA by default, but newer versions of the JCOP card provide these features.

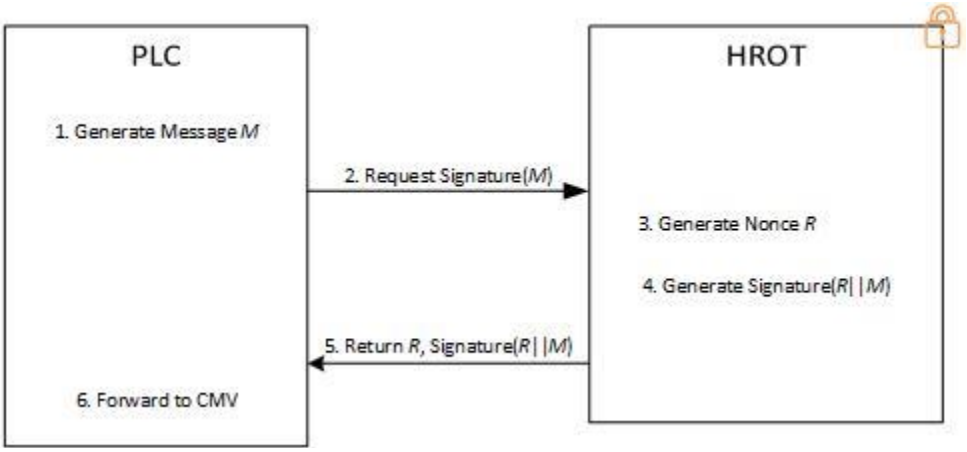

Figure 10 Secure Nonce Inclusion

In this scenario, the nonce generated by the HROT is 64 bits, and the HROT and PLC are not required to maintain a list of previous nonces as the probability of generated the same nonce twice assumed to be very small.

Defining the probability of collisions is commonly referred to as the "Birthday Problem" and can be approximated using

$1 - e^{-n^2/(2d)}$, where n is the amount of numbers generated and d is the amount of numbers possible.

Using the above formula, the HROT could generate a new nonce every one second for a calendar year (31,536,000) and the probability of generating a collision between any two numbers in that time would be approximately 0.000027.

Alternatively, for operations where the PLC has a network stack initiated (i.e., not during the boot process) the PLC could request a nonce from the CMV and include that nonce rather than requiring the HROT to generate it. This is only a notional example for a reporting mechanism and demonstrates that the HROT is capable of providing an acceptable level assurance to the CMV that the report received from the PLC is new and not replayed.

It is possible that the card could generate random numbers ahead of time to reduce timing delays associated with secure random number generation during the signature process. These stores of random numbers could be generated dozens at a time and stored in non-volatile memory. There are also other solutions that should be explored that may provide protections against replay attacks. For example, the HROT could be used to instantiate a session key with the CMV, which would only provide valid communications between the devices for a temporary amount of time. A potential issue with such a solution is that the PLC will likely not have the proper drivers loaded at the time of measurement to establish the network protocol necessary to negotiate a session key with the CMV. Though, the HROT could store the received measurement during the boot phases, and then provide the signed version (using a session key) once the PLC has finished its boot procedure. Another solution could be that the HROT contains a counter in non-volatile memory that is incremented with each signature. The counter will come with similar problems as a random nonce, though. The PLC will have to maintain its own copy of the counter and supply it to the CMV with each signature. If the PLC and the HROT ever become unsynchronized, the counter (that the PLC thinks is correct and therefore sends to the CMV) and message concatenation will not verify correctly to

the signature the HROT provides. A resynchronization procedure would need to be implemented to remedy this possible issue.

B.3. Test 3 - Hardware Changes

Similarly, to detecting firmware changes, hardware measurements can be signed and reported back to the CMV. Additionally, the card applet has enough offline data storage to maintain a listing of the hardware that should exist in the system. In this manner the boot sector can report the management to the card and receive the digital signature of its measurement. Along with the response APDU is always two status bytes. These status bytes can be used to indicate whether the reported information matches that which is stored on the card. The current boot sector is then able to either continue or terminate based on the response from the card.

The described procedure would still prove to be an improvement from contemporary I&C devices. Research conducted for [2] found that digital I&C devices at most report to implement a secure boot, but the specifics are generally not released. Secure boot typically indicates a simple check of the boot sector to be loaded before executing. This system allows for a secure and non-repudiated boot attestation which includes a hardware measurement. However, it should be noted that the test bed assumes a fully booted Linux-based device. This gives access to common GNU commands such as 'lspci', which returns a listing of all ICs on the Peripheral Component Interconnect (PCI) bus. Further research is required to determine the earliest possible time at which to take a measurement of hardware, where kernel modules needed for such commands may not be loaded. Additionally, it is possible that ICs on the board could provide a fake or spoofed response to the request for information initiated by 'lspci'. Because of the relationship between the devices at large and the smart card being a client and server (respectively), the card is unable to verify information given to it.

Figure 11 provides examples of both a successful hardware measurement report, and one where a malicious insertion has been detected. The first report raises no alert at the CMV, and the signature is verified. The second report raises an alert for invalid hardware configurations. Upon closer inspection, it is revealed that a potentially malicious device has been added to the PLC, which is shown as "Emtec USB Disk 2.0".

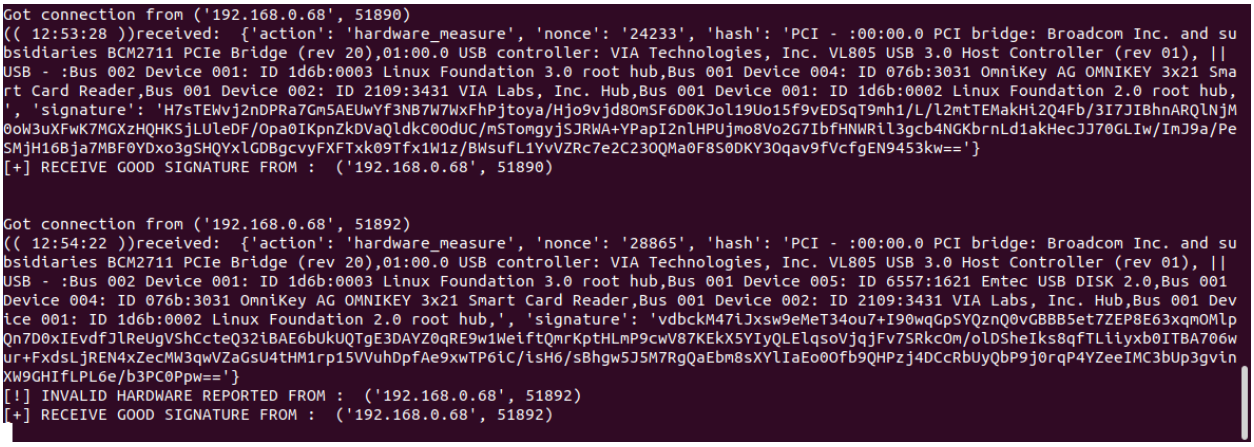

Figure 11 Hardware Measurement Examples

More research should be done to verify whether an FPGA would be better suited for the purpose of hardware component verification. PCI components contain a set of configuration registers which

can be accessed by the devices BIOS. An FPGA on the device bus may be able to access the registers of those devices to verify their configuration information as well potentially perform timing analysis on the devices which gives insight into whether the device may be a counterfeit.

B.4. Other Observations

A selection of 100 signature requests were sent to the HROT to be subject to a timing analysis. The fastest recorded response time was 2.412 seconds, and the slowest time was 2.673 seconds. Requests for the HROT public key (also 256 bytes) take between 71 and 73 milliseconds. This indicates that a significant portion of the time is spent by the HROT performing computation of the signature.

Research conducted for this paper confirms that smart cards can provide enough offline memory to store files and configuration information needed for a trusted boot process and trusted device reporting. The JCOP smart card that was acquired contains a total of 40 kilobytes of storage. Memory available to the applet at runtime was 1.8 kilobytes. This means that the smart card can easily process more than a kilobyte of information and can offline store tens of kilobytes. An embedded I&C device would not need more data than this to store crucial security information such as keys or an initial boot sector.

The card acquired for this effort responds to default Global Platform keys. Another consideration is that smart cards must be loaded with keys for maintaining their lifetime that must be closely maintained and guarded by either the card's manufacturer or the device's OEM. The card must also be set to the appropriate Global Platform lifecycle state to make sure that it cannot be loaded with additional applets or have the root of trust applet deleted or modified without proper permissions.

Two more tests were conducted to establish the behavior caused by an incorrectly initialized card. This could take place in one of two scenarios. The first scenario being that the card is initialized correctly but there is an issue that causes the CMV to not receive the public key for the device. The results are shown in Figure 12, where during the first report, the CMVs keys file does not include the cards public key, and then it is loaded, and the device is rebooted. Without the device's public key in the trusted list, the CMV cannot verify the signature received. So, this would result in a false positive detection of compromise.

Figure 12 Uninitiated CMV

Alternatively, it is possible that the OEM may install the card onto the device but fail to correctly initialize the card. This could cause the device to not have the root of trust applet installed. In this case, the selection of and communication to the smart card fails. This results in the device reporting its configuration information without any signature, because it is unable to generate one (shown below in Figure 13). Without a signature included in the report, the CMV will alert that signature verification failed. This would also be a false positive detection of compromise, though in both scenarios there is a fatal configuration error that would need to be resolved.

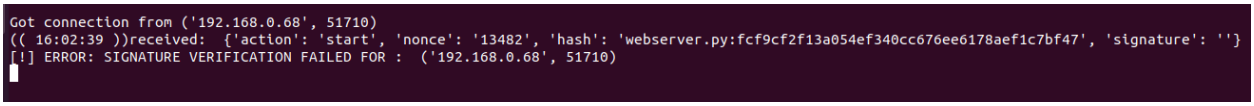

Figure 13 Uninitiated Smart Card

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Lon Dawson	08851	ladawso@sandia.gov
Michael Rowland	08851	mtrowland@sandia.gov
Benjamin Karch	08851	brkarch@sandia.gov
Technical Library	1911	sanddocs@sandia.gov

Email—External

Name	Company Email Address	Company Name
Katya LeBlanc	Katya.LeBlanc@inl.gov	INL
Shannon Eggers	Shannon.Eggers@inl.gov	INL

This page left blank

This page left blank

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