Trusted Software Updates for Secure Enclaves in Industrial Control Systems

Abhinav Gunjal

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Cameron D. Patterson, Chair
William T. Baumann
Pratap Tokekar

August 11, 2017
Blacksburg, Virginia

Keywords: Industrial control systems, programmable logic controller, industrial control systems security, secure enclaves, software updates, configurable system-on-chip
Copyright 2017, Abhinav Gunjal

Trusted Software Updates for Secure Enclaves in Industrial Control Systems

Abhinav Gunjal

(ABSTRACT)

Industrial Control Systems (ICSs) manage critical infrastructures such as water treatment facilities, petroleum refineries, and power plants. ICSs are networked through Information Technology (IT) infrastructure for remote monitoring and control of physical processes. As ICSs integrate with IT infrastructure, IT vulnerabilities are carried over to the ICS environment. Previously proposed process controller security architectures maintain safe and stable plant operation even in the presence of attacks that exploit ICS vulnerabilities. Security architectures are process control system-level solutions that leverage isolated and trusted hardware (secure enclaves) for ICS security. Upon detecting an intrusion, the secure enclave switches control of the physical process to a high assurance controller, making a fail-safe plant operation.

The process control loop components have an average lifespan of several decades. During this time, electromechanical components of process control loop may undergo aging that alters their characteristics and affects control loop performance. To deal with component aging and to improve control algorithm flexibility, updates to control loop parameters are required. Plant model, process control loop system specifications, and control algorithm-based security mechanisms at the secure enclave require parameter updates. ICSs have hundreds of process control components that may need be installed in hazardous environments and distributed across hundreds of square kilometers. Updating each component physically may lead to accidents, expensive travel, and increased downtime. Some ICS have allowable downtime of only 5 minutes per year. Hence, remote updates are desirable.

A proposed dedicated and isolated hardware module at the secure enclave provides authentication of the update and ensures safe storage in a non-volatile memory. A protocol designed for update transmission through an untrusted ICS network provides resilience against network integrity attacks such as replay attacks. Encryption and authentication of the updates maintain integrity and confidentiality. During the normal plant operation, the hardware module is invisible to the other modules of the process control loop. The proposed solution is implemented on Xilinx Zynq-7000 programmable System-on-Chip to provide secure enclave updates.

Trusted Software Updates for Secure Enclaves in Industrial Control Systems

Abhinav Gunjal

(GRNERAL AUDIENCE ABSTRACT)

Industrial Control Systems (ICSs) manage critical infrastructures such as water treatment facilities, petroleum refineries, and power plants. ICS process controllers interpret sensor output and depending on the set point, generate input signals for the actuator to control physical processes. The process controllers receive set points and periodically send process state to the supervisory network. For remote monitoring and control of physical processes, ICSs are networked through Information Technology (IT) infrastructure. As ICSs integrate with IT infrastructure, IT vulnerabilities are carried over to the ICS environment.

Previously proposed process controller security architectures maintain safe and stable plant operation even in the presence of attacks that exploit ICS vulnerabilities. Security architectures are process control system-level solutions that leverage isolated and trusted hardware (secure enclaves) for ICS security. Upon detecting an intrusion, the secure enclave switches control of the physical process to a high assurance controller, making a fail-safe plant operation.

The process control loop components have an average lifespan of several decades. During this time, electromechanical components of process control loop may undergo aging that alters their characteristics and affects control loop performance. To deal with component aging and to improve control algorithm flexibility, updates to control loop parameters are required. Plant model, process control loop system specifications, and control algorithm-based security mechanisms at the secure enclave require parameter updates. ICSs have hundreds of process control components that may need be installed in hazardous environments and distributed across hundreds of square kilometers. Updating each component physically may lead to accidents, expensive travel, and increased downtime. Some ICS have allowable downtime of only 5 minutes per year. Hence, remote updates are desirable.

A proposed dedicated and isolated hardware module at the secure enclave provides authentication of the update and ensures safe storage in a non-volatile memory. A protocol designed for update transmission through an untrusted ICS network provides resilience against network integrity attacks such as replay attacks. Encryption and authentication of the updates maintain integrity and confidentiality. During the normal plant operation, the hardware module is invisible to the other modules of the process control loop. The proposed solution is implemented on Xilinx Zynq-7000 programmable System-on-Chip to provide secure enclave updates.

Acknowledgments

I would like to thank my advisor, Dr. Cameron Patterson, for providing me an opportunity to work on this project. His guidance, support and patience are valuable to the completion of this thesis. Thanks to Dr. William Baumann and Dr. Pratap Tokekar for being part of my advisory committee.

I am extremely thankful to my parents, brother Abhijeet and other family members for their constant support and encouragement. A special thanks to my friends Anshuman Verma, Vikas Balapal and Ameya Khandekar who made my life in Blacksburg enjoyable.

This material is based upon work supported by the National Science Foundation under Grant Number CNS-1222656. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

ZedBoards and design tools were donated by Xilinx, Inc.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Control systems have been in existence for over 2000 years. As per Arabic and Greek manuscripts, Ktesibios invented the first control system to maintain a constant outflow rate in water clocks using automatic water-level regulation [5]. Over the years, technological advancements led to the evolution of process control systems in the manufacturing industry, power plants, transportation industry, etc. These Industrial Control Systems (ICSes) evolved from mechanical to computerized control reducing human labor, and improving production quantity as well as quality. ICSes are networked through Information Technology (IT) infrastructure to provide real time remote monitoring and control of the physical processes.

1.1 Industrial Control Systems

Figure 1.1 shows an abstract view of a conventional ICS leaf node. ICS leaf nodes are the only part of the ICS infrastructure that interacts directly with the physical process. ICS leaf node contains a control loop that includes sensors, actuators and the production controller. The production controller interprets sensor output and depending on the set

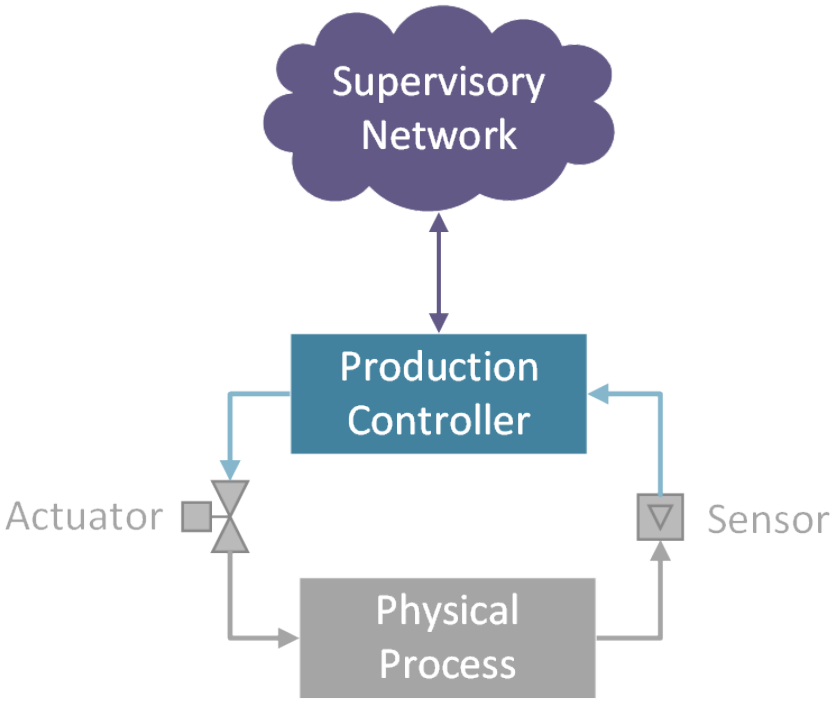

Figure 1.1: Conventional ICS leaf node

point, generates input signals for the actuator to control physical processes. The production controller receives set points and periodically sends process state to the supervisory network, incorporates the plant network and IT infrastructure required for command generation and remote monitoring of the physical process. Integration with IT infrastructure facilitates reliable and real time communication between humans and ICS control loop components.

1.2 ICS Security

An ICS directly interacts with the physical world. Any security vulnerability in an ICS may lead to infrastructural damage, loss of human lives, adverse effects on production, financial losses for an organization, and may even negatively affect a nations economy. Though the IT infrastructure facilitates remote monitoring and control of physical processes, it may also

expose an ICS to cyber security vulnerabilities. Traditional IT security solutions are not sufficient to protect the integration of cyber and physical components of the ICS. This is reflected in a continuous increase in attacks on ICSs. In 2015 alone, ICS Cyber Emergency Response Team (ICS CERT) responded to 295 incidents, which is a 20 percent increase over 2014 incidents [6]. In June 2010, the Stuxnet computer worm damaged a uranium nuclear facility in Iran [7]. On December 23rd, 2015, a cyber attack on a Ukraine regional power company led to power outages for 225,000 customers for three hours [8].

1.2.1 German Steel Mill Attack

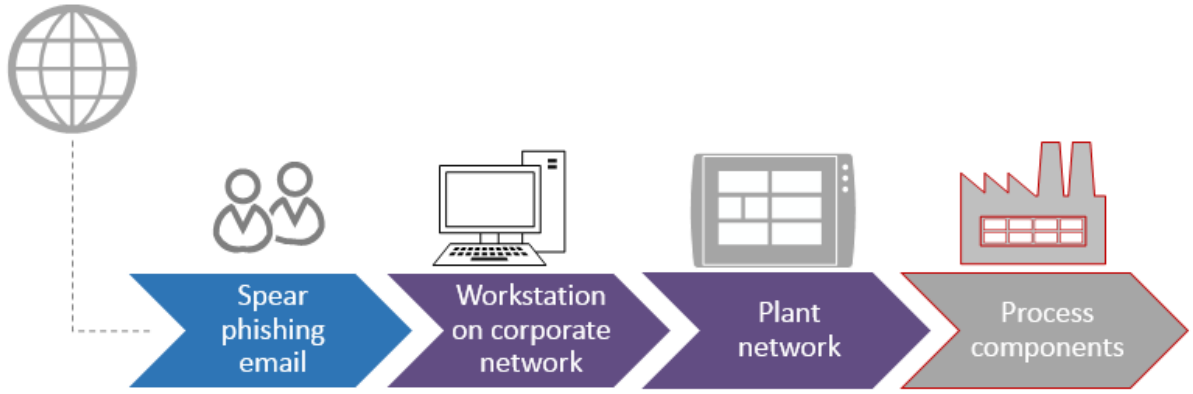

Figure 1.2: Attack Flow

In December 2014, a German steel mill witnessed an attack on its physical infrastructure [9]. The adversary infiltrated the corporate IT network to the plant network and was able to cause multiple components of the ICS to fail, leading to massive physical damage to the mill. The attack on the steel mill was a spear phishing attack that involved an email containing an attachment (doc, pdf). When opened, it planted malware on a corporate workstation of the IT network. The malware opened up a network connection for the adversary. Information available on the corporate IT network helped the adversary learn more about the system

internals. The adversary was then able to travel through trust zones and connections between the IT network and a plant network. As the adversary penetrated the plant network, prior knowledge about the plant ICS helped them compromised multiple components associated with the critical physical process (Figure 1.2). The attack led to massive physical damage.

Traditional IT infrastructure security measures address remote, external threats but do not protect an ICS from threats such as malwares that can be introduced through the IT network, false sensor data injection attacks, etc. Any intrusion into the IT infrastructure may give access to ICS leaf node components which directly interact with the physical process and associated infrastructure. Any violation of the ICS leaf node integrity jeopardizes physical process integrity. Hence, ICS requires additional security measures for the integrity of the computational and control components at the ICS leaf nodes. Previously proposed ICS leaf node security architectures (LNSAs): Trustworthy Autonomic Interface Guardian Architecture(TAIGA)[10], Secure System Simplex Architecture(S3A)[11], and Controller Controller (C^2) policy enforcement mechanism [12] ensures safe interaction of the leaf node components with the physical processes in the presence of ICS intrusion.

1.3 Secure Enclaves Updates

LNSAs are system -level solutions for ICS leaf node security. LNSAs incorporate intrusion detection and plant stability methods during such intrusions and also leverage isolated and trusted hardware - a *secure enclave* for the overall ICS security. LNSAs ensure physical process integrity even if an attacker gains access to the production controller. Secure enclaves enforce mechanisms to detect an attack on an ICS, then respond to the attack independently without any help from the supervisory network to recover from the attack.

Figure 1.3 depicts the TAIGA secure enclave that monitors the unsafe production controller's

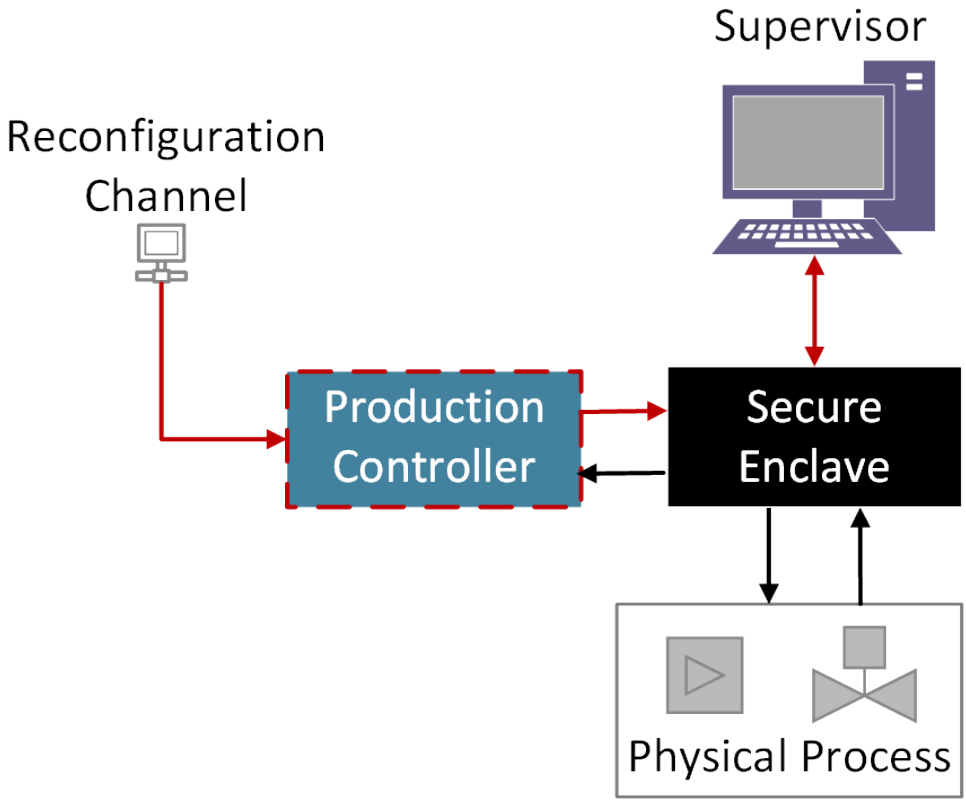

Figure 1.3: Leaf node security architecture : TAIGA

communication with the physical process and supervisory network. The secure enclave implements methods to preemptively identify the plant's unsafe behavior in the future. TAIGA brings the plant to a safe and stable state in case of such unsafe plant conditions. Similarly the S3A and C^2 secure enclave implement their own methods to identify intrusion and maintain a stable plant operation before any such intrusion jeopardizes plant integrity.

The average lifespan of ICS leaf node control components is several decades. During this period, the production controller undergoes optimization or performance updates through a dedicated reconfiguration network shown in Figure 1.3. The secure enclave may also require updates, although with less frequency than the production controller does. Secure enclave components store different minimum and maximum values of the system parameters such as

constants used by the plant model, specification guards, and the controller logic. Only range - limited updates to only these system parameters are allowed. These updates facilitates control algorithm flexibility process tuning to cope with issues like process component aging. Control components may be difficult to access. Physical access increases the opportunity for accidents that may harm operators and may make the system vulnerable to physical attacks. Sometimes ICS leaf nodes are distributed across thousands of square kilometers. Updating each node by physical access may involves expensive travel and increased downtime. In industries where the allowable per year downtime is a few minutes, physical updates increase downtime and leads to production loss. Hence, secure remote access is required to provide updates to secure enclave trusted components.

1.4 Contributions

The proposed update transmission protocol ensures integrity and confidentiality of updates. Mechanisms to deal with accidental packet loss and transmission delay or malicious replay attacks and bit manipulation attack are implemented. A dedicated secure enclave update processor(SEUP) at the secure enclave handles all the supervisory traffic during the updates and hosts cryptographic services required for authentication of updates. Encrypted storage of the updates and decryption key in the non volatile memory enhances overall system security. This solution is integrated with TAIGA implementation on a programmable System-on-Chip (SoC) platform. No interaction between the introduced SEUP and TAIGA during normal plant operation assures negligible performance overhead. Failure in the authentication of received updates immediately discards the updates and the update session is restarted.

1.5 Thesis Organization

Chapter 2 provides background information regarding ICS infrastructure and topologies. ICS security vulnerabilities and process controller security architectures that address vulnerabilities are also discussed. Chapter 3 defines a secure enclave, and addresses several questions: Why are updates required for the secure enclave? Why are remote updates preferred over physical access? What are the attributes of remote updates? Chapter 4 and 5 propose update transmission protocol and architecture at the secure enclave for update validation. Details of TAIGA are discussed in Chapter 6. Chapters 7 and 8 discuss implementation of the proposed mechanism on a reconfigurable SoC, with results and conclusions.

Chapter 2

Background

ICSs are mainly categorized in two types : Supervisory Control and Data Acquisition (SCADA) systems and Distributed Control Systems (DCS).

SCADA [13] systems are typically used in industries where process control components such as Programmable Logic Controllers (PLCs) are distributed across thousands of square kilometers. A few examples of these industries are water distribution systems, petroleum product refining and distribution pipelines, and telephone systems. SCADA provides high-level monitoring, analysis, and control of the remote process. The control components are installed at the remote stations, which send physical process information to the SCADA control center. In turn, the control center sends supervisory commands to the control components to monitor local processes. Supervisory commands are either machine generated or manually entered.

DCSs [13] are an automated control system that incorporate a central supervisory controller and distributed autonomous control elements. They integrate several control loops to provide overall control for complex interdependent processes. Typically, DCSs are used in continuous

and batch production process like petroleum refineries and complex manufacturing assembly lines.

2.1 ICS Hierarchical Topology

ICSs have strict operating requirements such as real time control and monitoring of physical processes, co-ordination between complex system components, and maintenance of physical process stability. To address this, an ICS incorporates a layered network architecture with various types of subsystems and protocols [13].

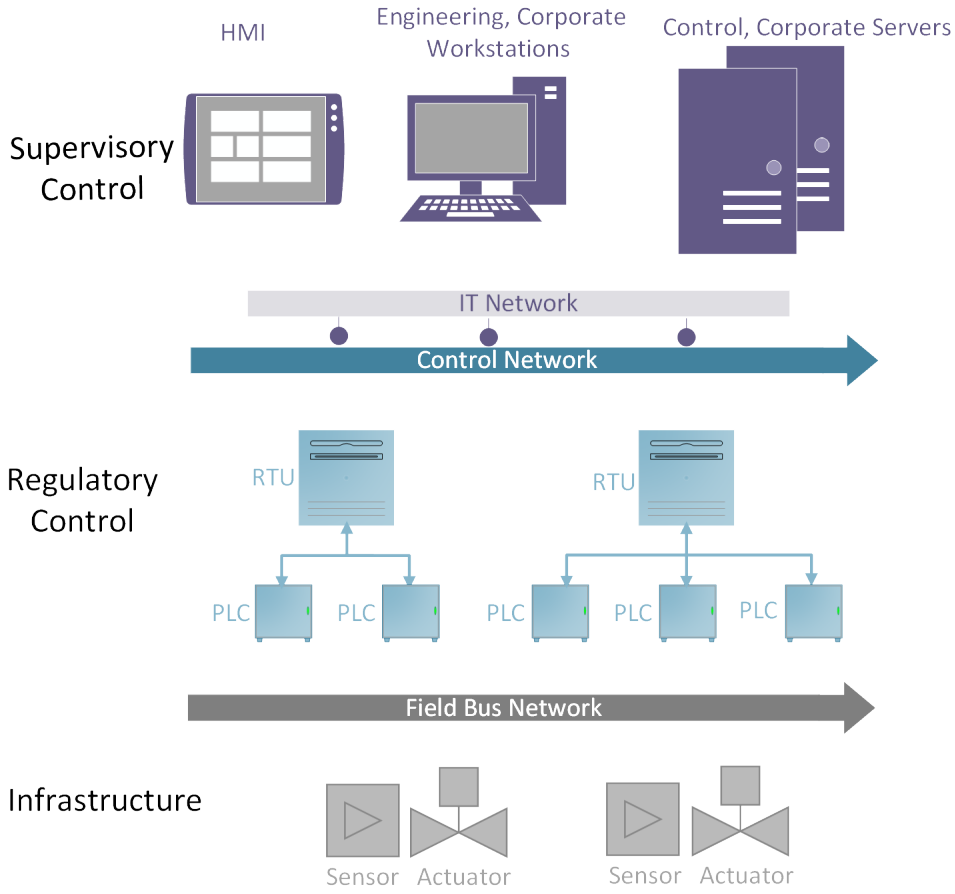

Figure 2.1: ICS Hierarchical Topology

The *supervisory layer* sends out operator-generated manual or automated control commands to regulatory controllers and monitors the systems with a Human Machine Interface (HMI). The HMI is hardware and software that displays real time process parameters and information to plant personnel at the supervisory layer. The HMI provides an interface to monitor and modify any process parameters. In case of an emergencies the operator-generated commands can override automatic control commands .

The *regulatory control layer* components - Remote Terminal Units (RTUs) and Programmable Logic Controllers (PLCs) are customized embedded platforms. RTUs are field devices that acquire physical process data and communicate via wired or wireless with the supervisory layer. PLCs are digital computers ruggedized to function reliably in harsh manufacturing environments. They implement physical process controller logic, interact with the sensors and actuators, and apply the desired control actions.

The *infrastructure layer* contains physical processes components (e.g., sensors, actuators) and required infrastructure.

2.2 ICS Security Vulnerabilities

Traditional ICSs used to be isolated systems which implemented proprietary control and communication protocols. In contrast, the IT infrastructure incorporates widely available network communication protocols, real time operating systems and heterogeneous IT components. As IT infrastructure integrates with ICSs, IT vulnerabilities are carried over to the ICS environment. Any intrusion into the IT network may give an adversary access to the control components regulating physical processes. During network integrity attacks, adversaries aim to damage physical processes before the attack is noticed and shut down. Such network integrity attacks are categorized with the help of three dimensional attack space

(figure 2.2).

2.2.1 Network Integrity Attacks

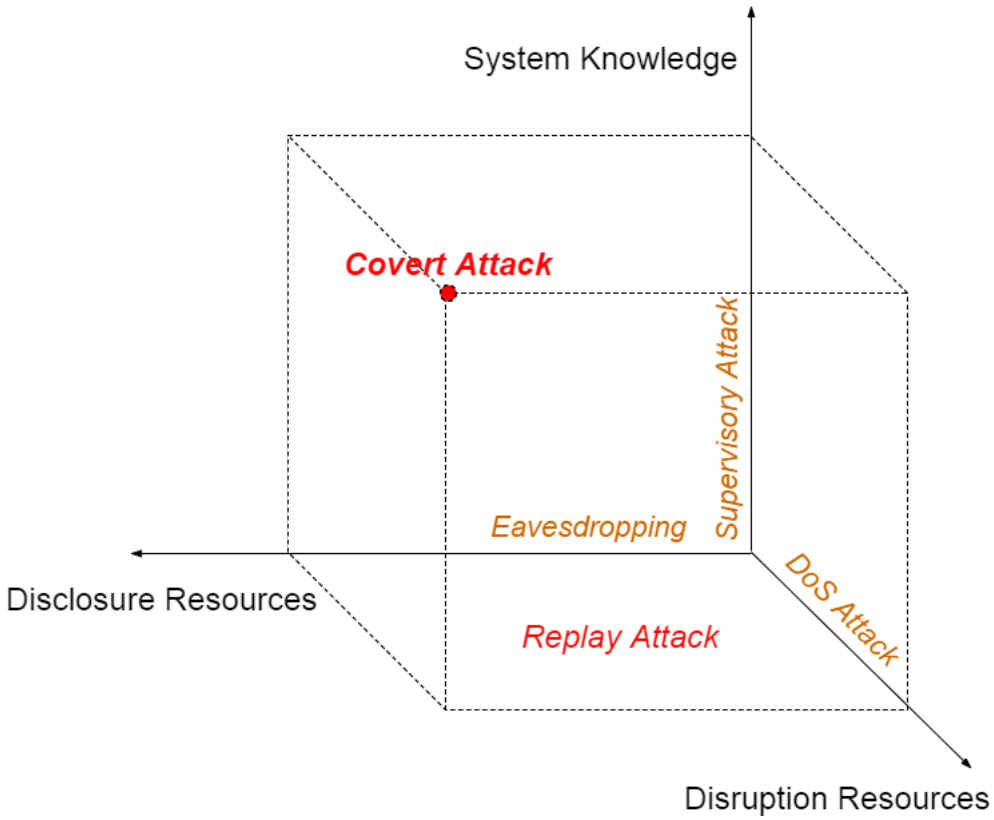

Figure 2.2: Network Integrity Attack Space
[14]

System knowledge about the behavior of the physical processes and controller enhances the stealthiness of an attack. With knowledge of the system, an adversary can design complex attacks such as bias injection attacks or zero dynamics attacks [14]. Such attacks constantly inject bias in the system to disrupt it without getting noticed by detection mechanisms. If such detection mechanisms contain any vulnerabilities, system knowledge helps the adversary to exploit vulnerabilities, bypassing them and jeopardizing physical process integrity [15].

Disruption resources allow physical attacks on a system or the manipulation of sensor and actuator signals of the controller. An adversary may execute a Denial of Service (DoS) attack using disruption resources [14] [16]. In a DoS attack, the adversary interferes with or blocks communication between the supervisory and control layers. After getting access to regulatory control layer components, the adversary may disrupt physical processes.

Disclosure resources enable adversaries to eavesdrop on sensor and actuator signals, violate data confidentiality, and obtain sensitive information. Once the adversary receives sensitive information such as telemetry between regulatory and supervisory control layers, replay attacks are possible [14]. In a replay attack, the adversary sends stale data to the supervisory network to hide any intrusion into the system and carries out the disruptive attack on the plant. Off-line analysis of obtained sensitive information using machine learning techniques may disclose plant control and response commands.

2.2.2 Reconfiguration Attacks

A supervisory reconfiguration channel provides firmware and performance optimization updates to regulatory control layer embedded components. Once an adversary get access to the reconfiguration channel, they may use that access for malicious reconfiguration. The malicious reconfiguration may nullify security measures placed at the ICS leaf node components. As regulatory layer components directly interact with the physical layer, such attacks jeopardize the integrity of physical processes. This makes reconfiguration attacks much more critical than network integrity attacks [2].

Traditional IT infrastructure security mainly deals with the protection of information and security at the network nodes. Protection of information is carried out using anti-virus softwares to detect and remove viruses and their footprints. Firewalls provide security at the

nodes by thwarting any intrusion from outsiders. An individual responsible for IT security is always busy addressing zero-day exploits. Such an IT security setup is not helpful in addressing ICS security. An attack on IT systems may lead to destruction of information or denial of access to sensitive information. An attack on ICS may give access to the control components and lead to damage of the physical infrastructure. Still most of the ICS security measurements are based on a similar model.

2.3 ICS Security Architectures

ICS leaf node components directly interact with the physical world and are susceptible to network integrity attacks and reconfiguration attacks. Any intrusion into leaf node components jeopardizes the integrity of the physical process and may lead to damage of plant infrastructure. Mechanisms to maintain physical process integrity in presence of such attacks are required. LNSAs apply the idea of *using simplicity to control complexity* [17] to protect ICSs.

2.3.1 Production Controller

The production controller is the main controller, driving control operations of the plant. It is a high performance controller and uses advanced complex control strategies to meet production requirements. The production controller requires optimization and control logic updates that makes it susceptible to reconfiguration attacks.

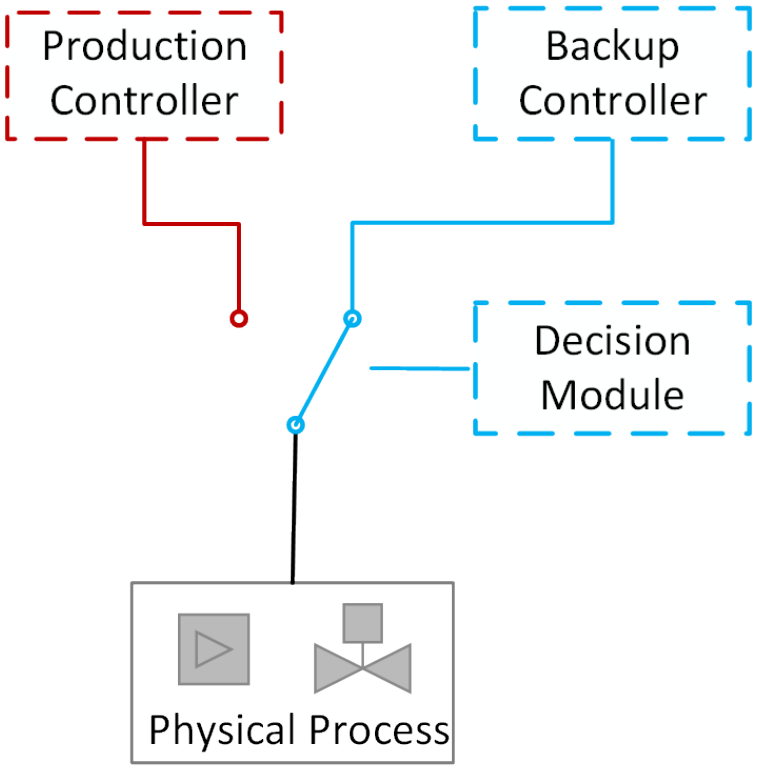

Figure 2.3: Secure Enclave Modules

2.3.2 Decision Module

The decision module implements mechanisms to detect any intrusion into the ICS leaf node that may jeopardize plant safety. When the decision module detects an intrusion, it immediately switches control of the plant from the production controller to the backup controller to maintain safe and stable operation of the plant. The main aim of the decision module is to accurately identify unsafe process behavior before it damages plant infrastructure. It is important that the decision module be able to identify any noise in the system and that it does not invoke the backup controller at the slightest disturbance sinc attackers may take advantage of overly sensitive decision module to affect plant production.

2.3.3 Backup Controller

The backup controller is a high assurance controller, rigorously verified to maintain process safety and stability. The backup controller does not undergo any updates and it is verified for any malware before installation.

Switching from the production controller to the backup controller is bumpless transfer, maintaining plant stability.

2.4 Secure Enclave : Decision Methodologies

TAIGA [10]

TAIGA ensures that control components follow defined trust requirements and enforced process stability specifications to identify unsafe production controller commands and malicious sensor measurements. The decision module derives a state vector representation of the plant's current operation and predicts the plant's future behavior using state estimation. The decision module preemptively detects any violation of the process stability specifications and initiates switch over from the production to backup controller.

S3A [11]

S3A decision module maintains a copy of high-level system control flow and execution time profiles for the production controller logic. It continuously monitors the execution time of the production controller and physical state of the plant for any deviations from the desired execution time and high-level system control flow. If any such deviation gets detected, the decision module switches control of the process from the production to the backup controller. S3A is able to detect any intrusion into the system in less than 6μS. This time scale is difficult

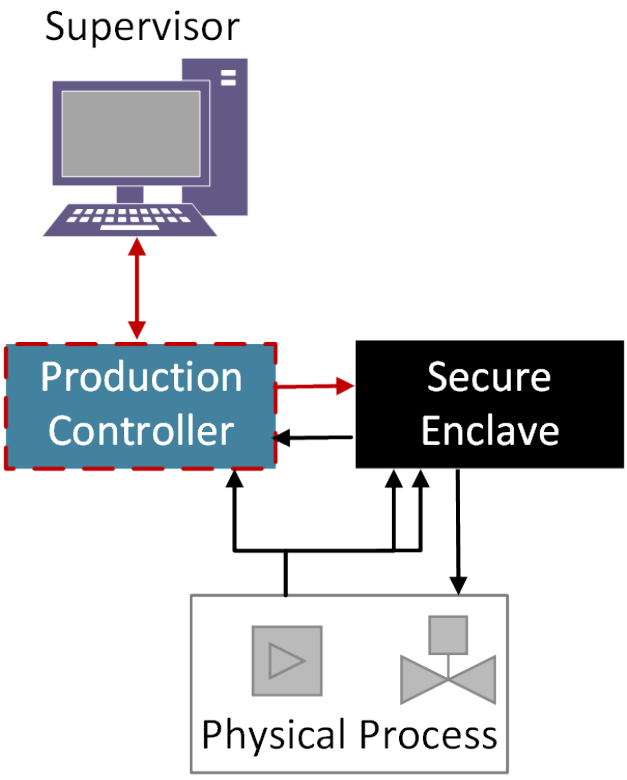

Figure 2.4: Leaf Node Security Architecture : S3A

for an adversary to defeat.

C^2 Architecture [12]

C^2 enforces engineered supply safety properties while the production controller is running. The C^2 sits between the production controller and the physical process. At the end of each control cycle C^2 checks the production controller control signals against safety properties. Any violation of safety properties results in a denial of control signal and notification to the production controller. C^2 detects any violations just before they occur. A simple denial of the signal may affect physical process stability. To avoid this C^2 follows corrective action steps to maintain safe and stable operation.

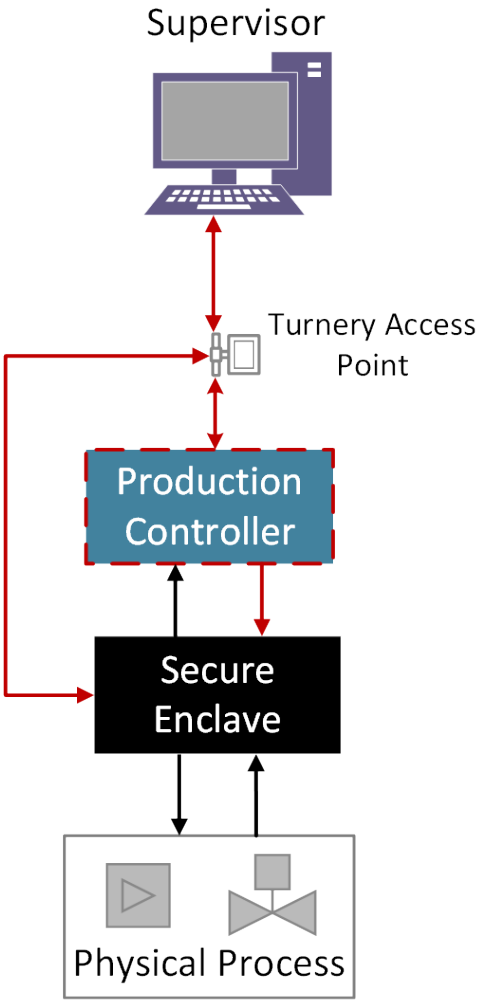

Figure 2.5: Leaf Node Security Architecture: C^2

Chapter 3

Secure Enclaves and Updates

The secure enclave is a trusted hardware unit that incorporates multiple modules such as a decision, a backup controller etc (Figure 3.1).

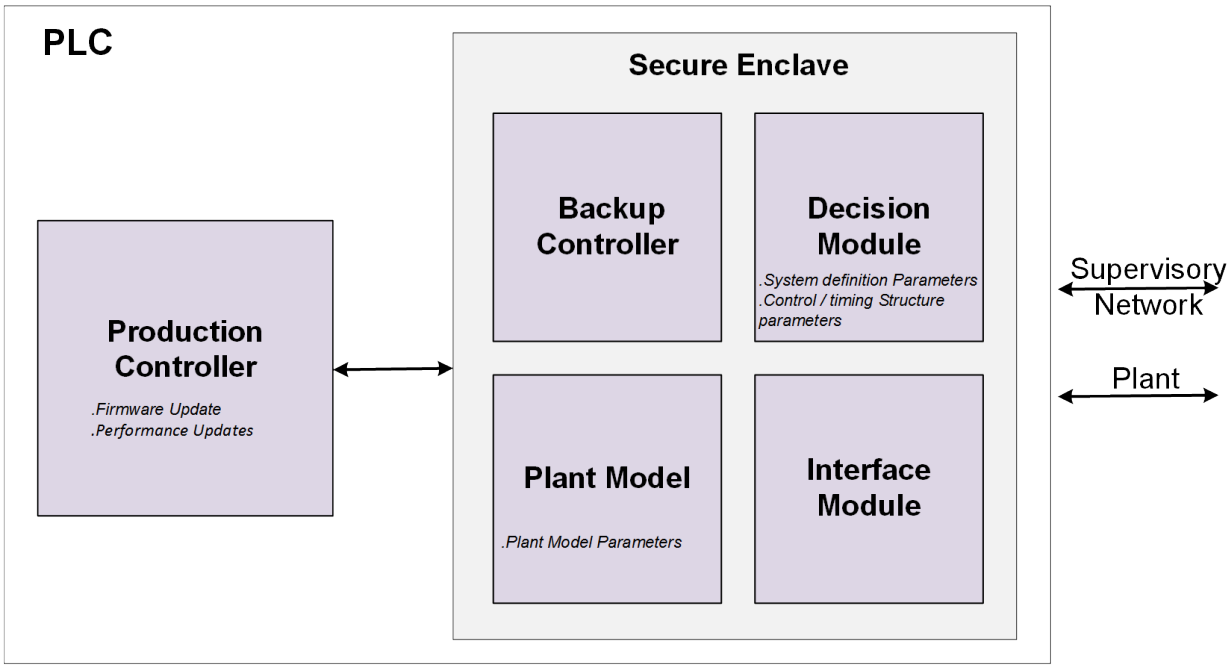

Figure 3.1: Secure Enclave Modules

Formally verified logic code, physical isolation from the untrusted production controller,

and trusted and isolated submodules create a root of trust in ICS leaf nodes. The decision modules implement mechanisms to detect intrusion in the system before it jeopardizes the integrity of the physical process. Intrusion detection mechanisms are based on the plant model, safe operational specifications of the physical proces, and control algorithms of the production controller. The high assurance backup controller contains a simpler version of the production controller algorithm.

3.1 Leaf Node Components : Software Updates

The process control loop components (controller, sensor, actuator, process components) of the plant are ruggedized to function reliably under environments such as extreme temperatures and high vibrations. Once installed at the location, components are expected to function for several decades without needing hardware upgrades.

The control and computation components are customized embedded platforms that run control algorithms and leaf node security mechanisms. During such a long period, if required, software updates to the production controller are provided through the supervisory reconfiguration network. These updates include control algorithm performance improvement or optimization updates, firmware updates, and updates for enabling new functionalities as per business requirements.

Component Aging

An ICS process control loop typically uses various electro-mechanical components. Aging of the component degrades performance characteristics of the component and process control effectiveness. As the component characteristics change, the transfer function of the system alters which leads to changes in the controlled parameters [18]. The transfer function is a

mathematical representation of the input-output of the control system. Depending on the function of the aging component in a process control loop, changes in controlled parameters can adversely affect plant production or physical process stability. Limited allowable downtime of ICS prevents replacement of aging components. Finding exact replacement components is difficult as components may not be available or may no longer be manufactured (manufacturers may cease to exist 10 to 15 years after the installation).

The secure enclave implements the backup controller algorithms and security mechanisms based on the plant model and system definition (Figure 3.1). To cope with component aging or to allow flexible control of the process, associated parameters at the secure enclave also require appropriate modification. For the S3A secure enclave, modifications to the control and timing structure of the code may be required. Unlike the production controller, unrestricted updates to the secure enclave are not allowed. Parameter updates must not be allowed to modify safety properties or system guards. Such updates could nullify the protections provided by the secure enclave and jeopardize plant integrity.

3.2 Remote vs Physical

In many industries with an annual uptime of 99.99%, only 5 minutes per year are allowed for updates [19]. Sometimes ICSs may requires updates at geographically distributed locations and each location has multiple leaf nodes. A physical update process involves expensive travel, increases downtime, and leads to a production loss.

SCADA system such as a water distribution are geographically distributed over thousands of kilometers as shown in figure 3.2. The water reservoir facility, a water treatment plant, water storage facility and water distribution pipelines are different stages in the water distribution process. Each stage has multiple leaf nodes networked to provide remote control

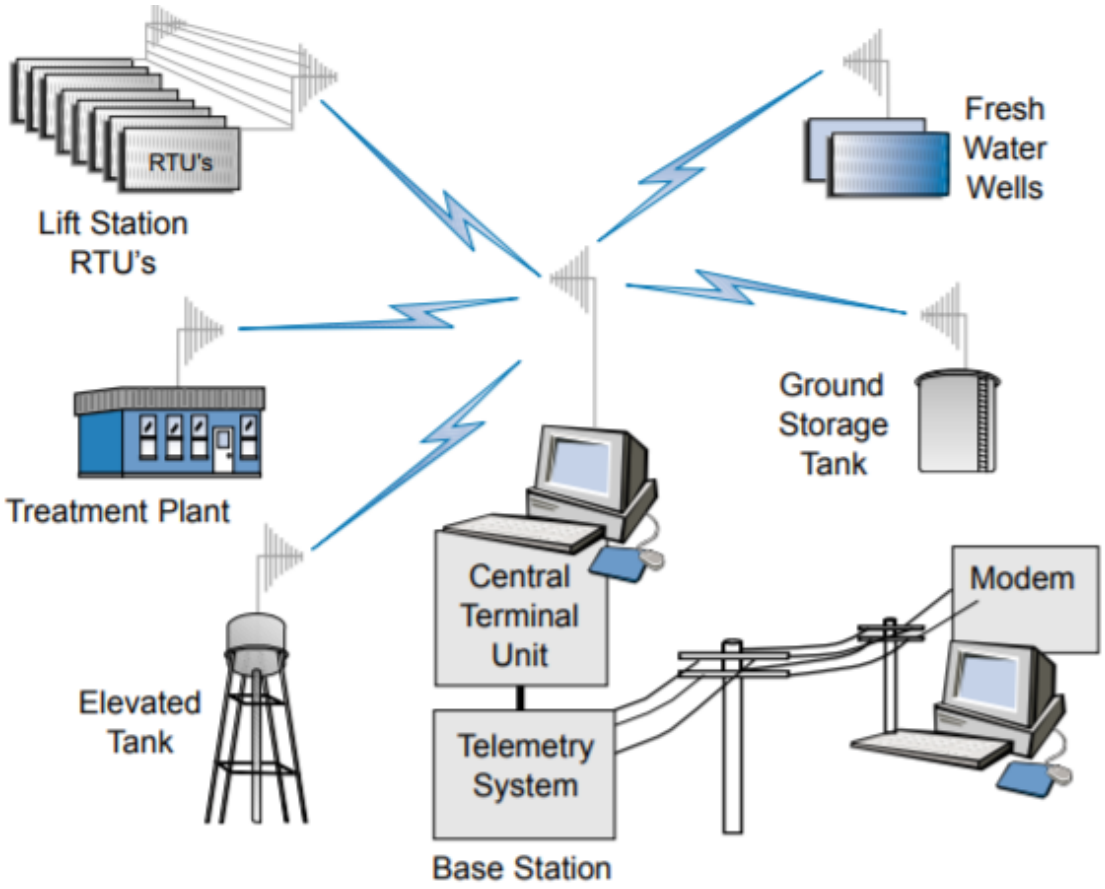

Figure 3.2: Water Distribution SCADA System [1]

and monitoring for facilities. The physical update process for leaf node components involves expensive travel to each remote station. Leaf node components may be installed inside moving entities or at heights which makes access difficult. The physical update process results in leaf node components that may be vulnerable to physical attacks from disgruntled employees with malicious intent or could result in unintended accidents costing human lives or damage to the infrastructure.

3.3 Remote Updates and Security

Researchers of the proposed LNSAs discarded an idea of updates by remote fashion due to associated security implications . The secure enclave communicates with operators and supervisory layer devices through a supervisory network. Supervisory layer receives process information and parameters from the secure enclave and sends back control commands. Remote updates to the secure enclave are carried out through supervisory communication channel. Updates to the secure enclave through the supervisory network makes the enclave susceptible to reconfiguration attacks. As previously discussed, an adversary with a successful network integrity attack may carry out malicious reconfiguration and destroy trust provided by the secure enclave. Broken security mechanisms of LNSAs jeopardizes plant integrity. An adversary may gain knowledge about the safety specifications and security mechanisms implemented at the secure enclave. Such information helps adversaries to create stealthy attacks.

However, we argue that considering high availability, safety and stability requirements from the ICSs, remote updates are essential.

3.4 Attributes of Remote Updates

Considering the security implications of the remote updates, hardware and software security mechanisms are required to transfer and configure updates. This section discusses rules that need to be enforced for the entire update session. The ICS operations are divided into two sessions. During the production session, the plant is in normal operating conditions. Updates are carried out during the update session. Plant personnel ensure that production controllers are not running and all the physical processes are stopped.

Confidentiality and integrity

Updates travel through the supervisory network. Network integrity attacks on the supervisory network expose sensitive information. Encryption protects the confidentiality of sensitive information and prevents any attempts to eavesdrop and gain system knowledge. An authentication of the updates at the secure enclave ensures integrity of the update and resilience against bit manipulation attacks. Confidentiality must be maintained when the updates are stored at the secure enclave.

Restricted Access

At the secure enclave, access to sensitive information such as the backup controller algorithm or decision module logic should be denied. Otherwise a successful network integrity attack may break the security mechanisms of the secure enclave.

Invisibility during the production session

ICS control loops have high performance requirements. The update mechanism should not incur any performance overhead. Interaction between the secure enclave and the update mechanism must be restricted to only the update session. Any interaction during the production session opens a back door for an adversary to the secure enclave resources.

Policy for unforeseen circumstances

If attack is detected, the update mechanism must implement rules to follow, such as roll back mechanism to configure updates of the previous version. Coordination between the plant organization, vendor and plant personnel is important during the updates and at the beginning of the production session. Guidelines to follow during such events are similar to the patch management process guidelines provided by the Industrial Control System Cyber Security Response Team (ICS CERT) [19].

3.5 Case Studies

Trusted and authorized remote access to the secure enclave is required to ensure plant integrity before and after the updates. Remote access technology already exists for computing devices used in enterprise IT infrastructure. Apple cellular phones implement a mechanism to authenticate accesses even if its operating system is compromised.

3.5.1 Intel Active Management Technology

Intel vPro technology [20] is a set of resources embedded into the hardware of Intel processor. Intel Active Management Technology (AMT) is a part of the VPro that allows an administrator remote access to the device for debugging, maintenance and update configuration.

Organizations use IT infrastructure to automate their business processes. This improves operational efficiency and cost. IT infrastructure incorporates devices such as work stations, data servers, and networking devices. Remote diagnosis, maintenance, and updates of the device helps maintenance personnel to work efficiently without traveling to a device location. AMT is a hardware + software solution that enables remote access to Intel devices. It works even if the device is powered off, or if the operating system is unresponsive, or if a hard drive has crashed, as long as the device is powered up and connected to a network.

AMT is independent of the main processor system and facilitates remote operations such as power control of the device, system installations, etc. A micro-controller based dedicated subsystem called a Management Engine (ME) is placed at the application processor's memory and graphics controller. Flash memory stores ME firmware, device BIOS, and cryptographic keys. The multiple masters - ME, application processor, and LAN are allowed to access flash memory although the access is read only. ME firmware protection is extremely important

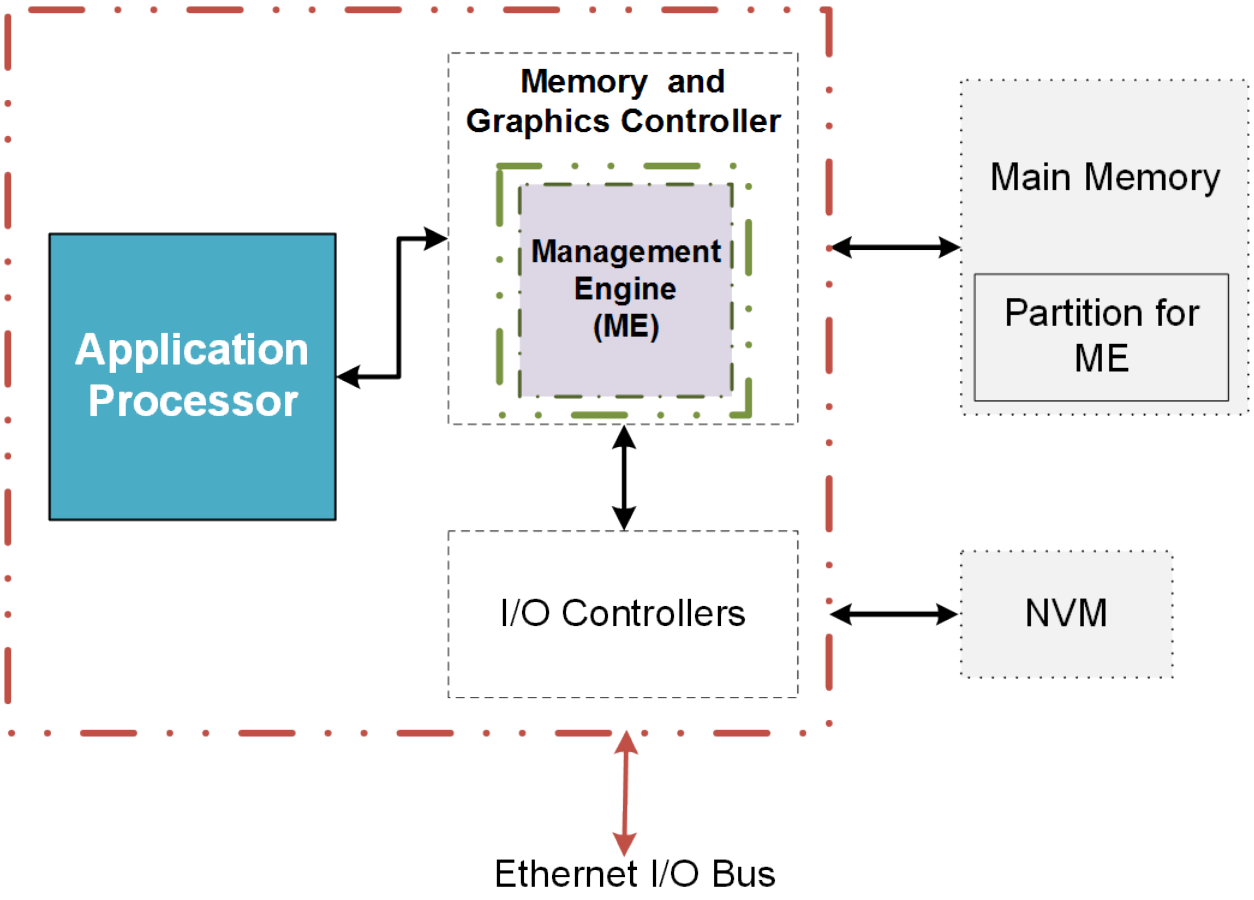

Figure 3.3: Intel AMT

since an unauthorized firmware image running on ME will allow direct access to system resources and jeopardize the security of IT infrastructure services. During the development phase, Intel generates a public and private firmware signing key pair and encrypt firmware image with the private key. The public key and the firmware image is stored on the flash and the private key is stored securely at an Intel location. To protect the public key from any unauthorized changes, a hash value of the public key is stored at the ME boot Read Only Memory (ROM). During secure boot of the system, the public key is first verified using boot ROM; after successful verification of the key, firmware at the flash is validated, followed by the typical boot process from flash. As the application processor runs, AMT

allows bidirectional asynchronous communication with the ME as well as with the LAN.

3.5.2 Apple Secure Enclave

[21] Apple secure enclave processor (SEP) is a part of an Apple device's hardware and coexists with the main application processor. The SEP implements all cryptographic functions and ensures trusted operation even if the application processor is compromised.

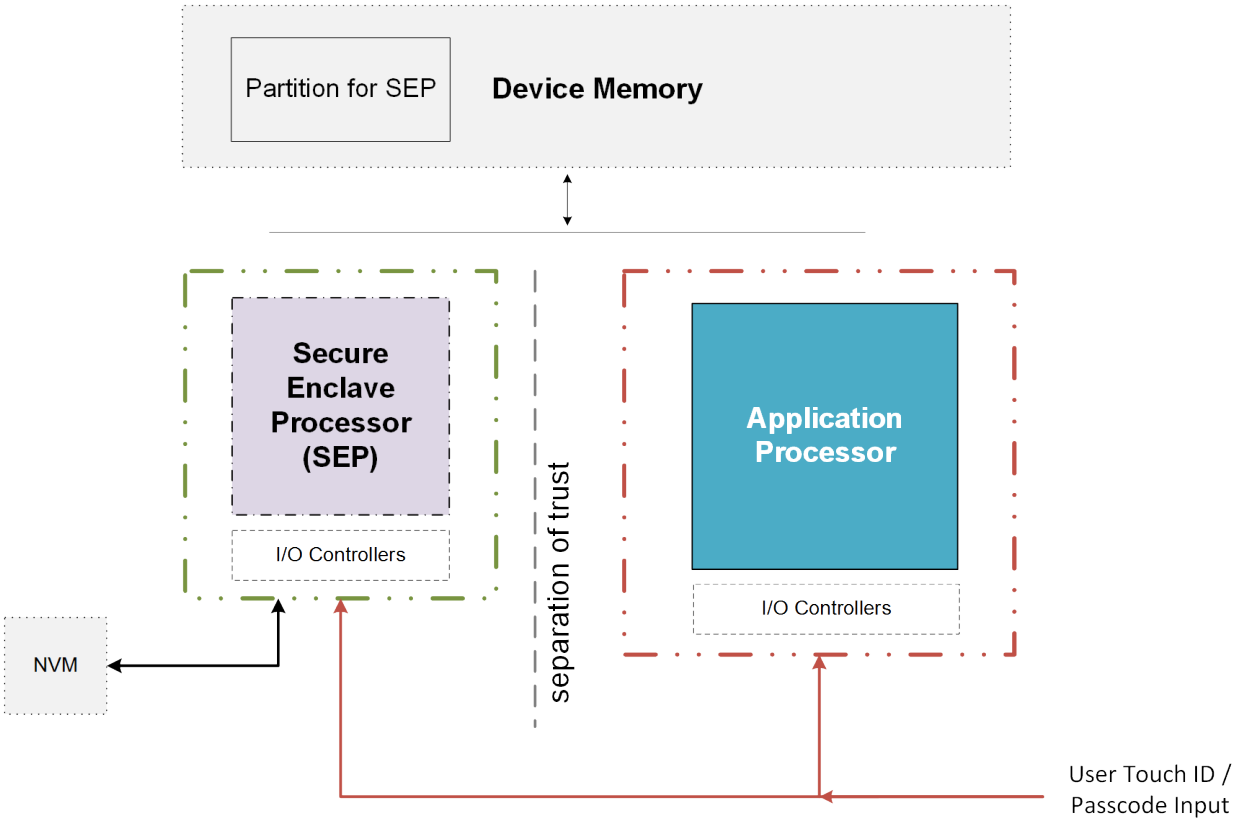

Figure 3.4: Apple Secure Enclave Processor

The SEP is a coprocessor present in Apple A-series processors its own kernel, cryptographic services, key management, and encrypted memory partition.

A Unique Identification Number (UID), generated during fabrication of the device, is accessible only to the SEP and is unknown to Apple or the application processor. An ephemeral

key generated at the start of the boot process is combined with the UID to encrypt the SEP portion of the device memory. This restricts access from any other part of the device to SEP data. The SEP receives fingerprint data from the touch sensor and determine its match with registered fingerprint data. A scanned fingerprint print data is forwarded by the application processor to the SEP. The finger print data is encrypted using AES-CCM, an authenticated encryption mechanism to prevent eavesdropping by the application processor. If matched, the SEP grants the user access to the device and notifies the application processor. Having a dedicated processor for all cryptographic services, an encrypted memory partition, and encrypted data transfer between user and the SEP ensures safe and trusted device operations.

Embedded platforms such as personal computers and handheld mobile devices frequently undergo remote updates. Typically such devices implement widely available operating systems and a secure socket layer (SSL) stack for secure remote communication. SSL runs on top of the TCP or the UDP layer and provides data confidentiality and integrity during remote access. A firewall at the remote embedded device thwarts unauthorized accesses. Full-featured operating systems supporting SSL and firewalls are generally unavailable in embedded process controllers. PLCs are resource-constrained devices designed to support the intended control algorithm and may not have enough memory and computing resources for these security measures. Hence, we require resource-efficient solutions that are customized for the PLC platforms.

Chapter 4

TAIGA

4.1 TAIGA Overview

The TAIGA secure enclave serve as the last line of defense to protect integrity of physical processes. The decision module acts as a intermediary for controller input and output. It creates a physical layer of trust between the controller and the physical process [10].

4.1.1 Production Controller

The production controller is a closed loop feedback controller. It receives sensor outputs and operational set points from the supervisory layer. The production controller has a high performance requirements and controls multiple processes. A dedicated Ethernet connection for network updates makes it susceptible to reconfiguration attacks. It is consider as an untrusted component in the TAIGA framework.

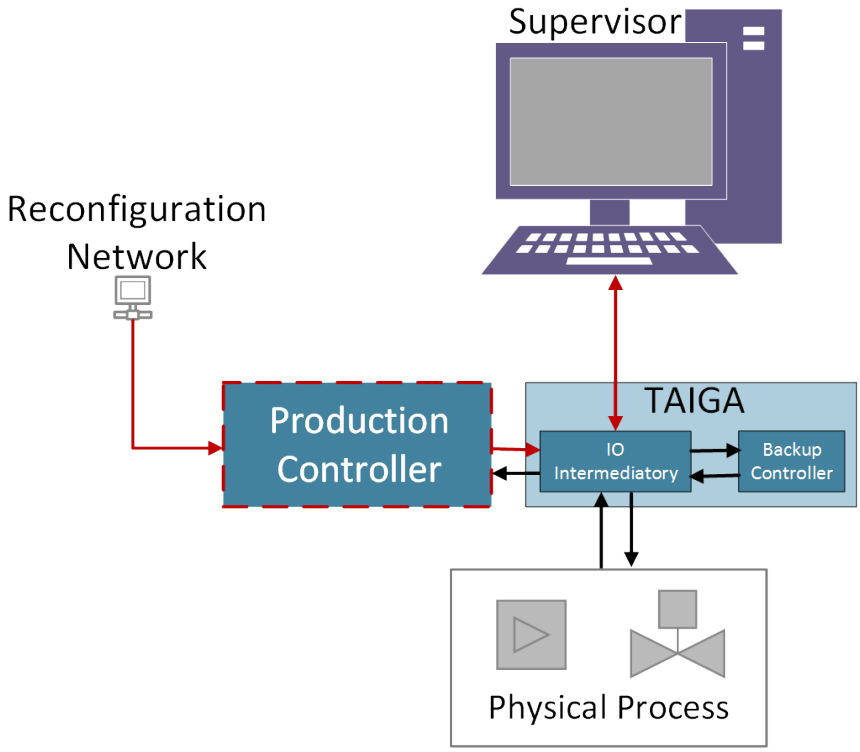

Figure 4.1: Leaf node and TAIGA

4.1.2 IOI Module

The IOI module monitors the communication between the production controller and the supervisory network. This allows IOI module to detect an attack from the supervisory network. The IOI module includes a trigger mechanism, which monitors the plant behavior based on the sensor outputs. TAIGA monitors the plant control I/O bus and detects attacks originated from the infrastructure layer.

Operational Guards

Operational guards are simple hard bounds on operating limits for the sensor measurements and control signals. These guards are enforced to meet the standard process specification.

Safety Guards

Safety guards protects plant from safety hazards and ensures plant stability.

Trigger Mechanism

During normal plant operation, the production controller controls the physical process. The IOI module continuously monitors telemetry between the production controller and the physical process. The trigger mechanism checks for any violation of specified guards. If any violation is detected, the switchover logic transfers control of the physical process from the production controller to the trusted backup controller and ensures plant stability. The backup controller is not susceptible to any reconfiguration attack.

4.1.3 Backup Controller

The backup controller is a high assurance controller, verified for correct operation. Even if the production controller fails or is compromised, the backup controller ensures stable plant operations. The production controller is not accessible through the network and does not share any resources with the production controller. The backup controller ensures stable plant operation during unsafe supervisory commands. During normal plant operation output of the backup controller is dropped and output of the production controller is forwarded to the plant. Before switching from the production to the backup controller, the IOI module sends current state vector to the backup controller to ensure a bumpless transfer.

4.1.4 Isolation of Trust

TAIGA components meet the following thrust requirements [10].

TR1 The source code and implementation for the entire component are analyzed.

TR2 The component uses private hardware resources for computation, internal communi-

cation, and memory, and does not invoke external components as sub-functions.

TR3 All external communication with untrusted components is through hardware-implemented, bounded, and isolated queues.

TR4 The component cannot be bypassed or disabled, and has a fixed repertoire of essential services, such as I/O or cryptography.

TR5 Critical functionalities of the component, such as rule checking logic, cannot be updated without provably secure or physical access.

These trust requirements add resilience to the ICS leaf node. TAIGA establishes a level of trust similar to a Trusted Platform Model(TPM).

4.2 Implementation : System Overview

Figure 4.2 presents a system that serves as a mock ICS. The system implements regulatory controllers on the ZedBoard platform. The ZYBO board (PLC) interacts with the supervisory network through a Raspberry Pi (RPi) that resembles an RTU.

4.2.1 PLC : ZedBoard

TAIGA is implemented on a ZedBoard [3] which contains a Zynq-7010 SoC [4], as shown in Figure 4.3. The Zynq-7000 configurable SoC is partitioned into two subsystems: a PS and a PL. The PS contains a dual-core ARM processor while the PL contains an FPGA fabric. The Zynq SoC maintains a separation of resources, memory, and peripherals between the PL and PS.

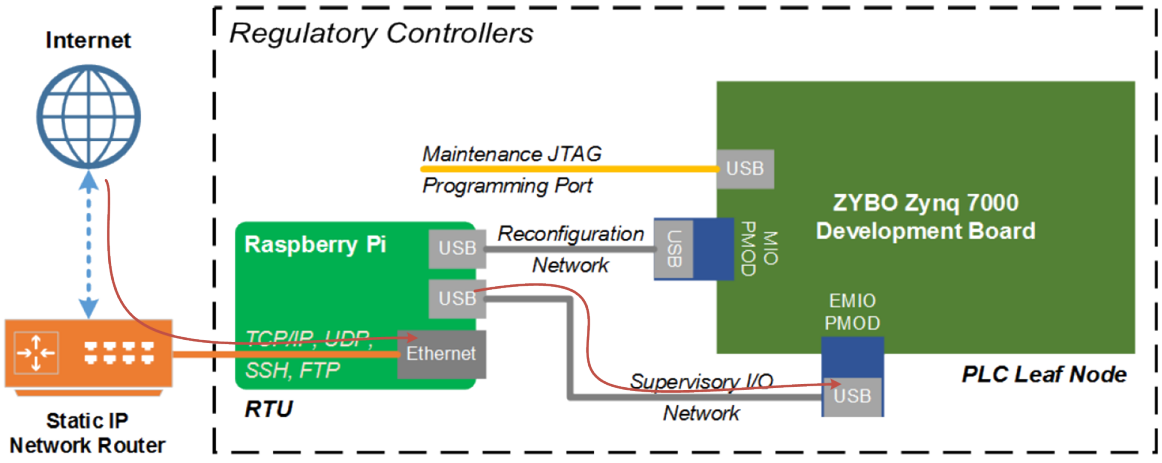

Figure 4.2: System Overview and Update Path in Red [2]

Processing System

The PS contains two ARM cores, each with a distinct floating point unit (FPU), memory management unit (MMU), and L1 caches (Figure 4.4) for data and instructions. Both ARM cores shares the L2 cache, on-chip SRAM, internal peripherals such as timers and interrupt controllers, and external peripherals such as off-chip memory, flash, and I/O.

The ZedBoard supports two non volatile memories that retains data across power cycles: SD card and external flash memory. The SD card slot is accessible through the SD peripheral controller in the PS. External flash is accessible to both the PS and PL.

Programmable Logic

The PL is an FPGA fabric, consisting of a large set of configurable logic blocks (CLBs), block RAM (BRAM), digital signal processing (DSP) slices, I/O blocks (IOBs), and registers (Figure 4.4). CLBs are groupings of lookup tables (LUTs) configured for combinational logic requirements, and flip-flops (FFs) for sequential logic. Typically the PL FPGA fabric

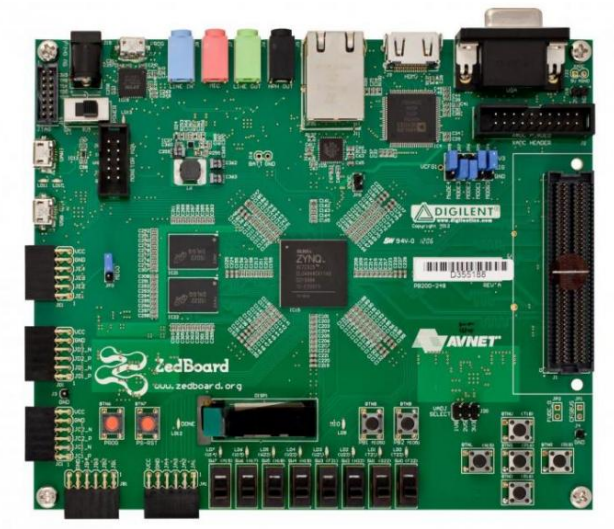

Figure 4.3: Xilinx Zed Board [3]

is configured using a hardware description language (HDL) such as VHDL or Verilog. The Vivado design flow enables FPGA configuration using existing high-level functional blocks.

MicroBlaze Processor

MicroBlaze is a softcore processor implemented in the FPGA fabric. The MicroBlaze has direct BRAM read/write access. All processor registers are local to the MicroBlaze and addressed within the PL. The IOI and the backup controller are implemented on independent MicroBlaze cores. TAIGA uses a 145 MHz clock.

4.2.2 RTU : Raspberry Pi

The PLC interacts with the regulatory controller (RTU) implemented on a RPi platform [22]. RPi interacts with the IT networking services. A UDP suits the supervisory network requirements of efficient real time monitoring and robust network protocol. A UDP web server is implemented on the Raspberry Pi. The UDP server opens one port of the router

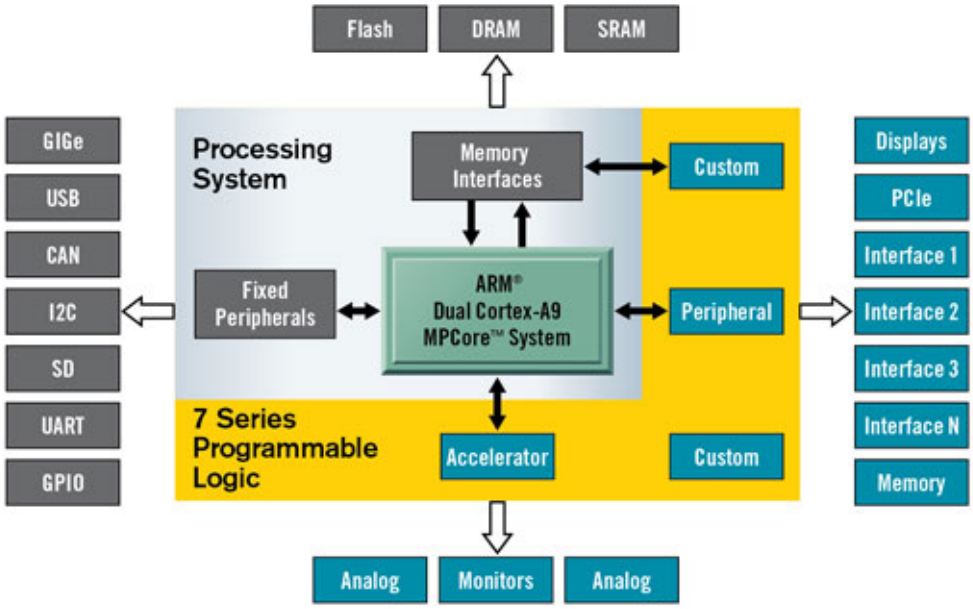

Figure 4.4: Zynq SoC [4]

connected to Raspberry Pi. After a request from remote client, Raspberry Pi transmits process data to clients and receives data request and control commands on this port. Given port at the router is forwarded to the private Raspberry Pi's IP address that grants public access to the Raspberry Pi to any client on the Internet.

4.2.3 Implementation: Internals

Figure 4.5 shows the implementation of TAIGA on a Zynq SoC. To enhance ICS security, TAIGA isolates the trusted components and restricts the intermodule communication through hardware implemented queues. TAIGA is invisible to the production controller, the physical process and the supervisory network. The production controller functions as the PLC in an ICS architecture. The production controller is implemented on the PS and communicates with regulatory layer components (sensors, actuators) for process control.

The IOI module and the backup controller are implemented as a MicroBlaze processor in the

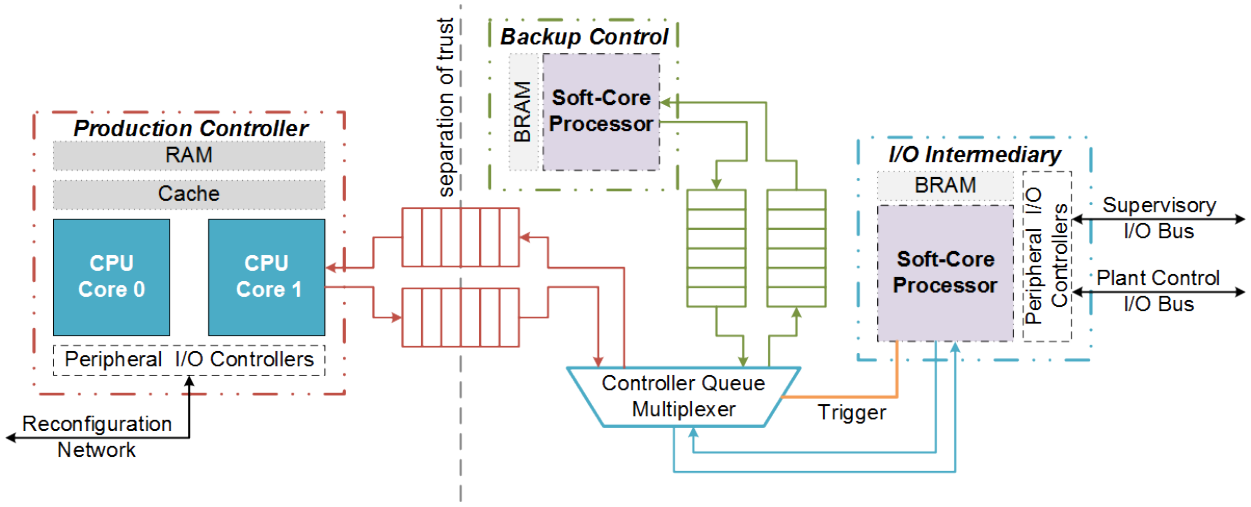

Figure 4.5: TAIGA [2]

PL. The IOI module and the production controller communicate with each other via FIFO queues and ensures commands from the supervisory network are within safety limits. The backup controller implements a simpler control algorithm than the production controller and is inaccessible through the supervisory network.

Chapter 5

Update Transfer Protocol

Previously, researchers discussed an approach to provide remote updates to the secure enclave parameters of TAIGA [10] . An update module at the TAIGA secure enclave facilitates cryptographic services to decrypt and authenticate received updates. Formally analyzed code of the module also verifies that new parameter values are acceptable. Our approach introduces a dedicated hardware module at the secure enclave to handle all operations during the update session. The proposed update transfer protocol is independent of the underlying network protocol stack and implements mechanisms to deal with packet loss, network transmission errors, or network integrity attacks.

Plant personnel send updates from a trusted workstation at the supervisory layer as shown in Figure 5.1. Updates are transfered as a sequence of blocks of defined size, referred to as *update blocks*. An update block travels through the supervisory network and reaches RTU at the remote location. In DCS or SCADA systems, PLCs are the lowest-level embedded platforms that do not directly interact with the supervisory network. An RTU is responsible for interacting with the PLCs on a control network and relaying messages on the IT network. Multiple PLCs are connected to a single RTU. The RTU forwards the update block to the

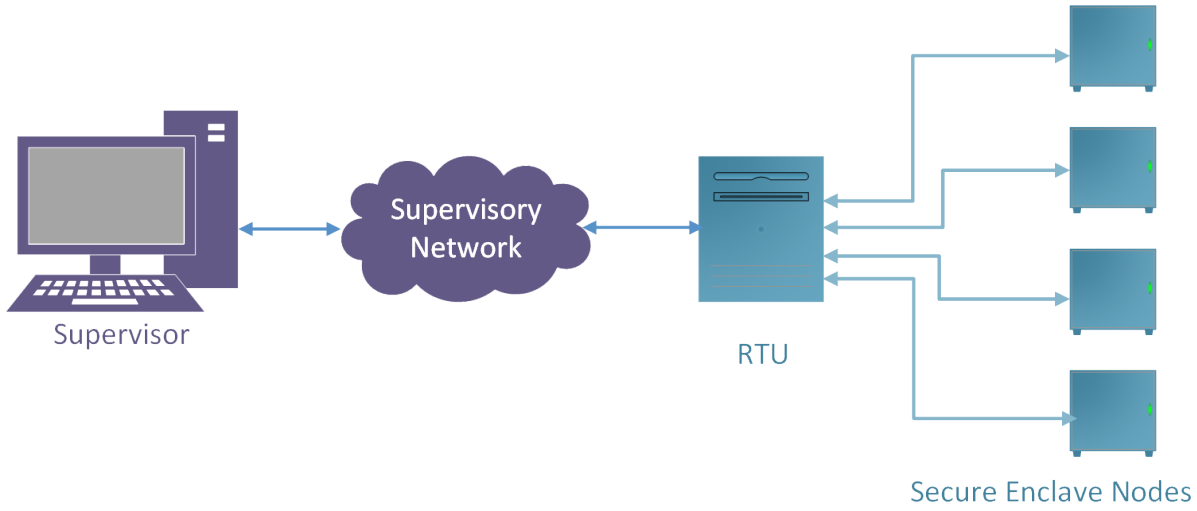

Figure 5.1: Path for Updates

PLC that hosts the destination secure enclave node. Each node has multiple application processors that implement secure enclave modules such as the backup controller and decision module. The node receives all update blocks for each application processor. The update path is highlighted in red in Figure 4.2.

5.1 Building an Update Block

An update block contains multiple update data words (1 word = 32 bits) and few handshaking words. To build the update block, we start with raw update words without any handshaking words, encryption or authentication hash value.

Table 5.1: Payload

Size	total words for one application processor
Description	payload

Step 1 : Payload + Handshaking Words

Table 5.2: Payload + Handshaking Words

2 words	1 word	total words for one application processor
control flags	start address + number of update blocks for one node	payload

Out of two control flag words, the one word is assigned for a unique ID of the secure enclave node connected to the RTU. The second word of the control flag has four byte fields.

Byte 1: Application processor ID (Decision Module 0, Backup Controller 1 and so on) at the node. Can support a maximum of 256 processors per node.

Byte 2: Version ID of the update. An adversary may replace the update block of the current version with that of a previous version of updates to conduct replay attack. The version ID prevents replay attacks.

Byte 3: Update block sequence number. The sequence number helps to prevent replay attacks where an adversary captures and retransmits an older packet from the same update version.

Byte 4: All the update blocks are of the same size. Depending on the total size of the update, the last block may require padding words to make a perfectly sized block. Byte 4 stores a number of padding words in the give update block. One update block can a hold maximum of $2^8 - 1 = 255$ words

The **start address** is a 24-bit application processor memory address where the current block will be copied. For a single application processor, 8 bits are assigned for the total number of update blocks. This restricts the total size of an update for one application processor to $2^8 - 1 * 255 * 32 = 2$ Megabits approximately. This number is much greater than the size

of parameters stored at the secure enclave. For customization purposes, assigning more bits instead of 8 bits easily increases the total size of payload.

Step 2 : Payload + Handshaking Words + Authentication

The supervisory network incorporates open networking standards and protocols for information exchange. In the open systems interconnection (OSI) model, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the most dominant transport layer protocols used on the Internet. For error-free data delivery, measures such as a checksum are built into the protocols. However they do not incorporate cryptography-based security measures for data confidentiality or integrity. Encryption of updates prevents the adversary from violating data confidentiality and obtaining sensitive information by eavesdropping on supervisory communication. Authentication of the updates detects any unauthorized changes by the adversary. The proposed protocol implements a symmetric authentication - encryption algorithm that uses the same key for encryption and decryption. A copy of the key K_{ae} is stored at the secure enclave and the supervisory workstation. It is assumed that supervisory workstation is trusted and securely stores a copy of the cryptographic key embedded in each secure enclave node.

K_{ae} is stored in the non-volatile memory of the secure enclave at the very beginning, during deployment and commissioning of the control hardware. The key is not transfered over the unsafe supervisory network. For authentication the Secure Hash Algorithm 1 (SHA1) [23] are used. However, the protocol is independent of the type of authentication and encryption algorithm. To authenticate a meaningful data instead of a encrypted data, fist authentication and then encryption of the update block is carried out.

Hash functions are mathematical computations that take an arbitrary amount of data as an input and produce a fixed amount of data as output. For the same inputs, a hash function

produces the same output. Knowing only the output, it is computationally infeasible to determine the input of a hash function. SHA1 is a hash algorithm developed by the National Security Agency. SHA 1 accepts input that is a multiple of 16 words and generates a unique hash value of 5 words.

For update blocks, IP addresses and port numbers are not encrypted and authenticated as the RTU is required to access them in order to validate the sender and forward the update block to the correct node. The remaining words of the update block require authentication and encryption. The maximum payload size for a single update block is restricted to 253 words (based on data input requirements of the encryption algorithm). SHA1 generates a hash key for 253 words payload + 3 Handshaking words = 256 words shown in Table 5.3. If the total size of the payload is less than 253 words, padding words are added to reach 253 words. At the destination node, a hash value is generated and compared with the received hash value. Matched hash values authenticate the received update block.

Table 5.3: Payload + Handshaking Words + Authentication

2 words	1 word	253 words	5 words
control flags	start address + number of update blocks for one node	payload	Hash Key

Step 3: Payload + Handshaking Words + Authentication + Encryption

AES [24] is a symmetric key data encryption algorithm. AES works on a input data size of 128 bit and variable key length. For AES-128, the encryption key length is a 128 bit. For the update block, 1 word is padded to make the total size of the update block = 262 encrypted words Table 5.5.

Additional words for first update block

Table 5.4: Payload + Handshaking Words + Authentication + Encryption

2 words	1 word	253 words	5 words
control flags	start address + number of update blocks for one node	payload	Hash Key
1 word			
Padding			

Table 5.5: First Update Block

1 word	1 word	1 word	262 Words
Supervisory workstation IP Address	Destination PLC IP Address	Port numbers	update block

The update block is encrypted and only the secure enclave node can decrypt it. An RTU requires the IP address of the PLC and the port number of the secure enclave to forward the update block to the destination node. Three words without any encryption and authentication are added at the beginning of the first update block. The IP addresses are private IP addresses assigned to the device connected to a private local network. The destination PLC has multiple ports such as reconfiguration and process telemetry ports. The 16-bit port numbers identifies the secure enclave node port and the supervisory workstation. The three words are not added to every update block because it reduces the information exposed during an eavesdrop on the network channel. Subsequent blocks after the first block are forwarded by the RTU to the same destination node.

5.2 Update Transfer Protocol

The supervisory workstation dedicated for the updates sends the first update block to the RTU, which verifies the IP address and port number of the sender and forwards it to the PLC's secure enclave. The secure enclave node receives the update block, validates it, and send feedback to the supervisory network.

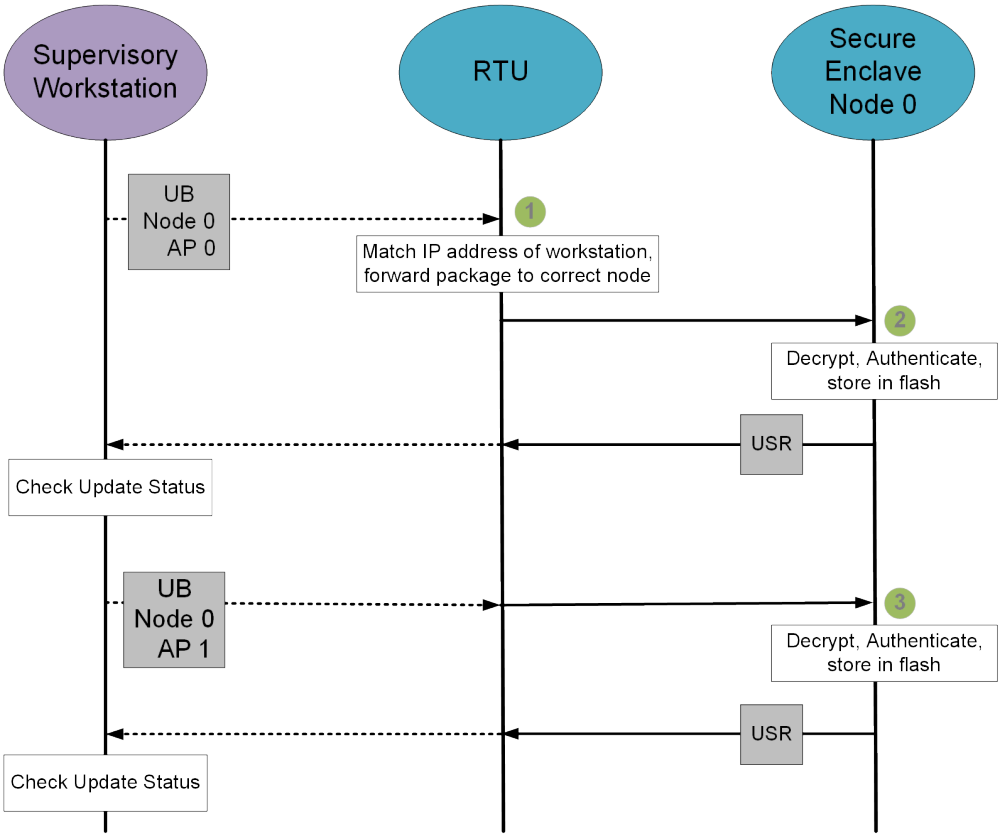

Figure 5.2: Update Protocol

Update Status Register

Meaningful feedback is very important during the block transfer to indicate a successful or an interrupted update session. At the node, the secure enclave maintains the update status register(USR) as shown in Table 5.6. Each bit of the USR has a significance in block validation.

Table 5.6: Update Status Register

Bit	**Description**
0	1-correct node, 0-incorrect node
1	1-correct update version, 0-incorrect update version
2	1-correct sequence number, 0-incorrect sequence number
3	1-All blocks are received, 0-Not all blocks are received
4	1-Authentication no error, 0-Error during authentication
5	1-Blocks are successfully transferred to application processor, 0-transfer unsuccessful
6-7	Undecided

Chapter 6

Architecture Overview

The secure enclave application processors interact with the other application processors, the plant, the production controller and the supervisory network. Figure 6.1 shows the interfaces of application processors with the supervisory network and boot flash. The boot flash is a non volatile memory that holds application software during the production session. At the start of the production session, first stage boot loader stored in ROM loads an application software and firmware from the boot flash. Typically the application processor that runs the decision module communicates with the supervisory network through the supervisory channel [10].

6.1 Architecture for Updates

A dedicated hardware module called the secure enclave update processor (SEUP) implements formally verified code and handles all incoming and outgoing traffic at the secure enclave during the update session as show in Figure 6.2. The SEUP provides all cryptographic services needed during the update session and after a PLC power cycle. The SEUP shares

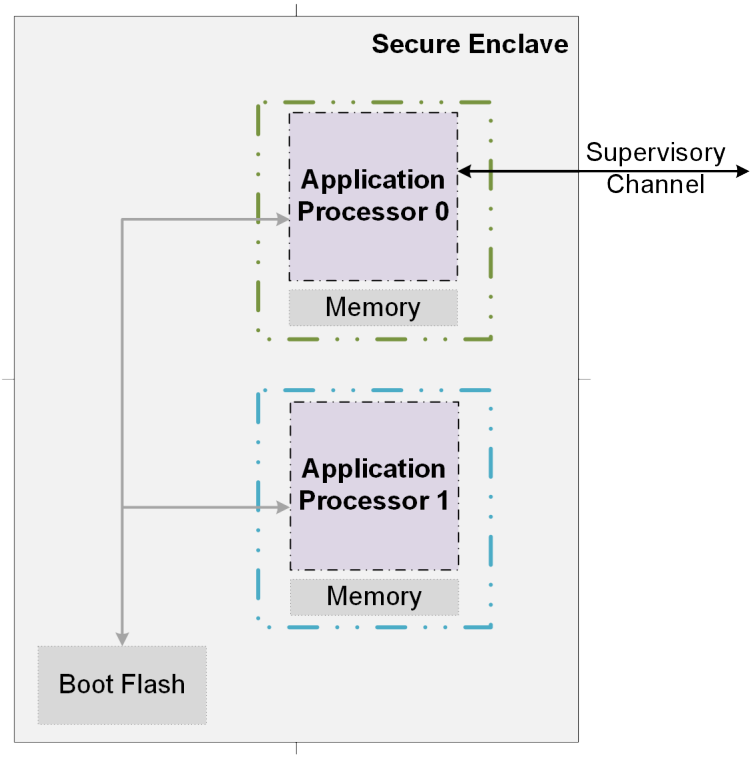

Figure 6.1: Secure Enclave Node

the supervisory channel of the application processor to receive update blocks and send update status to the supervisory layer workstation. A dedicated supervisory channel for the SEUP is not required as it does not interact with the supervisory network only during the production session. During the update session, the SEUP receives update blocks from the supervisory network channel. Cryptographic services are called to decrypt and authenticate these block.

The untrusted production controller and the SEUP do not interact or share any resources. In addition the application processors and the SEUP do not share any resources. Communication between the SEUP and the application processor is nonblocking, interrupt free, simplex and occurs only during the update session. This preserves trust generated at the secure enclave. Only the SEUP can access and modify the USR during an update session.

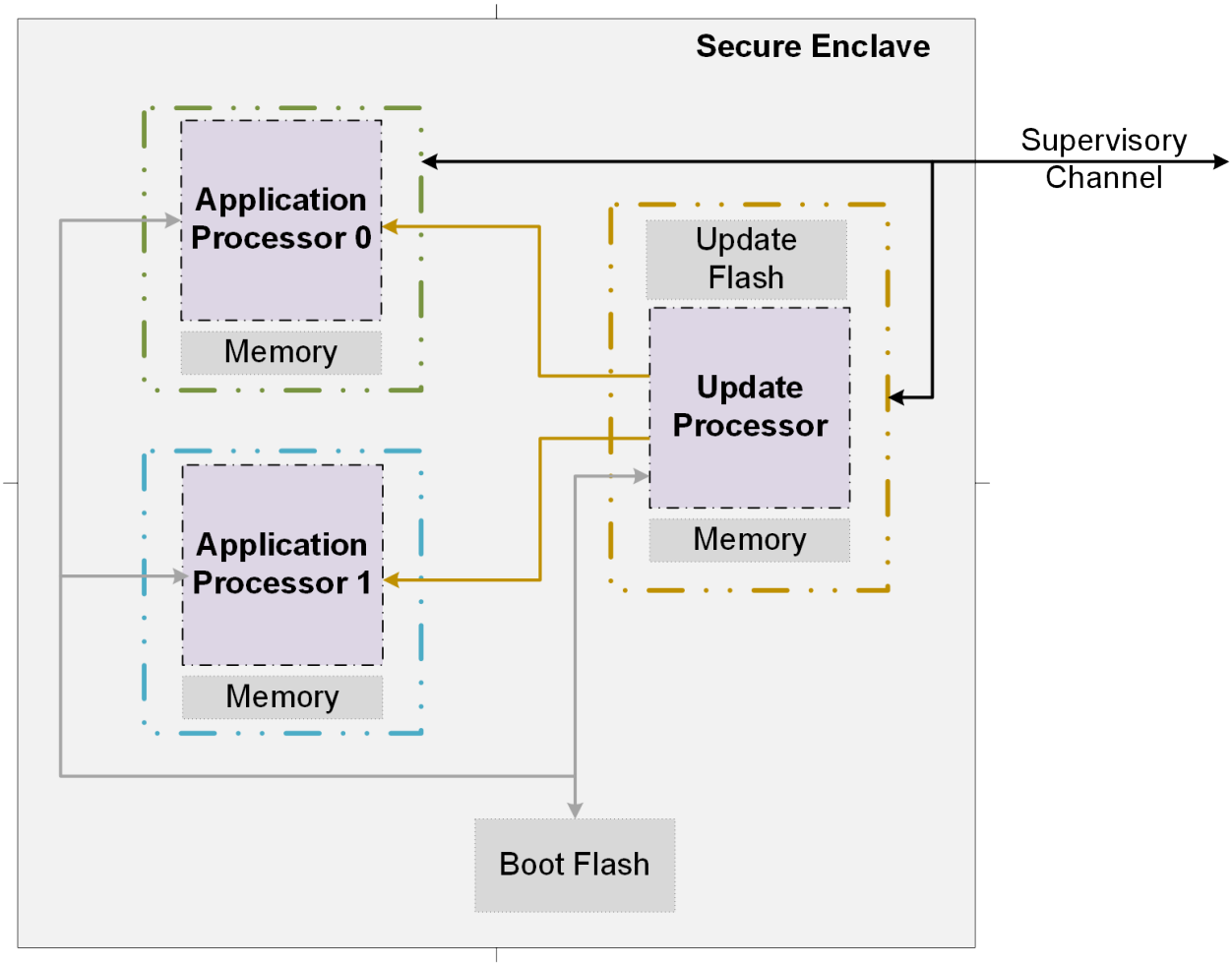

Figure 6.2: The Secure Enclave with SEUP

6.1.1 Update Flash

Non volatile storage is required to store the authentication and encryption key K_{ae} , application softwares and update blocks. The update flash memory is partitioned into three storage compartments as shown in Figure 6.3. Partition 0 holds authentication and encryption key K_{ae}. Partition 1 stores application softwares and update blocks sequentially for each application processor. Freshly received update blocks are stored at the partition 2.

Only the SEUP has access to the update flash. No other system resources can read or modify

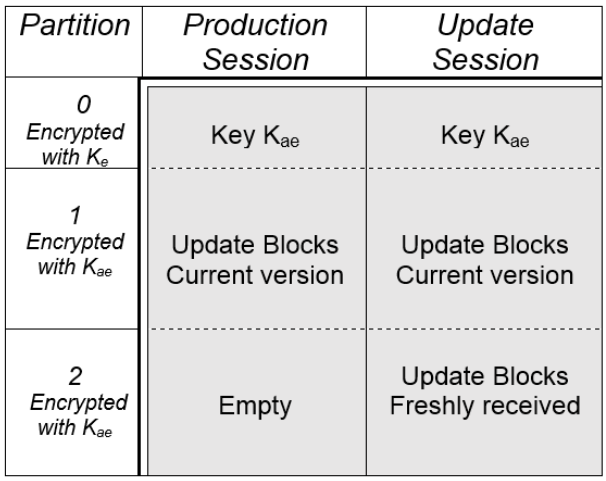

Figure 6.3: Update Flash

information in the flash. Every bit stored in the update flash is encrypted to avoid any data recovery from stolen or discarded equipment. System information such as a node port number, application processor numbers, and the update version ID can help an adversary gain more insight about the system and conduct covert attacks. We assume only authorized personnel have physical access to the hardware so physical attacks such as side channel attacks are not possible. The encryption algorithm for partition 0 is lightweight and encryption key K_e is stored at the hardware registers. Only the SEUP is allowed to read K_e.

6.1.2 SEUP

Update Session

The IP address and the port number fields of the first update block are not encrypted. The RTU uses them to forward blocks to the correct destination. The RTU forwards the next blocks to the same node. The SEUP at the node receives the update block and carries out the following steps.

Step 0: At the start of the update session the SEUP reads and decrypts partition 0 to retrieve K_e.

Step 1: Validity of the IP address is checked for the first update block. On successful authentication, the first three words are dropped. Then the SEUP decrypts and authenticates the update block with key K_{ae}.

Step 2: The update version ID and the update block sequence number are verified. The SEUP maintains a record of the last update block number. After dropping the IP address and port number words, the encrypted update block is stored in partition 3.

Step 3: The SEUP retrieves the encrypted update blocks from partition 2, one block at a time. Only decryption is sufficient, and authentication is not required . The block is forwarded to the application processor. Once all the update blocks are transfered to each application processor, the SEUP modifies the USR and sends it to the supervisory workstation. The contents of partition 2 are copied to the corresponding memory location of partition 1, and then partition 2 is erased. Accidental interrupts such as a transmission delay, power failure or malicious interrupts by an attacker may occur during a block transfer process. During such events, the flash memory prevents the system from going into an unrecoverable state. It is assumed that the application processors are already waiting for the update blocks at the interface. Without any additional handshaking, the SEUP forwards the update blocks to the application processors as shown in Figure 6.4.

6.1.3 Application Processor

The application processor implements single or multiple modules in the secure enclave. Depending on the application, the application processor communicates with other application processors or the production controller. Every application processor is connected to the

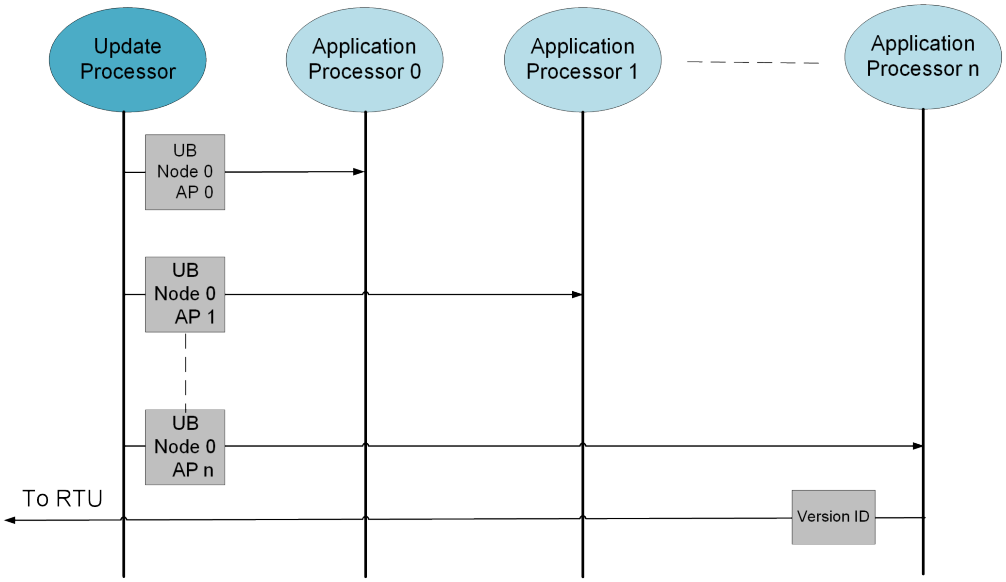

Figure 6.4: Update block distribution to the application processors

SEUP through a unidirectional bus to receive the update blocks. As shown in Figure 6.5, the application processor memory has three partitions that holds independent software modules. The partition 0 is reserved for vector tables, interrupts and peripherals. The partition 1 holds the application software that has empty memory locations for the update blocks to be written. The partition 2 is allocated for the dedicated software module - second stage boot loader (SBL) that handles communication with the SEUP.

Protocol After Power Cycle

After the power cycle, the FBL configures the application processor (Figure 6.5) and loads the SBL in the processor memory. SBL is the first program that runs on the application processor. It receives and copies the application software blocks and update blocks to memory locations. Once all blocks for the application processor are received, the application processor jumps to the start of the application software as shown in Figure 6.6.

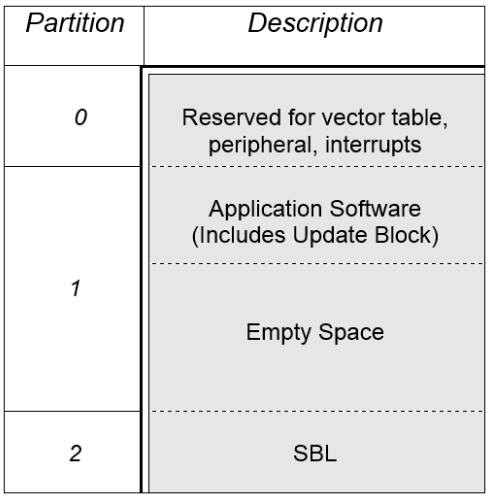

Figure 6.5: Application Processor Memory

Protocol during the Update Session

At the start of the update session, the application software again directs a processor to the start address of the SBL. During the update session, the SBL receives update blocks from the SEUP. The processor extracts a memory address from the received update block and copies the block to the memory location. After all update blocks are received, the SBL again directs the processor to the start address of the application software. During the update session, the SBL also updates the update version ID stored at the application processor. During the production session the application processor that communicates with the supervisory network sends the version ID of the update blocks as shown in Figure 6.4.

As stated previously, updates to the secure enclave modules are only parameter updates. Trust requirements of the secure enclave [10] [11] do not allow code changes. Bitstream updates for the SEUP require physical access. Since the SEUP's only function is to securely initialize other secure enclave processors, it does not contain data or code that is associated with physical processes and hence is affected by process aging. Changes to cryptographic keys, protocols, or algorithms require physical access to update the FPGA bitstream.

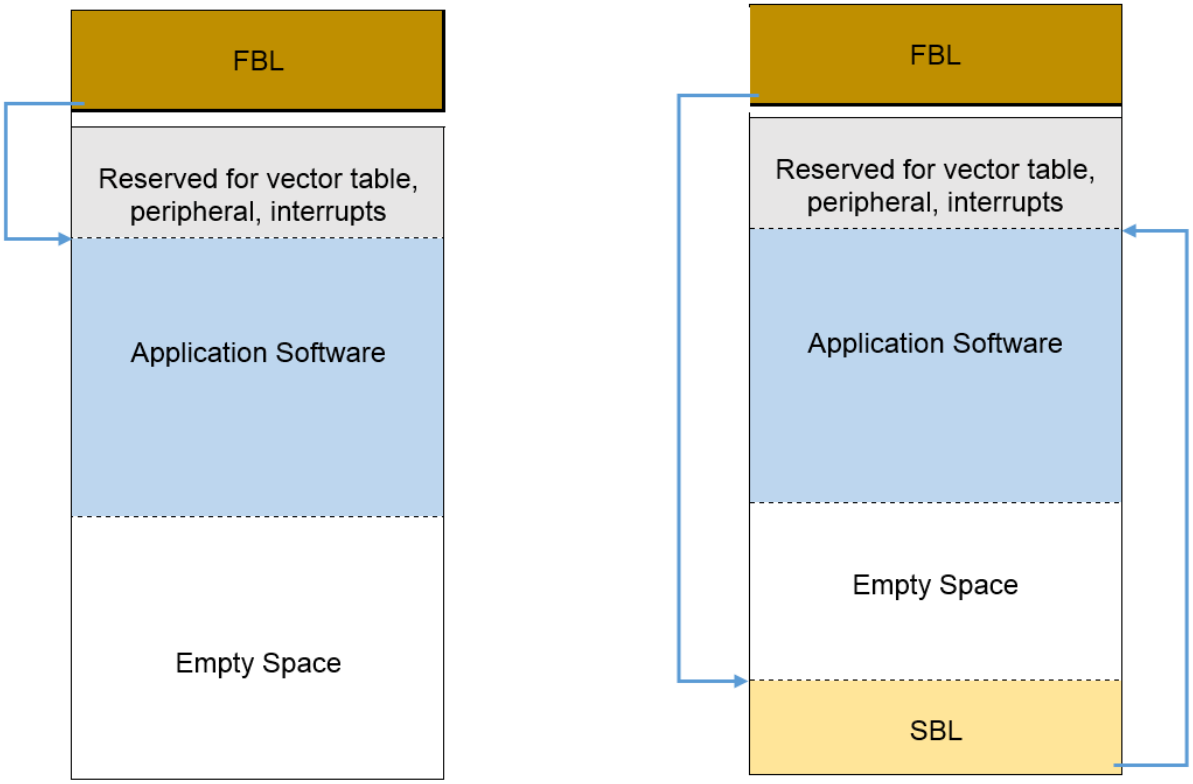

Figure 6.6: Change in Boot Loader Flow

Chapter 7

Implementation and Results

7.1 Implementation

The implementation of TAIGA is analyzed and our proposed update mechanism is integrated with TAIGA. Two hardware components - the update processor and the external update flash- are added to the TAIGA architecture.

7.1.1 Update Flash

As discussed in the previous chapter, the ZedBoard supports two non-volatile memories: SD card and external flash. These memories retain data during power cycles of the board. The update processor requires flash storage that is not accessible to other system resources. However, both these memories are accessible to the PS side of the SoC. Considering the unsafe production controller on the PS side, external flash memory is required. External flash memory, is connected to the ZedBoard EMIO bank that is accessible only to the PL.

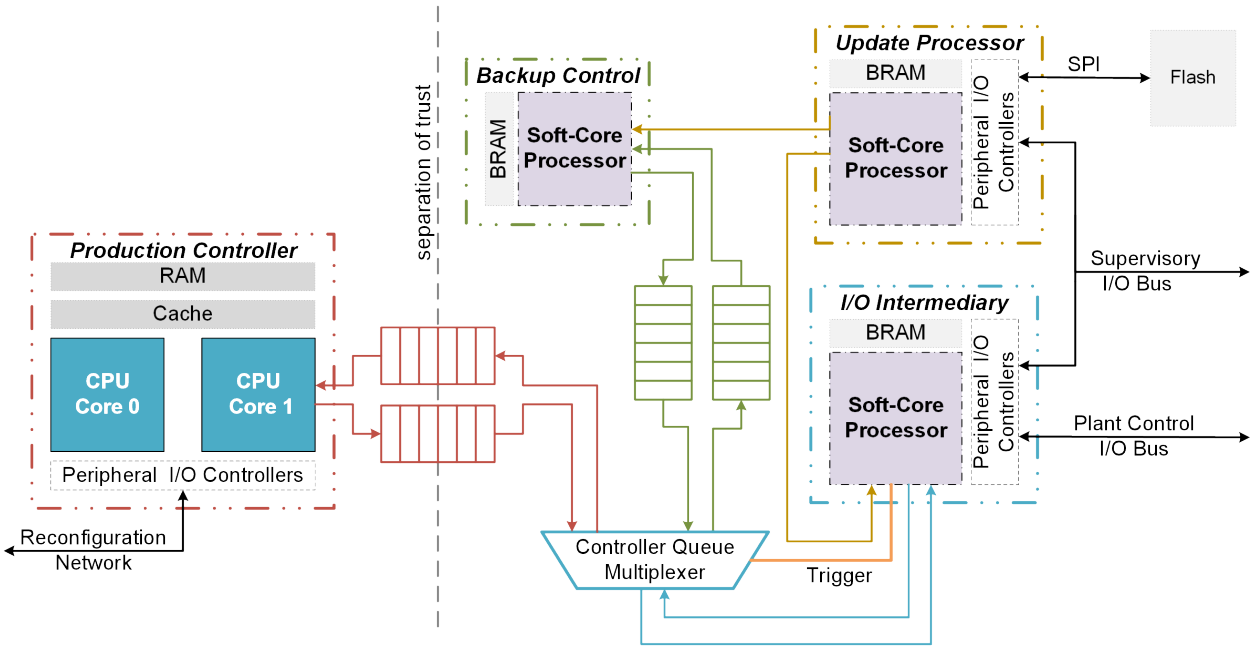

Figure 7.1: TAIGA and Update Processor

Peripheral Module Interface Serial Flash

Pmod SF [25] has a ST Microelectronics 16 MegaBit serial flash [26]. The update processor communicates with the flash using serial peripheral interface (SPI). SPI is a synchronous serial communication protocol with one master and multiple slaves. Each slave has a dedicated slave select signal (SEL). Only one slave can communicate with the master. The other three communication signals are : SCK - clock, MOSI - Master out Slave in and MISO - Master in slave out. Memory is organized into 32 sectors (of 16k words each) and each sector has 256 pages (of 64 words each). Byte address ranges from 0x00 to 0x1fffff.

7.1.2 Update Processor

The update processor is a softcore bare metal processor, and the software is developed in C. Bare metal applications use standalone Board Support Packages (BSPs) that incorporate

drivers and libraries required to interact with the hardware. Two BRAM modules are allocated to the update processor: 64 KB for local instruction memory and 64 KB for local data memory. During the update session, received update blocks are buffered in the BRAM before being authenticated and copied to the flash. During the production session, the update processor does not interact with the IOI module or the backup controller.

7.1.3 Interprocessor Communication

Xilinx provides a Fast Serial Link (FSL) [27] bus for communication between the update processor and multiple slave application processors. Each master-slave has a dedicated 32-bit wide FSL bus and one master can support a maximum of 16 slaves. FSL is a unidirectional FIFO-based communication bus that can be blocking or non-blocking. The implemented FSL is non-blocking, as the update processor without any additional handshaking writes validated update block to the FSL. It is assumed that the application processor is already waiting at the slave end of the FSL.

7.1.4 Update Processor Software

A USB-to-UART peripheral is connected to a Pmod port for supervisory communication. The Pmod is routed to the PL using EMIO pins and connected to an AXI UART peripheral configured with 8 data bits and a baud rate of 921,600. At the start of the update session, the update processor receives the update block from the supervisory node. The update undergoes decryption and authentication before it gets stored in the update flash. Authentication of the update block has two stages. At the first stage the cryptographic services are called by the update processor to generate a hash value of the update block. The generated hash value is compared with the received 5 words hash value and upon successful matching the control

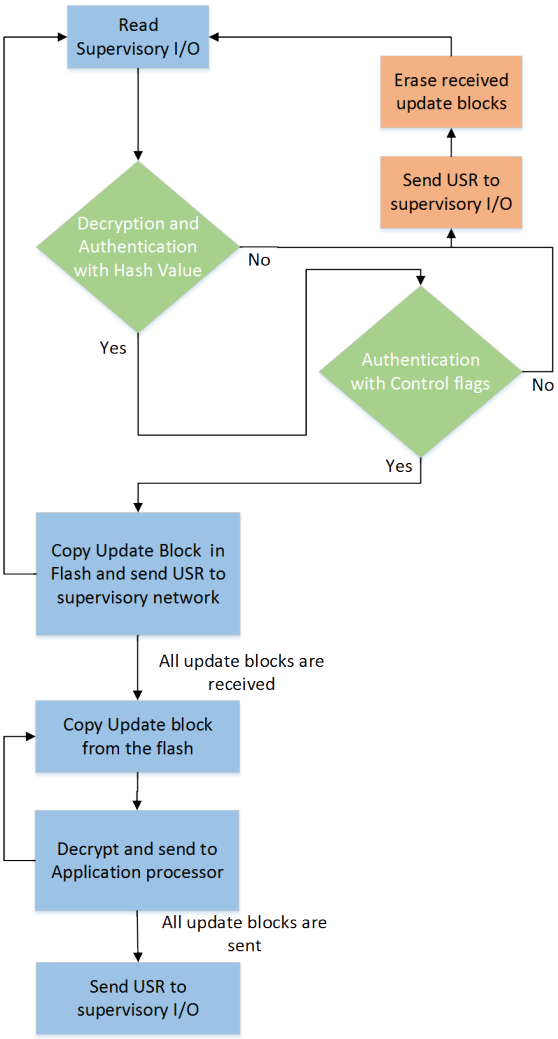

Figure 7.2: Update Processor Software flow

flags are checked. After authentication, the update blocks are stored in the application flash and updated USR is sent to the supervisory node. The same steps are repeated for the next update block until all update blocks for the node are received. A failed authentication immediately puts an end to the current update session. All received update blocks for given node are erased, the supervisory network is notified through the USR to end the current update session and start a new update session.

7.2 Results

The primary purpose of the SEUP is to provide trusted updates to the trusted components of TAIGA. Effectiveness of the SEUP and protocol is validated under different simulated network integrity attacks. Since SEUP is applied to TAIGA, the added cost in terms of resources and computation are important considerations. The addition of SEUP is compared to a TAIGA implementation wihout SEUP. Performance metrics such as resource utlization and execution time of various methods are discussed.

7.2.1 Resilience Against Attacks

Resilience of the update mechanism is assessed under replay attacks, and bit manipulation attacks.

Replay Attack

For replay attacks, an attacker copies an update block from the previous transmission, and replays it during the current update block transmission. The SEUP maintains parameters related to the update session such as an update block number, application processor ID, node ID, update version ID. Attack scenarios and raised control flag IDs are described in Table 7.1.

Bit Manipulation Attack

It is assumed that an adversary is able to change bits from the update block. During authentication of the update block, if the generated hash value is different from the received hash value, the update processor immediately rejects the packet, informs the supervisory node and aborts the update.

Table 7.1: Replay Attack

No.	**Replayed update block description**	**Control flag raised by**
1	same node and same application processor	Update block number
2	same node and different application processor	application processor ID
3	different node and same / different application processor	Node ID
4	all above scenarios with old version of updates	Update version ID

7.2.2 Resource Utilization

Resource utilization is an important metric to analyse the cost added by the SEUP. The SEUP is implemented with a micro-controller containing built-in peripheral resources, additional peripheral block for the update flash interface, and an external component. Comparison of the resources with the TAIGA and TAIGA + Update mechanism is listed in Figure 7.3. Single global buffer(BUFG) and Digital Signal Processor(DSP) slices are used by the production controller and the TAIGA. The SEUP does not add any new resources in both cases. FFs are used during routing for registers and buffers. BRAM is allocated for the instructions and local data memories of the SEUP MicroBlaze. While a significant portion of the FPGA resources are used, the Zynq-7010 used in the ZedBoard platform has fewer resources available than other chips in the family. The SEUP is successfully incorporated without requiring any excess resources.

7.2.3 Execution Time

PLCs are embedded platforms and execute multiple processes during the update session that have computation and processing latencies. For the SEUP, execution times of operations are

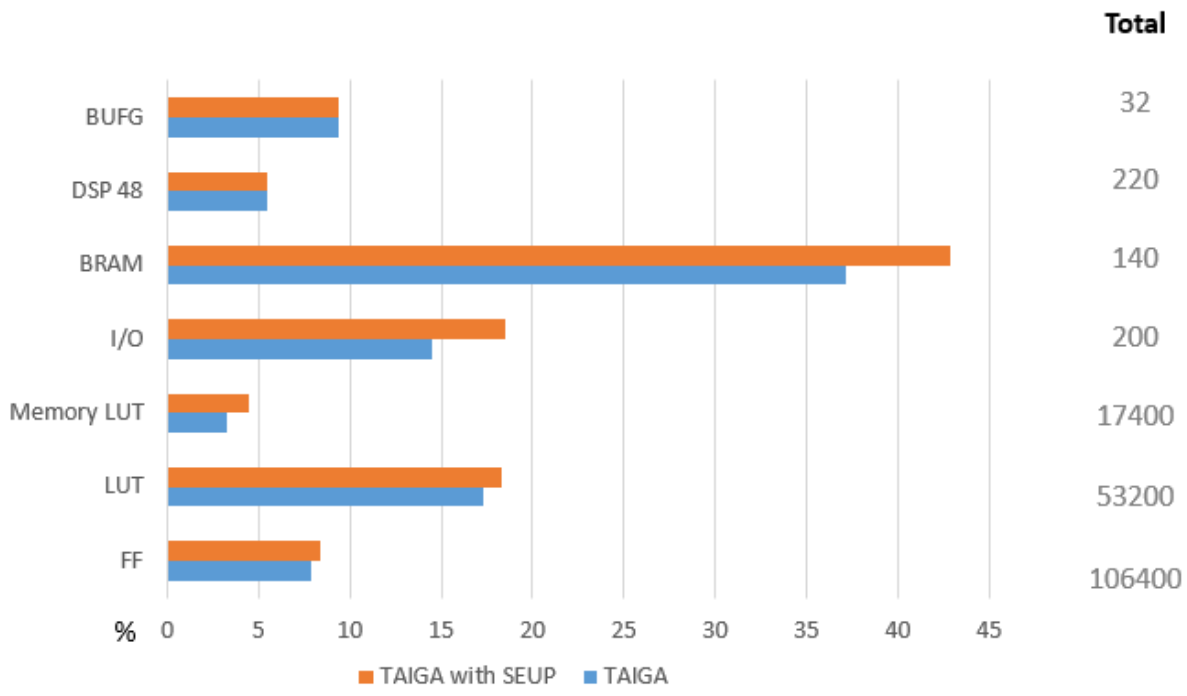

Figure 7.3: Resources Utilization

measured with a digital oscilloscope by toggling general purpose I/Os in software. The cryptographic services take the major portion of the time for the update session. The top bar graph from Figure 7.4 depicts the time required to authenticate a received update block, transfer it to an application processor and write it to the application processor's memory. It does not include transmission time of the update block from the supervisory workstation to the secure enclave node. Compared to a physical update process, the time required for the remote updates is much less. Update blocks are transfered to the secure enclave node during the update session, after the production session ends. In our mock ICS, communication between the RTU and the PLC is through a USB-to-UART channel that limits transmission speed to 921,600 baud. In practice however, communication between the RTU and the PLC is through Ethernet that has communication speeds up to 100 Mbits per second. Such high speed is enough to receive a set point command and the update block from the supervisory

network within one control cycle. This cuts down the time required for decryption and authentication during the update session. The bottom bar graph shows time required to load the application software from the update flash in the application processor memory after a power cycle.

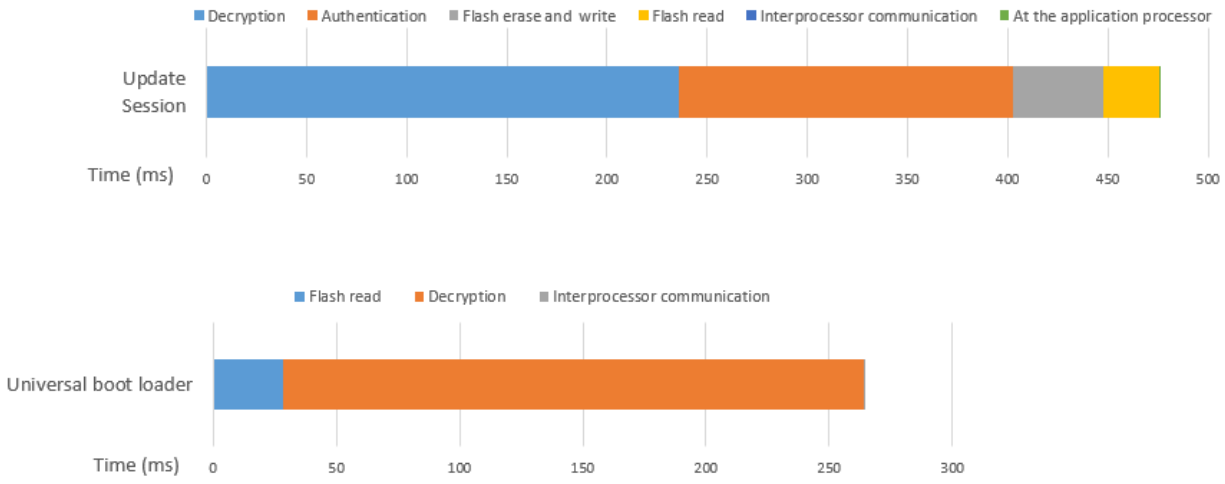

Figure 7.4: Execution Times

Chapter 8

Conclusions

The proposed remote update mechanism was successfully added to the TAIGA implementation on the programmable SoC. The proposed update transfer protocol is scalable for multiple nodes or application processors at a single node. The SEUP ensures safe storage of the update blocks in nonvolatile memory. The previous update blocks are erased only after the new updates are configured successfully. This prevents the system from going into an unrecoverable state during events such as a power failure. The proposed mechanism shows resilience against replay and bit manipulation attacks. The SEUP added to TAIGA maintains the trust generated at the secure enclave by the TAIGA since the isolated and dedicated SEUP does not share any resources with other processors.

8.1 Future Work

A SEUP is successfully implemented and incorporated with TAIGA. However to better integrate with the leaf node, implementation needs to be tested in the presence of physical processes. Current ICS control loops have a wired communication between the RTU and

PLCs. ICS environments also incorporate wireless communication and the current solution needs to implement wireless remote updates. Other technologies such as the Internet of Things incorporate multiple embedded devices connected to a central system through the Internet. Trusted remote updates to these embedded components can be provided using a similar architecture.

Bibliography

[1] Z. M. Lahlou, "System control and data acquisition," 2017.

[2] N. T. Chiluvuri, *A trusted autonomic architecture to safeguard cyber-physical control leaf nodes and protect process integrity.* PhD thesis, Virginia Tech.

[3] AVNET, *ZedBoard (Zynq Evaluation and Development), Hardware Users Guide*, 1 2014. Rev. 2.2.

[4] Xilinx, *Zynq-7000 All Programmable SoC Data Sheet: Overview*, 6 2017.

[5] A. M. Lepschy, G. Mian, and U. Viaro, "Feedback control in ancient water and mechanical clocks," *IEEE Transactions on Education*, vol. 35, no. 1, pp. 3–10, 1992.

[6] U. S. Department of Homeland Security, "ICS-CERT year in review 2015," 2016.

[7] P. K. Kerr, J. Rollins, and C. A. Theohary, *The Stuxnet computer worm: Harbinger of an emerging warfare capability.* Congressional Research Service Washington, DC, 2010.

[8] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. 32, pp. 3317–3318, July 2017.

[9] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," *Industrial Control Systems*, vol. 30, 2014.

[10] L. W. Lerner, *Trustworthy embedded computing for cyber-physical control*. PhD thesis, Virginia Polytechnic Institute and State University, 2015.

[11] S. Mohan, S. Bak, E. Betti, H. Yun, L. Sha, and M. Caccamo, "S3a: Secure system simplex architecture for enhanced security and robustness of cyber-physical systems," in *Proceedings of the 2nd ACM international conference on High confidence networked systems*, pp. 65–74, ACM, 2013.

[12] S. McLaughlin, "CPS: Stateful policy enforcement for control system device usage," in *Proceedings of the 29th Annual Computer Security Applications Conference*, ACSAC '13, (New York, NY, USA), pp. 109–118, ACM, 2013.

[13] K. A. Stouffer, J. A. Falco, and K. A. Scarfone, "Guide to industrial control systems (ICS) security," *Special Publication (NIST SP)-800-82 Rev 1*, 2013.

[14] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st international conference on High Confidence Networked Systems*, pp. 55–64, ACM, 2012.

[15] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*, pp. 355–366, ACM, 2011.

[16] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pp. 1806–1813, IEEE, 2012.

[17] L. Sha, "Using simplicity to control complexity," *IEEE Software*, vol. 18, no. 4, pp. 20–28, 2001.

[18] C. H. Houpis and S. N. Sheldon, *Linear Control System Analysis and Design with MATLAB®*. CRC Press, 2013.

[19] S. Tom, D. Christiansen, and D. Berrett, "Recommended practice for patch management of control systems," tech. rep., Idaho National Laboratory (INL), 2008.

[20] *Active platform management demystified: unleashing the power of Intel VPro (TM).*

[21] Apple Inc, *iOS Security Guide, iOS 10*, 1 2017.

[22] Raspberry Pi (Trading) Ltd., *Raspberry Pi Compute Module 3 (CM3)*, 10 2016. 1.

[23] D. Eastlake 3rd and P. Jones, "Us secure hash algorithm 1 (sha1)," tech. rep., 2001.

[24] N. F. Pub, "197: Advanced encryption standard (aes)," *Federal information processing standards publication*, vol. 197, no. 441, p. 0311, 2001.

[25] Digilent, *PmodSF Reference Manual*, 5 2016.

[26] ST Microelectronics, *M25P16*, 1 2007. Rev. 10.

[27] Xilinx, *AXI Reference Guide*, 7 2011.