U.S. DEPARTMENT OF ENERGY

PNNL-18744 Vol. 2

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Cryptographic Trust Management Design Document

Version 1.1

TW Edgar
SL Clements
MD Hadley
WM Maiden
DO Manz
SJ Zabriskie

January 2010

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm


This document was printed on recycled paper.
(9/2003)

Revision History

Release	Date	Comments
0.5	01/28/2010	Initial draft
1.0	01/31/2010	Final 1.0 version
1.1	05/31/2010	Release process edit

Acronyms and Abbreviations

This section contains a list of acronyms and abbreviations used in this document, along with their corresponding meanings.

3DES – Triple Data Encryption Standard or Triple DES. A block cipher algorithm that provides symmetric key cryptography. A legacy algorithm that should be avoided.

AAA – Authentication, Authorization, and Accounting. A framework of services for providing centralized authorization.

AES – Advanced Encryption Standard. A block cipher algorithm that provides symmetric key cryptography. Currently recommended with the Galois/Counter Mode (GCM) for symmetric encryption. FIPS approved symmetric key cryptography.

AMI – Automated Metering Infrastructure. A Smart Grid technology using Smart Meter technology to enable remote monitoring, control of metering, and demand and response efficiencies.

API – Application Programming Interface. A set of routines, data structures, object classes, and/or protocols provided by libraries and/or operating system services in order to support the building of applications.

CBC – Cipher Block Chaining. A mode of encryption where the cipher block from the previous round of encryption is XORed with the next round's plaintext block before being encrypted.

CCM – Counter with CBC-MAC. A mode of operation for block ciphers. CCM provides authentication and encryption for a cryptographic algorithm.

CFB – Cipher Feedback. A mode of encryption where the cipher block from the previous round of encryption is encrypted and then used as a cipher stream to be XORed with the plaintext block to create the next round's cipher block.

COTS – Commercial-Off-The-Shelf. Commercial-Off-The-Shelf products are ready-made commodity products that can be used in place of in-house creation or development. COTS are designed to quickly fulfill a needed product or solution.

CTM – Cryptographic Trust Management.

CTR – Counter mode encryption. A mode of operation for block ciphers. CTR enables a block cipher to turn into a stream cipher.

DH – Diffie-Hellman Key Agreement Protocol. The seminal Diffie-Hellman Key Agreement Protocol allows two parties to exchange information over an insecure environment such that, at the end of the exchange, both parties can communicate securely without eavesdroppers overhearing. DH Key Agreement does not require any pre-shared or out-of-channel information to be passed. FIPS approved for key exchange and agreement.

DNP or DNP3 – Distributed Network Protocol. An open protocol used within process control networks.

DNS – Domain Name System. A hierarchical naming system for computers, services, or any resource connected to a network.

DOE – U.S. Department of Energy.

DSA – Digital Signature Algorithm. DSA provides a method for creating and using digital signatures. Digital Signatures are a means of computationally verifying the authenticity of digital information. FIPS approved for digital signatures.

EAP – Extensible Authentication Protocol. EAP provides a framework for modular authentication. Commonly used for wireless networks, EAP provides an interface and function commonality for implementing authentication on a network.

ECB – Electronic Code Book. ECB is an early mode of operation for providing cipher block symmetric key cryptography. ECB provides only confidentiality or integrity but not both at the same time. ECB has been deprecated in favor of newer, more robust modes of operation.

ECDH – Elliptical Curve Diffie-Hellman Key Agreement. ECDH is a variant of DH Key Agreement that utilizes underlying elliptic curve algebraic structures. Functionally identical to traditional discrete logarithm DH key management, ECDH provides equivalent security at reduced key sizes and increased processing speeds. FIPS approved for key exchange and agreement.

ECDSA – Elliptical Curve Digital Signature Algorithm. ECDSA is a variant of DSA that utilizes underlying elliptic curve algebraic structures. Functionally identical to traditional DSA, ECDSA provides equivalent security. FIPS approved for digital signatures.

GUI – Graphical User Interface.

HAN – Home Area Network. A network of computers and resources contained within a user's home. This term is usually used in conjunction with Smart Meters or other service provider equipment that bridges the gap between a user's home network and the service provider's network.

HMAC – Hash-based Message Authentication Code. HMAC is a method for calculating message authentication codes which are used to authenticate messages. HMAC is used to provide information integrity and authenticity.

HTTP – Hypertext Transfer Protocol. HTTP is a popular method for sharing information on the Internet. An application layer protocol, HTTP provides the means to share hypertext documents and led to the establishment of the World Wide Web.

ICCP – Inter-Control Center Communication Protocol, also known as IEC 60870-6 TASE.2. A protocol to provide data exchange services between peer control stations.

ICS – Industrial Control System

IDS – Intrusion Detection System. IDSs are software or hardware that monitor networks and computers for unauthorized or malicious activity and behavior. Often categorized into two methods, signature-based and anomaly-based, IDSs are used to provide awareness of cyber attacks on a system.

IEC – International Electrotechnical Commission.

IEEE – Institute of Electrical and Electronics Engineers. IEEE is an international organization for the advancement of technology. The IEEE provides membership, publications, and standards associations for various electricity and technology related fields.

IP – Internet Protocol. IP is an address routable protocol for communicating data across a packet-switched network.

IPS – Intrusion Prevention System. An IPS is a variant of an IDS that actively attempts to thwart and prevent malicious and unauthorized behavior on a system or network.

ISA – The International Society of Automation.

IT – Information Technology. The science of design, development, and support of computer-based information systems.

KMIP – Key Management Interoperability Protocol. KMIP is designed to create a comprehensive key management protocol enterprise-wide.

LAN – Local Area Network.

LDAP – Lightweight Directory Access Protocol. An application protocol for querying and modifying directory services running over TCP/IP.

LUNA – A hardware module for protecting certificate authority signing keys.

MD5 – Message Digest algorithm 5. MD5 is a popular cryptographic hash function. Hash functions create a fixed size message digest for any message inputted. Hash functions provide a means of reducing a message size but still provide uniqueness, which is useful in digital signatures and authentication.

MITM – Man-in-the-middle attack. Man-in-the-middle attacks consist of a third party interjecting itself between two communicators. The attacker impersonates each communicator to the other. This way, the malicious third party can interject messages, redirect messages, or even read and modify messages.

NAS – Network-Attached Storage. Network-attached storage is accessible directly on the LAN through LAN protocols such as TCP/IP.

NIC – Network Interface Controller. The NIC is a hardware device that allows a computer to connect to a computer network. The NIC operates at the physical and data link layers for network communication, allowing applications to communicate over the connected network. The most popular protocol is Ethernet over wires.

NIST – National Institute of Standards and Technology.

NSTB – National SCADA Test Bed. A U.S. Department of Energy program that addresses the security challenges of energy sector control systems.

OASIS – Organization for the Advancement of Structured Information Standards. A non-profit standards body that generally focuses on web related standards.

OFB – Output Feedback. OFB is an early mode of operation for providing cipher block symmetric key cryptography. OFB provides only confidentiality or integrity but not both at the same time. OFB has been deprecated in favor of newer more robust modes of operation.

OS – Operating System. An Operating System is a large application that runs on top of device hardware. The OS acts as an interface and mediator between the various hardware components and higher level applications and users.

PCSF – Process Control System Forum.

PKI – Public Key Infrastructure. Public Key Infrastructure is an overarching conglomeration of individuals, devices, software, hardware, and policy used to create, manage, and delete digital certificates. PKI requires a trusted party to vouch for and approve trusted credentials for entities. This web of trust requires an entity to only trust the third party to accept a new credential for a new entity.

PNNL – Pacific Northwest National Laboratory.

RAID – Redundant array of independent disks. A technology that provides data redundancy and duplication across multiple independent storage devices.

RBAC – Role-Based Access Control. RBAC is a means of limiting access to resources and information based upon an entity's role. Roles are dictated by policy and determine what resources or information a person or entity can access based upon the role assigned.

RC2 – Rivest Cipher 2. RC2 is a block cipher used to provide symmetric key cryptography. Created by Ron Rivest, RC2 has been largely deprecated in favor of newer block ciphers.

RC4 – Rivest Cipher 4. RC4 is a block cipher used to provide symmetric key cryptography. Created by Ron Rivest, RC4 in one of the most wide-spread block cipher in use today. Used in SSL and WEP among other applications, RC4 does have significant vulnerabilities and should be avoided in favor of a newer method.

RC5 – Rivest Cipher 5. RC5 is a block cipher used to provide symmetric key cryptography. Created by Ron Rivest, RC5 was designed to address the issues in RC4.

RSA – Rivest, Shamir, and Adleman algorithm. RSA is an algorithm for public-key cryptography. Widely used, RSA allows two parties to communicate securely over an insecure medium without using pre-shared keys. RSA relies on the difficulty of prime factorization for its security and is believed to be secure with up-to-date and correct implementations.

SAN – Storage Area Network. A high-speed special-purpose network that interconnects different kinds of data storage devices with associated data servers for a larger network of users.

SATA – Serial ATA (Serial Advanced Technology Attachment). A computer bus technology primarily designed for transfer of data to and from a hard disk.

SCADA – Supervisory Control and Data Acquisition. SCADA refers to an industrial control computer system for controlling a process.

SHA – Secure Hash Algorithm. SHA is a set of cryptographic hash functions designed to create a fixed size digest or summary of a given message. SHA is used in digital signatures and authentication. Earlier

versions of SHA include SHA-0 and SHA-1, both of which have been deprecated in favor of SHA-2 with 256 and 384 bit digests.

SSCP – Secure SCADA Communications Protocol. A software approach developed by PNNL to provide secure communications in SCADA systems.

SSH – Secure Shell. A secure remote terminal protocol.

SSL – Secure Sockets Layer. SSL is a cryptographic protocol that provides communication security over computer networks. SSL encrypts communication at the Transport Layer from sender to receiver. SSL has been deprecated in favor of TLS. SSL is widely used on the Internet.

TCP – Transmission Control Protocol. TCP is a fundamental component of the Internet protocol. The TC protocol defines computer network communication at the Transport Layer. TCP provides reliable in order message delivery between computers.

TGT – Ticket Granting Ticket.

TLS – Transport Layer Security. TLS is a cryptographic protocol that provides communication security over computer networks. TLS encrypts communication at the Transport Layer from sender to receiver.

TPM – Trusted Platform Module. TPM is a specification for a secure embedded device that stores and manages cryptographic keys and information within a crypto-processor. TPM can be embedded into devices that require reliable authentication and encryption.

UI – User Interface.

VPN – Virtual Private Network. A VPN is a logical network overlaid and tunneled within an existing computer network. VPNs are designed to provide additional functionality and connectivity to geographically diverse computing devices. VPN is naively thought to automatically provide security but care must be taken in implementation and use of VPNs to ensure that security is maintained.

XOR – Exclusive Disjunction. A type of logical operation on two logical values that produces a result of "true" if exactly one of the operands has a value of true.

WEP – Wired Equivalency Protocol.

Contents

Figures

Tables

1.0 Introduction

1.1 Identification

This document is the design document for the PNNL NSTB Cryptographic Trust Management system.

1.2 Document Purpose, Scope, and Intended Audience

1.2.1 Document Purpose

The purpose of this document is to detail the design choices that were evaluated for the Cryptographic Trust Management (CTM) System. This document will be used to focus and drive the development of a prototype system. The document will clearly define how the system should be developed and why the specified design choices were made.

1.2.2 Document Scope and/or Context

The scope of this document is the design of a Cryptographic Trust Management System that complies with the 17.1. This document is intended to be a living document that starts at a high level and continues to be updated with details until a very low level design is produced that is usable for the development of a prototype system.

In 2006, the Department of Energy published the Roadmap to Secure Control Systems in the Energy Sector document (Roadmap). This Roadmap identified near and long-term milestones and needed technologies, some of which require the ability to manage cryptographic information in a manner that supports the operational requirements of the electric sector. An example milestone is "Widespread implementation of methods for secure communication between remote access devices and control centers that are scalable and cost-effective to deploy." In order for a cryptographic solution to be deployed by an asset owner, and in order for a technology to scale to the required number of devices, a new method to manage the cryptographic keys is necessary. This NSTB project task is intended to meet these requirements and assist with the adoption of secure communication technologies in the electric sector.

1.2.3 Intended Audience for Document

The intended audience of this document is industry advisors, NSTB clients, and the developers of the Cryptographic Trust Management prototype system. The industry advisors and NSTB clients will be reviewers of the document and evaluators for applicability and necessary course corrections. The developers will leverage this document to design a prototype system.

1.3 System and Software Purpose, Scope, and Intended Users

1.3.1 System and Software Purpose

The purpose of this project is to create a system to manage cryptographic material and negotiate trust between entities to support the deployment of technologies developed to meet multiple milestone and goals of the DOE NSTB Roadmap to Secure Control Systems in the Energy Sector.

1.3.2 System and Software Scope/Context

The scope of this project is to create a system, including both hardware and software, to manage cryptographic keys and identities to support the deployment of cryptographic solutions to secure control system environments. This project will focus on the control system network. The focus will be to support systems and services only within the boundaries of the control system network and will not focus on external enterprise applications.

1.3.3 Intended Users for the System and Software

Intended users for the CTM System include operational security personnel, auditors, and incident response/cyber forensic investigators. Examples of each user group and the functions their functions that will be assisted by assisted by the CTM include:

- Operational Security Personnel
  - Monitor current state security of the process control network
  - Remedy policy violations
  - Perform manual key management functions.
- Auditors
  - Review audit reports and ensure policy and regulatory compliance.
- Incident Response/Cyber Forensic Investigators
  - Review audit logs for security events or policy violations.

1.4 Document Overview

This document consists of the following sections:

System Overview – Provides an overview of the problem space, how this system will apply, and why it is important. System use cases will also be described in this section.

System Architecture – Provides a definition of the high level architecture of the system. This section will describe the system as a whole and how all of the pieces fit together. Later sections will define the design of the system's components in more detail.

Cryptographic Trust Repository Component – Describes the design of the Cryptographic Trust Repository component.

Key Generator Component – Describes the design of the Key Generator component.

Key Manager Component – Describes the design of the Key Manager component.

AAA Service Component – Describes the design of the AAA Service component.

Cryptographic Remote Trust Component – Describes the design of the Remote Trust component.

User Interface Component – Provides a definition of the user interfaces of the system and the work flow of the user processes.

Trust Lifecycles – Provides a definition of the work flow for the lifetime states of cryptographic material in the Cryptographic Trust Management System.

Communication Architecture – Provides definitions of the components of the system in more detail than the System Architecture section provides.

Device Registration – Defines the data necessary for registering and provisioning a device to be used with the CTM System.

Third Party AAA Service Requirements – Defines the minimum functional requirements necessary for a third party to interoperate with the trust negotiation portion of the CTM System.

Legacy Retrofitting – Describes possible methods of retrofitting current process control networks with applications that would work with the CTM System.

References – Provides a list of references utilized in this document.

Appendix A: Requirements Traceability – Provides a mapping of requirements from the Requirements document to the components of the system and how the requirement is met.

Appendix B: Glossary – Provides an alphabetical list of technical terms and their definitions in the context of this document.

2.0 System Overview

2.1 Problem Domain

Process control systems are moving toward more connectivity to IT infrastructure to improve efficiencies and increase reliability and safety. Routable communication infrastructure provides the benefit of remote control and increased connectivity between applications. The added connectivity of IT however, increases the attack surface of the environment as well as integrating common technologies that have well established risks.

As the industrial control system (ICS) networks converge, security technologies are being developed to mitigate the risk of the added connectivity. However, the security technologies currently being developed are all silo systems that only provide management services for a single application or a single vendor's products. Such operational cost of managing and maintaining these separate security systems is significant for industries that historically have not had the knowledge and expertise for these purposes. The operational costs become prohibitive given that it takes multiple security systems to reduce the risk to an acceptable level.

The lack of a scalable technology to manage cryptographic keys for control systems hinders the deployment of vendor products to secure control system communication. Without an industry acceptable, scalable, secure, and robust mechanism to create cryptographic keys that supports the operational requirements of critical infrastructure asset owners, no cryptographic solution will be widely deployed. Comments received during the 2008 PCSF conference and peer reviews for the Hallmark project echo this sentiment. Industry requires a cryptographic key management solution to further the deployment of technical solutions and to eliminate the risk associated with control system communication.

The introduction of cryptography into control systems represents a significant challenge to vendors, asset owners, and standards bodies. Security solutions can be very complex and easily misconfigured without expertise. Improperly configured security can lead to vulnerabilities at best and operational problems at worst. Operational impact is a major concern because the cryptographic goals for control systems differ significantly from corporate IT or Internet sites due to their control of physical processes. The security objectives of IT Systems in order of importance are confidentiality, integrity, and availability. In contrast, the security objectives for control systems are ordered availability, integrity, and lastly confidentiality. The physical processes controlled by control systems require high availability and reliability and the integrity and confidentiality security objectives are in support of the continued operation of the system.

In addition, the use of cryptography in control systems must support the multiple operational needs of asset owners without adversely affecting safe and reliable operations. The cryptographic key management problem is made more complicated by the number of vendors creating security products, regulatory requirements to identify and protect critical information, the automation of manual processes, and the predictable nature of control system communication.

Another hurdle to overcome is the potential size of a control system network. Managing cryptographic keys for a small utility with 30 substations can be done without automation. Managing keys for an automated meter reading environment with millions of smart meters cannot.

2.2 Standards Recognition

At the time this document was written, there were numerous standards activities surrounding cyber security in critical infrastructure networks. Large standards organizations, such as NIST, ISA, IEEE, and IEC, are all actively working on standards to secure process control networks. The system design defined in this document is targeted at meeting some of the requirements laid out in these standards. The list below is representative, not comprehensive, of the requirements targeted by this system.

All requirements are excerpted from "Catalog of Control Systems Security: Recommendations for Standards Developers," September 2009 and "Smart Grid Cyber Security Strategy and Requirements," September 2009.

Table 2.1. Standards Requirements for Critical Infrastructure Networks Cyber Security

Requirement	Text	Compliance
2.8.8 Communication Integrity	The control system design and implementation protects the integrity of electronically communicated information.	CTM System supports applications providing communication integrity.
2.8.9 Communication Confidentiality	The control system design and implementation protects the confidentiality of communicated information where necessary.	CTM System supports applications for communication confidentiality.
2.8.11 Cryptographic Key Establishment and Management	When cryptography is required and employed within the control system, the organization establishes and manages cryptographic keys using automated mechanisms with supporting procedures or manual procedures.	CTM System provides centralized manual and automated key management functionality.
2.8.12 Use of Validated Cryptography	The organization develops and implements a policy governing the use of cryptographic mechanisms for the protection of control system information. The organization ensures all cryptographic mechanisms comply with applicable laws, regulatory requirements, directives, policies, standards, and guidance.	CTM System centralizes the generation of and policies governing cryptographic material. This makes it much easier to ensure that the cryptography used across the process control network is strong enough for what it is protecting.
2.8.19 Security Roles	The control system design and implementation specifies the security roles and responsibilities for the users of the system.	CTM System utilizes a role-based access control model.
2.8.20 Message Authenticity	The control system provides mechanisms to protect the authenticity of device-to-device communications.	CTM System supports applications providing message authenticity.

Table 2.1. (contd)

Requirement	Text	Compliance
2.14.7 Software and Information Integrity	The system monitors and detects unauthorized changes to software and information.	CTM System logs changes to the policies, roles, and cryptographic material that it manages.
2.15.1 Access Control Policy and Procedures	The organization develops, disseminates, and periodically reviews and updates: 1. A formal, documented, access control policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance 2. Formal, documented procedures to facilitate the implementation of the access control policy and associated access controls.	CTM System forces users to create roles for access privileges. Also, the system provides centralized authentication and authorization services.
2.15.2 Identification and Authentication Policy and Procedures	The organization develops, disseminates, and periodically reviews and updates: 1. A formal, documented, identification and authentication policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance 2. Formal, documented procedures to facilitate the implementation of the identification and authentication policy and associated identification and authentication controls.	CTM System forces users to authenticate before authorizing any access requests.
2.15.5 Authenticator Management	The organization manages system authenticators for users and devices.	CTM System forces users to authenticate before authorizing any access requests. All devices and users require credentials for authentication to the system.
2.15.12 Device Identification and Authentication	The system uniquely identifies and authenticates an organization-defined list of devices before establishing a connection.	The purpose of the CTM System is to identify and authenticate devices as they join the network and as they request access and permissions.

Table 2.1. (contd)

Requirement	Text	Compliance
2.18.5 Control System Connections	The organization: 1. Authorizes all connections from the system to other systems outside the authorization boundary through the use of system connection agreements 2. Documents the system connections and associated security requirements for each connection 3. Monitors the system connections on an ongoing basis verifying enforcement of documented security requirements.	CTM System provides a capability for real time per connection trust negotiation with third parties for assurance.

Additionally, "Smart Grid Cyber Security Strategy and Requirements," September 2009 Appendix D, "Bottom Up Security Analysis of the Smart Grid" contains a list of cyber security issues and problems that need to be addressed. The CTM System design described in this document will address, at least partially, the following issues identified:

D.5	Authenticating and Authorizing Users to Substation IEDS
D.7	Authenticating and Authorizing Maintenance Personnel to Meters
D.9	Authenticating Meters to/from AMI Head Ends
D.17	Key Management for Meters
D.23	Non-Specific Cyber Security Issues
D.24	Key Management and PKI
D.27	Authentication
D.30	Distributed vs. Centralized Model of Management
D.31	Local Autonomy of Operation
D.36	Trust Management
D.37	Management of Decentralized Security Controls
D.40	Authenticating Users to Control Center Devices and Services
D.41	Authentication of Devices to Users
D.42	Entropy
D.48	Forensics and Related Investigations
D.49	Roles and Role-Based Access Control

2.3 System Goals

The Cryptographic Trust Management System is being designed to enable the integration of cryptographic technical security controls while still maintaining an affordable operational cost and a scalable system. Underlying this overall goal are the following six functional goals of the system.

Centralize cryptographic material generation – Process control system field equipment generally does not have the hardware or available resources to properly and efficiently generate cryptographic material. Therefore, a centralized generation facility would ensure that all cryptographic material is generated according to best practice. Also, centralizing the generation of all cryptographic material provides an efficient method to enforce cryptographic strength policies across the network.

Centralize cryptographic audit enforcement – Centralized cryptographic audit logs enable users to quickly view the current status of cryptographic material across the process control network. It can also provide notifications of expired cryptographic material and policy violations in near real-time.

Secure storage and backup of cryptographic material – The cryptographic material is the foundation of cryptographic security. Maintaining its secrecy is critical to maintaining the security of the security controls. Centralized storage of cryptographic material enables strong security controls to protect it. Loss of cryptographic material can also cause problems when needing to retrieve old data. Securely archiving cryptographic material ensures that it will be available should it be needed in the future.

Automate key management services – Device-to-device operation should require little human interaction to operate. A framework for automating key management for devices relieves the operational burden of key management and increases the appeal of security applications. As part of the framework, a notification system to alert automation-enabled devices of expiring or revoked cryptographic material protects against hidden or forgotten cryptographic material that would remain as a vulnerability.

Centralize AAA Services – Centralized AAA Services will provide a framework to enable AAA Services across the process control network, and in doing so make identity and authorization management more efficient. Centralized services will enable role-based access control (RBAC). Also, centralized authentication will enable system-wide access policy creation and personnel credential management.

Increase assurance of third party connectivity – Current process control networks often require access to equipment for third parties to provide maintenance. Best practice requires the distribution of credentials to third party employees for access control. However, the personnel controls of these third party employees are under the auspices of the third party. This places the additional requirement of having contractual agreements for notification of personnel actions and potentially having supervision of on-site access. A framework for trust negotiation of third party employees enables dynamic real-time access control, instead of relying on contractual assurances.

2.4 Use Cases

2.4.1 Provision System

Goal – The Cryptographic Trust Management System is configured and in an operational state ready to perform tasks.

Summary – The CTM System will require some initial configuration before it is operational. This use case defines the process to perform the initial configuration of the system.

Actors – User.

Preconditions – None.

Triggers – User manually triggers this use case.

Event Flow – User installs/creates initial key material (root certificate).

2.4.2 Request Key Material

Goal – The key material requested is created, stored, and delivered to the requesting actor.

Summary – This use case describes an automated method for applications, systems, or devices to retrieve key material from system.

Actors – Application, system, or device.

Preconditions – The actor must communicate via a supported protocol to enable the request and delivery of the key material.

Triggers – The actor sends a request for key material to the CTM System.

Event Flow –

1. Actor establishes secure connection with the CTM System.
2. Actor sends key material request to the CTM System. The request includes what type of key material, how much key material, and other vital information.
3. The CTM System generates key material.
4. The CTM System stores key material and metadata (e.g., key material users and lifetime of key material).
5. The CTM System transmits key material to actor.

Implementation Examples –

- SSCP device requesting master keys or pre-shared key list.
- ZigBee® gateway requesting link and application keys for new joining device.

2.4.3 Request Communication Key

Goal – Keys enabling two actors to communicate are distributed to the two actors.

Summary – This use case describes an automated method for an actor to request a key that enables it to communicate with another actor. This key may already exist and be in use by the second actor, or it may need to be created and distributed to both devices.

Actors – Application, system, or device.

Preconditions – The actor must communicate via a supported protocol to enable the request and delivery of the key material.

Triggers – The actor sends a request for a key to communicate with another actor.

Event Flow –

1. Actor sends request to the CTM System for key material in order to communicate with another actor. The request includes information identifying with whom the actor will communicate.
2. The CTM System checks trust relationships to see if it will allow communication between the identified actors.
3. If allowable, The CTM System sends key material to requesting actor.
4. The CTM System sends key material to second actor. (This step is optional depending on the type of cryptographic system used by second actor.)

Implementation Examples –

- New network sensor requesting key to communicate with visualization tool.
- SSCP slave device requesting key for communication with SSCP master device.

2.4.4 Expire Key Material

Goal – The key that has expired is stopped from being used by the actor and a replacement key is distributed to the actor.

Summary – The CTM System is responsible for key material and now must enforce key expiration. The system must notify and rekey the actor when their key material expires.

Actors – Application, system, or device.

Preconditions – The actor must communicate via a supported protocol to enable the request and delivery of the key material. The actor must have previously requested key material from the CTM System.

Triggers – Time threshold expires for key material.

Event Flow –

1. Key material lifetime expires.
2. The CTM System creates secure connection with actor.
3. The CTM System notifies actor of key expiration.
4. Actor starts Request Key Material use case (2.3.2) for replacement key (optional).

Implementation Examples –

- Certificate expires.
- SSCP master keys expire.
- ISA100.11a session keys expire.

2.4.5 Negotiate Trust Session

Goal – An actor from a third party is allowed to communicate on the CTM System, and necessary communication keys are distributed by communication with the third party's trust system.

Summary – Third party actors may need access to the environment being protected by the CTM System. This use case describes the process of negotiating the trust between the parties and distributing the necessary key material to the third party or visiting actor to allow this actor to perform work.

Actors –

- Application, system, or device.
- Third party trust management system.

Preconditions – None.

Triggers – Request by third party actor to communicate with actor managed by the CTM System.

Event Flow –

1. Third party actor requests access to actor managed by the CTM System.
2. As part of the request, the third party actor provides credentials (e.g., identifier, authentication information, and third party trust management system information).
3. The CTM System checks trust relationships and trust stance.
4. If relationship and stance checks are satisfactory, the CTM System communicates with third party trust management system to authenticate and define access rights of third party actor.
5. The CTM System sends necessary key material to third party actor to access local actor.
6. If necessary, the CTM System sends key material to local actor to allow communication.

Implementation Examples –

- Mobile Plug-in Hybrid Electric Vehicle (PHEV) infrastructure.
- Integrator/Vendor access to utility equipment.

2.4.6 Revoke Key Material

Goal – The key material used by an actor or actors is revoked.

Summary – The CTM System is responsible for managing key material for the actors that use it. At some point, the key material used by an actor may become un-trusted due to an event, (e.g., compromise or lost device). This use case describes the process a user would go through to revoke a key.

Actors –

- User.
- Application, system, or device.

Preconditions – None

Triggers – User initiates key revocation.

Event Flow –

1. User or actor invokes process to revoke key material (GUI or API).
2. The CTM System communicates with all actors using key material that the key has been revoked.
3. The CTM System archives key material and metadata that has been revoked.

Implementation Examples –

- Employee termination.
- Compromised system.

2.4.7 Configure Trust Relationship

Goal – A relationship is enabled between this CTM System and a third party trust management system.

Summary – The CTM System is designed to manage the trust for the entity. It is expected that external parties will intermittently require access to the entity's actors. This use case describes the process to configure trust relationships with third parties to enable automatic access for third party actors to communicate with the entity's actors.

Actors – User.

Preconditions – None.

Triggers – User manually triggers this use case.

Event Flow –

1. User configures trust relationship (GUI or API).
    a. Third party information.
    b. Access allowed for each configured trust stance.

Implementation Examples – Integrator/Vendor access to utility equipment.

2.4.8 Retrieve Key Information

Goal – User extracts key material or metadata from archived data.

Summary – The CTM System will be required to archive and store key material for auditing and forensic purposes. This use case describes the process a user takes to retrieve current or archived key information from the CTM System.

Actors – User.

Preconditions – Key material must be archived before it can be retrieved.

Triggers – User requests key information.

Event Flow –

1. User requests key information (GUI or API).
2. The CTM System authenticates user for access privileges.
3. The CTM System returns key information.

Implementation Examples –

- Forensic investigation.
- Regulatory audit.

2.4.9 Manual Key Material Request

Goal – User installs key material for actor that cannot communicate with the CTM System directly.

Summary – The trust management system must work with applications that were not developed to interface with it. This use case defines the process a user takes to use the trust management system with the external application.

Actors

- User.
- Application, system, or device.

Preconditions – None.

Triggers – User manually requests key material.

Event Flow –

1. User requests key material (GUI or API).
   a. Identifier of system that will use key material.
   b. Type of key material to be created.
2. The CTM System authenticates user.
3. The CTM System generates key material and stores it.
4. The CTM System delivers key material to user.
5. User installs key material into actor.

Implementation Examples – Any current system that was not developed to operate with the CTM System.

2.5 Delay Tolerant Centralized Security Management

Typical process control systems (specifically SCADA systems) within the energy utilize a network design with a centralized control station and geographically disperse outstations. These outstations house field equipment for monitoring and manipulating physical processes. The field equipment needs secure communication back to the control center, as well as secure communication within the outstation network. Also, maintenance personnel, both from external organizations under contract and within the utility, must communicate securely with field equipment for configuration and maintenance. Maintenance communication may be done remotely or from within the outstation.

Security applications that match the centralized architecture of the process control network integrate well with the processes and procedures in use today. Central creation of policies and control, while only deploying staff for intermittent maintenance, fits into the culture of the industry. Security mechanisms built around this type of architecture often require constant communication with a centralized service. This constant communication is a risk that is not acceptable for this environment. The backhaul communication medium between the control station and outstations cannot be assumed to be reliable. Even in the event of a communication failure of the backhaul network, the communication within the outstations and between maintenance personnel and field equipment must still be available. Therefore, security controls that are placed within the outstation environment must be able to operate in the face of communication delays and maintain autonomous operation.

Delay-tolerant centralized security management architectures are needed for process control networks. A delay-tolerant centralized architecture maintains a central control mechanism but has distributed autonomous agent nodes within outstations that can provide short spans of continued operation in the event of a communication loss. The agent nodes briefly store, or cache, the information going to and from the outstation to ensure that communication failure does not prohibit operation. When communication is restored, the agent nodes can resynchronize with the central control station and continue caching new information in preparation of another communication problem.

3.0 System Architecture

3.1 Architectural Design

The Cryptographic Trust Management System architecture is designed to accommodate the challenges and unique characteristics of process control environments. Process control networks place more emphasis on availability and reliability than do other more generic information technology networks. Therefore, the CTM has been designed with the assumption that the communication infrastructure between non-physically connected sites is unreliable. In addition to providing the functionality prescribed for the system, the Cryptographic Trust Management System is designed to reduce impacts on availability and operations as much as possible.

Figure 3.1 depicts the Cryptographic Trust Management System high level architecture. The diagram shows the interaction between facilities within the utility's process control network as well as with a third party entity that must interface with the process control equipment (e.g., an integrator or vendor). The various high level communication interfaces are captured to showcase how the architecture fits together and how it integrates into current process control system networks. The architecture depicted leans heavily towards SCADA-type infrastructure, but the CTM System architecture is designed such that it will accommodate other process control networks.

The control center facility houses the main functional components of the CTM System. The control center in a process control network is architecturally designed to control assets which are physically dispersed, either geographically or across disparate networks. The CTM System mimics this characteristic and was designed to centralize the trust management functionality of remote cryptographic assets. Therefore, the majority of the CTM System components are housed within the control center.

The backup control center facility is specified in the architecture to accommodate the failover requirements of process control systems. In the event of a critical emergency, the control center may become inoperable. Consequently, process control networks have stringent requirements for failover to backup control center facilities. The architecture of the CTM System is designed to maintain the failover requirements of the environment. The goal of cryptographic processes is to secure communication between only those authorized to communicate. Therefore, the addition of cryptographic processes, if done incorrectly, becomes a barrier to efficient failover of communication facilities. The CTM System was designed with this understanding. The architecture diagram (Figure 3.1) includes the backup facility to assist in describing how the system performs failover operations.

The remote station represents a disperse collection of assets that are physically separated from but monitored and controlled by the control center. As mentioned above, the design of the Cryptographic Trust Management System was created under the assumption that communication infrastructure between remote stations is unreliable. The diagram depicts the remote station to assist in describing how the interface will operate in the face of unreliable communication infrastructure and a high availability requirement.

In the figure, all items shaded orange are components of the Cryptographic Trust Management System. The remaining items are included as representative applications and entities within the process control environment. These applications and entities will be used to show the integration of the CTM

System into the process control network. The interfaces and functionality that the CTM System will provide to the applications will be described using these representative applications.

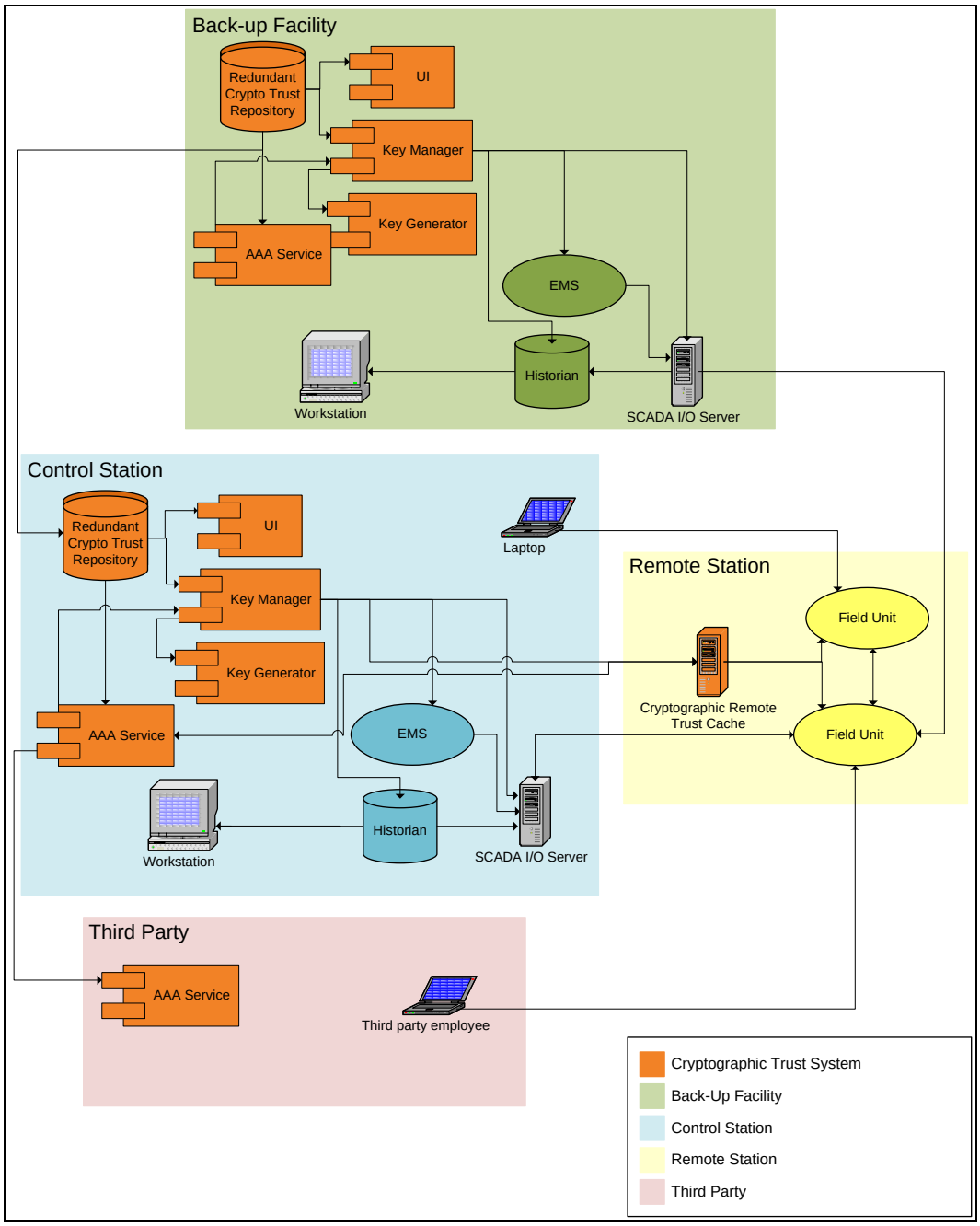

Figure 3.1. Cryptographic Trust Management System High-Level Architecture

The remainder of this section will consist of short descriptions of the Cryptographic Trust Management System components depicted in Figure 3.1. The descriptions in this section focus on the purpose of the components and their interactions. Sections later in this document will provide a high-level definition of the design specifications for the components and will describe how they operate and the functionality they will provide.

3.2 Cryptographic Trust Repository

The Cryptographic Trust Repository is the storage area for all cryptographic material managed by the system. All cryptographic keys, their associated metadata, and role definitions are securely stored in the Repository. The Trust Repository component is also responsible for maintaining the process control system device registration data necessary to perform key management functionality. All cryptographic policies are maintained by this component. These policies will be used by the Key Manager component to perform automated key management functionality and the Key Generator component to ensure proper cryptographic material is generated for applications using the system. All the information maintained by the Trust Repository is the basis for the audit and forensic reporting capability for the system. Therefore, this component is responsible for calculating and maintaining the metrics necessary to create forensic and audit reports.

The Trust Repository provides back-end storage for all of the other components. The data it stores represents the current trust state of the process control system. The stored information also represents previous trust states for a time period specified by regulatory requirements (e.g. three years). The User Interface component will be able to interface with the Trust Repository to configure policies, register devices, perform manual key management functionality, and retrieve reports. The Key Generator component will utilize the Trust Repository policies to generate cryptographic material which will then be stored by the Repository. The Key Manager component will distribute the stored cryptographic material to all necessary devices and will also utilize the stored cryptographic metadata to notify process control applications of key lifetime information due to configured policies. The AAA Service will use stored certificates during authentication processes and will write audit logs back to the repository.

3.3 Authentication, Authorization, and Accounting (AAA) Service

The AAA Service provides centralized authentication and centralized authorization services. All devices requesting access to cryptographic material or entities requesting access to applications or resources must first be authenticated to the AAA Service component. Authorization roles are defined and stored within the Cryptographic Trust Repository component. These roles are utilized by the AAA Service to authorize entity access to applications and resources. All authorization and authentication actions are logged by the AAA Service to support auditing and forensic activities.

The Key Manager component is the interface to all embedded devices and acts as the intermediary for authentication of devices for key management actions and for authentication and authorization of entities requesting access to resources. The Key Manager component utilizes the AAA Service component to authenticate end devices before performing key management processes. The Key Manager component also forwards authentication and authorization requests for entity access requests to the AAA Service.

The AAA Service also interfaces with peer AAA Services. The AAA Service is designed to accommodate the requirement of external organization staff needing access to process control equipment for configuration and maintenance. To reduce the operational burden of managing third party entities, the AAA Service component is designed to support roles that extend to other organizations. A trust negotiation is performed between peer AAA Services when an authentication and authorization request is received for a third party entity. Only if the trust negotiation process is successful is authorization granted to the third party entity. The third party trust negotiation process is not only provided to reduce the

operational burden but is also designed to increase the security verification of third party connections. Current best practice in an optimal situation merely provides authentication of third parties but no authorization services. In that situation, connection authorizations are only protected by contractual guarantees of notification in the event of personnel actions, such as position changes or terminations. The use of AAA Service in the Cryptographic Trust Management System design is to move from contractual protection to actual automated dynamic verification of third party entity authorization.

3.4 Key Generator

The Key Generator component provides a reliable cryptographically entropic random source for the generation of cryptographic material. This component will provide the ability to generate all of the commonly accepted and used cryptographic material such as different forms of symmetric and asymmetric keys and certificates. All cryptographic material in the system will be generated by the Key Generator component. The Cryptographic Trust Repository will interface with the Key Generator, on behalf of the Key Manager and AAA Service components, to generate the cryptographic material they need. The User Interface will interface with the Key Generator to generate cryptographic material needed for manual key management associated with legacy equipment. The Key Generator component will deposit generated cryptographic material into the Trust Repository component.

3.5 Key Manager

The Key Manager component provides services to automate the key management process. All future devices that support the Key Manager's service protocols will be able to request cryptographic material, be notified of cryptographic material expiration based on policy, and establish shared cryptographic material with other devices. The Key Manager is designed to be the interface to the Cryptographic Trust Management System with all devices requiring cryptographic material to operate. The goal is to create a centralized service that will provide key management operations with which all devices can interface. This will relieve the operational burden of manual key management processes, eliminate separated pockets of cryptographic material managed by segregated application domains, and remove the burden of devices properly and efficiently creating their own cryptographic material.

The Key Manager component will interface with the User Interface component to enable manual key management operations, such as forcing a key update, generating key material for legacy systems, or expiring/revoking cryptographic material. The Key Manager will interface with the Key Generator component to request the creation of cryptographic material when it is needed. The Key Manager component will communicate with the AAA Service component to authenticate devices. Critical requests, such as a request for cryptographic material update or to retrieve current cryptographic material, must be done securely. Therefore, all such requests must be authenticated to ensure that requesting devices are only allowed to retrieve cryptographic material to which they should have access.

Authentication will require a device registration process to ensure that devices have the necessary credentials to operate with the Key Manager system. For further discussion, please see “Device Registration and Provisioning” (14.0).

3.6 User Interface

The User Interface component provides the human interface to the Cryptographic Trust Management System. The User Interface will provide views on the data within the system as well as functionality to force processes to run. It will also provide a real-time status view that will offer the ability to see all of the cryptographic material metadata (e.g., percentage of lifetime, associated devices, etc.) and alerts of any policy violations or system errors (e.g., cryptographic material not updated in proper timeframe, outstanding key update needed, failed authentication, communication failures, device unreachable, etc.). The User Interface will also provide the administrative portal to register devices and define which devices should share cryptographic material, roles and user access, and all other system configuration capabilities. The User Interface will also provide the ability to retrieve archived keys and manually-initiated key management processes. Also, the User Interface will provide the ability to generate reports for audit or incident response and forensic activities that detail policy violations and errors in a defined time window.

The User Interface communicates with the Cryptographic Trust Repository to retrieve the data necessary to create the real time view for system operators and also to create the reports for auditors and incident response/forensic investigators. The User Interface utilizes the Key Manager to enable the manual initiation of key management processes.

3.7 Cryptographic Remote Trust Cache

As previously stated, it is assumed that the communication path between the control center and remote stations will not be reliable. Therefore, the Cryptographic Remote Trust Cache component is designed to cache cryptographic material and metadata to enable the remote stations’ cryptographic services to continue functioning, for a time, in the event of failed communication with the control center. The Remote Trust Cache operates as the gateway for all cryptographic trust communication into and out of the remote station. When a device requests a new key or an entity attempts to log into a remote station device, the communication will first go to the Remote Trust Cache component which will then communicate back to the control center. During these communications, the cryptographic material will be passed back through the Remote Trust Cache and then forwarded to the devices to continue operation. The Remote Trust Cache will store these credentials and the associated cryptographic material at that time. If another request comes for the stored material, the Remote Trust Cache will not communicate back to the control center but will perform the services itself, while the stored material is still valid. In the event of communication failure, the remote station will be able to continue operation because the needed cryptographic material is stored locally.

4.0 Cryptographic Trust Repository Component

4.1 Architectural Design

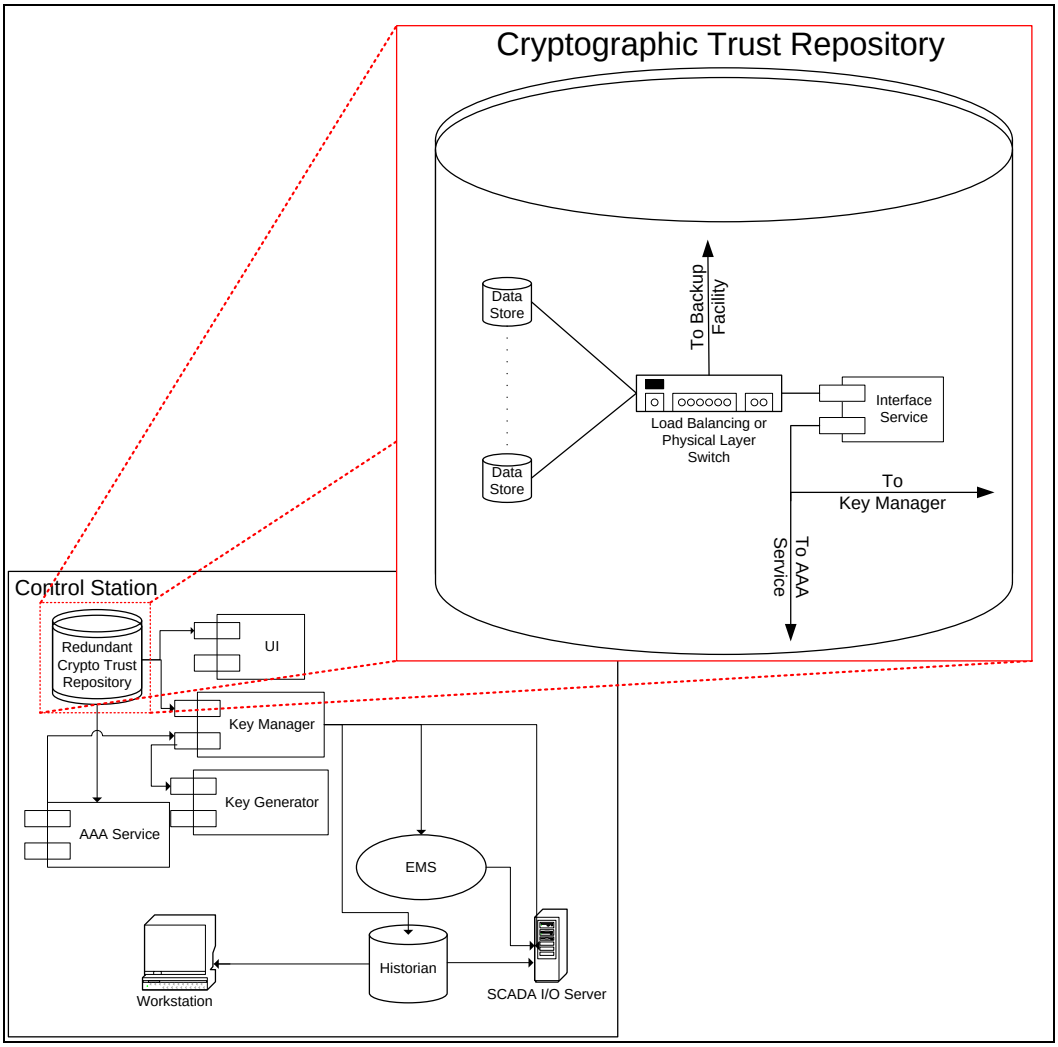

Figure 4.1. Architectural Design of the Cryptographic Trust Repository Component

The Cryptographic Trust Repository component provides secure storage for cryptographic material, audit logs, policy and role definitions, and third party trust configurations. The Trust Repository's main functions are securely storing all of the data related to the Cryptographic Trust Management System, archiving old data for retrieval if necessary, and providing redundant reliable service and failover mechanisms to a back up control facility to maintain operations in the event of a loss of the primary control station. The Trust Repository is the backbone for the rest of the CTM System. All of the other services in the system rely upon it to provide policies, guidance, and data to support operation.

The Trust Repository must be deployed in a redundant hardware and software environment to meet the necessary real-time data store and certificate status information requirements. Commodity hardware and software can be used to accomplish this goal and scale to the tens of millions of keys. The required components are:

- Database or LDAP directory servers for the data store
- Adequate storage such as a RAID, SAN, or NAS
- Load balancing or physical layer switch for redundancy and failover
- Health check software to ensure the repositories remain in sync
- Backup system(s) to ensure repository information can be recovered
- A process to ensure cryptographic keys for the repositories and other failover components are replicated
- A method to identify connections to other repositories/organizations (similar to the manner in which DNS can reference other DNS servers)
- A method to enable secure communication to the repository
- A method to enable secure storage of cryptographic material
- A method to store previous cryptographic keys for a user/device.

4.2 Redundancy

Redundancy is required within the Cryptographic Trust Repository to ensure that critical cryptographic information is highly available to all other system components. Without a redundant environment, the Cryptographic Trust Management System would have a single point of failure, potentially rendering the entire system susceptible to a loss of connectivity. While redundancy can be considered in terms of local and remote environments, the techniques used to replicate data can be designed to operate in either manner with similar techniques. Two primary design options are available to replicate data for a redundant environment. The first technique depends upon options available in the repository itself, and the other technique is based upon various infrastructure components. The selection of a redundancy method should be tightly coupled with the selected failover method to avoid the need to reconfigure client access to necessary data.

The first redundancy method is for the database or directory used as the repository to initiate the real-time backup or copy of cryptographic keys. For example, consider an environment where the primary repository provides all cryptographic functions for the various client components. This primary repository utilizes a transaction-based approach to replicate changes to the repository as they occur to the redundant repository. When a new key is generated for a client, the primary repository stores the information locally and sends a copy to the backup repository. Writing data to both the primary and backup repositories could also be performed by the Key Manager component. This method would require configuration of both data stores on the Key Manager, adding a degree of complexity as well as a potential performance burden on the Key Manager server.

The second option is to utilize physical infrastructure components and filtering to provide the necessary communication for storing data on redundant servers. One example is to utilize physical layer switches. In this model, data destined for the primary data store is replicated to the backup data store(s). However, only the primary data store responds to a request for information. This option removes the need for complex configuration settings and also reduces the potential performance impact on the primary repository. In addition, a single appliance can perform both replication and high availability failover functions.

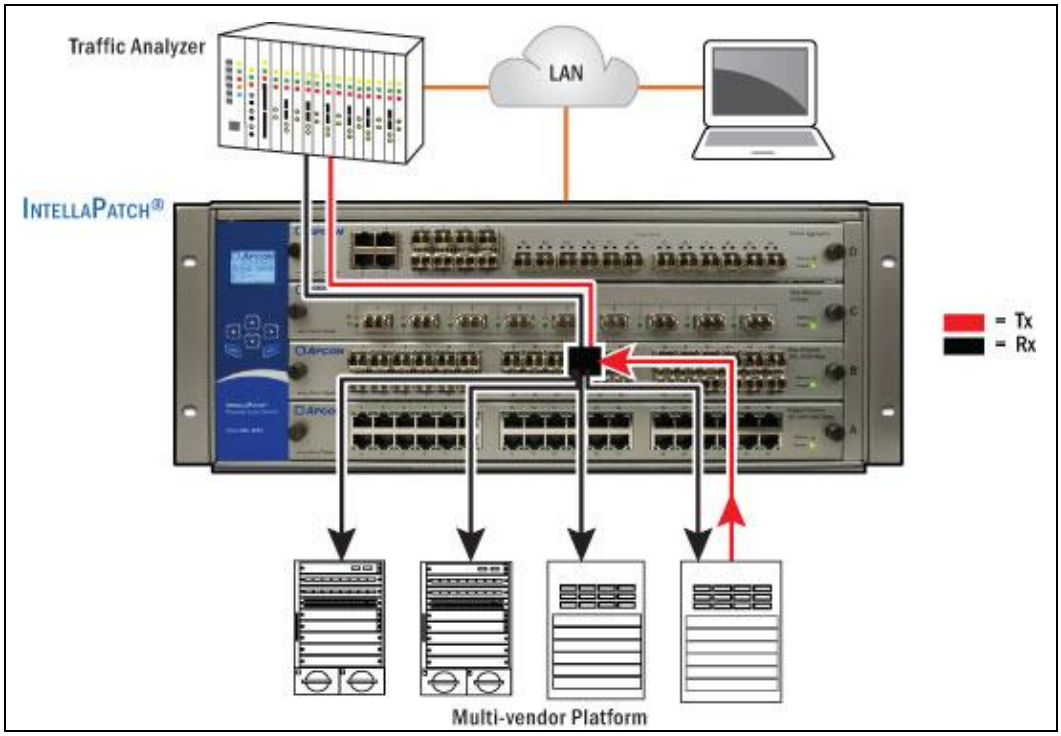

Figure 4.2. Physical Layer Switch

4.3 Failover

Designing and deploying a failover environment can be performed using a variety of techniques. Common methods include round-robin, primary and live backup, and maintaining a cold spare server that is ready to power on when the primary fails. For cryptographic trust applications however, the failover environment must be configured to support ease of configuration, instant failover, and no reconfiguration of end devices and/or applications. The failover system must also ensure the same master cryptographic key is installed on both the primary and backup repositories in order to simplify secure communication. For this reason alone, a cold spare is not a viable solution. Also, the complexities associated with managing bi-directional repository updates eliminate the round-robin failover implementation as an option. The most viable solution is to utilize the primary and live backup (hot spare) approach.

To accomplish this approach, load balancing switches or the physical layer switch described in the previous subsection can be deployed. Both solutions allow a single IP address that is known to all other devices/applications to be used for the repositories. The switch will monitor connectivity to the primary repository based upon private IP address and destination port number. If the service is not available, the

switch will direct requests to the backup repository. A more sophisticated solution will actively monitor the health of each repository and update switch settings accordingly.

During a failover, the devices in the backup control station will require the cryptographic material to communicate with the devices in the outstations that already have cryptographic material. With the live backup, the repository in the back up control center will perform the synchronization of cryptographic material when updated in the primary system. The devices in both the primary and backup facilities will be configured to use the same cryptographic material which resolves to the same unique identifier in the Cryptographic Trust Repository component. When the cryptographic material for the device in the primary control station is updated, the Trust Repository notifies the Key Manager in the back up facility to update the back-up facility devices. For more discussion on the synchronization of cryptographic material, see "Key Manager Component" (Section 6.0) and "Cryptographic Remote Trust Cache Component" (Section 8.0).

4.4 Repository Data

The Cryptographic Trust Repository is the main storage facility for all data used, processed, and created by the Cryptographic Trust Management System. The Trust Repository contains the cryptographic material and its associated metadata, and the policies that apply to cryptographic material and third party operations, roles and access rights, and audit logs.

4.4.1 Cryptographic Material and Metadata

Cryptographic material represents symmetric and asymmetric keys used in security applications and certificates and other credentials used for identity management. Cryptographic material is generally a blob of data that meets some mathematical requirements or is a data structure that holds the blob of data and associated information. All cryptographic material must meet the algorithm or standard for which it is used. For a list of supported types of cryptographic material, see "Key Manager" (Section 6.0). Metadata is data associated with and that describes the cryptographic material or is an attribute of the cryptographic material. The following table defines some expected cryptographic material metadata.

Table 4.1. Expected Cryptographic Metadata

Attribute	Description
Version	Indicates the version of the certificate or key.
Identifier	A unique identifier for cryptographic material for reference during key updates.
Lifetime	A value for how much time is left in the cryptographic material's lifetime.
Subject	The distinguished name of the user or device to which the certificate or key was issued.
Link	A link to a policy or device associated with this cryptographic material.

4.4.2 Policies

Policies are rules guiding the operation of the system. Policies define what and how something should be done or not done. Policies are created by the user and can be per application or per communication session. Policies are applied to cryptographic material, roles, and third party trust connections. The following table describes some examples of variables in a policy.

Table 4.2. Policy Variations

Variable	Description
Cryptographic Strength	The required cryptographic strength of the cryptographic material associated with the policy.
Validity Period	How long cryptographic material is valid before it must be updated. A soft expiration validity period is a user-configurable percentage of the hard validity period. This gives the device early notification of the need to update keys so the change can be made during a time with minimal impact on operations.
Trust Evidence	Different forms of requirements placed on a third party trust negotiation before authorization of a third party employee access request is provided.
Access Level	The level of access allowed by a role. This depends on the types of devices but could be engineering access or maintenance levels of access.
Cache Ticket Period	The amount of time the Kerberos style ticket is valid and Remote Trust Cache local authentication will be allowed.

4.4.3 Roles

The CTM System is a role-based access control system. A role defines which devices and services an entity or employee has rights to access. Roles are attached to credentials to provide access privileges to entities. Each entity must have at least one role attached to it. During authentication, the role is used to determine if an entity should be authorized to access a device. Roles are a method of simplifying access privileges by grouping them around job functions.

Third party employees often need access to devices for maintenance or integration purposes. Industry defined and accepted role classifications will be used to perform third party trust negotiation authentications of third party employees. Third party employees can only belong to one of the pre-defined role types. Any of the pre-defined role classifications can be configured and associated with any third party trust connection. If the third party authenticates that one of their employee's credentials are valid and belong to one of the configured predetermined roles, the configuration of the role will be used to authorize access to devices. For more discuss of third party trust negotiations, see "AAA Service Component" (Section 7.0).

4.4.4 Audit Logs

Audit logs are logs of events that have occurred during the operation of the system. Logs can be used to troubleshoot problems or to discover the root and sequence of events for a malicious attack. The Cryptographic Trust Repository component will log events associated with triggers that are described in the next subsection. The AAA Service, Key Manager, and User Interface will all log events associated

with their operation to the Trust Repository component. The following table describes some possible events that could be logged.

Table 4.3. Example of Events Logged by the Trust Repository

Event	Description
Key Expiration (hard or soft)	A time threshold according to a policy has been reached.
Key Update Request	A device has sent a request to update cryptographic material.
Authentication (successful/unsuccessful)	A device or entity has performed an authentication.
Login to User Interface (successful/unsuccessful)	A user has logged into the User Interface component.
Configuration Change	A configuration change has occurred to a policy, role, or attribute in the User Interface component.
Archived Data Accessed	Historical data has been accessed from an archive.

4.5 Triggers

The Trust Repository will notify the entity (end device, user, Cryptographic Remote Trust Cache) when meta-data associated with the key warrants a key update. The trigger can be related to any number of events including, but not limited to:

- A cyber security incident
- A key that reaches a percentage of its lifetime (soft key expiration)
- An expired key
- A change in approved algorithms
- Recovery from a failed component
- The return of a device from maintenance.

The notification will be initiated by the Trust Repository and communicated through the Key Manager to the necessary device. Triggers related to updating authentication credentials will be flagged and the new credentials will be exchanged during the next authentication by the device or entity.

4.6 History

The Trust Repository represents the current trust state as well as the historical state of the process control system. Regulatory requirements in some control system sectors require data relating to an event be stored for three years. If the event data is secured cryptographically, the cryptographic keys must also be available for the same period of time. The Repository must be able to scale to support storing historical data as required by regulation.

4.7 Security

When communicating cryptographic key material to the Trust Repository, the communication must be encrypted. In addition, the data stored within the Trust Repository will be encrypted using approved algorithms. This approach ensures that data is secure both during transit and at rest. In addition, the Trust Repository utilizes host-based security mechanisms (e.g., strong access control, IDS, hardened OS, real-time monitoring, etc.) to ensure the security of the data. While not in the scope of this documnet, good physical security practices such as a six-walled enclosure, access controls, and video security should also be deployed and monitored. Due to the large amounts of cryptographic material stored by the Trust Repository, it represents a high risk target for an adversary and must be secured accordingly.

5.0 Key Generator Component

5.1 Architectural Design

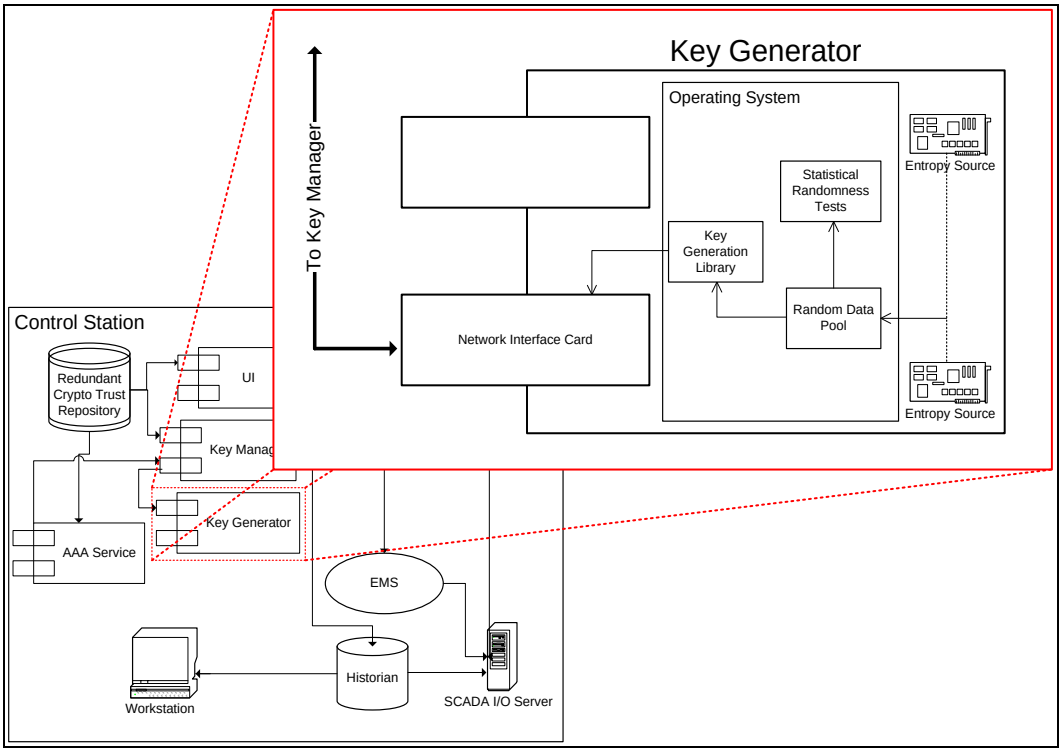

Figure 5.1. Architectural Design of the Key Generator Component

The function of the Key Generator component is to centralize the generation of cryptographic material and deliver that material to requesting systems. The Key Generator will produce all of the cryptographic material for the entire process control network. The Key Generator component is designed with this functionality for two reasons: it is difficult for embedded devices to generate quality entropic data for key generation; and to ensure that the cryptographic material used across the network is created correctly and in a manner following all policies.

The Key Generator is a computer with hardware and software components. At the lowest level, the Key Generator utilizes a network interface card (NIC) and multiple hardware entropy sources, as needed for scalability (see Section 5.2 (Entropy) for further discussion). At a higher level, the component requires an operating system that has the drivers necessary to communicate with the NIC and the entropy hardware and software to provide the functionality of the component. The Key Generator component has a Statistical Randomness Test application and a key generation software library. It also requires other common hardware necessary for a computational device, such as a CPU and memory. However, a writeable long term storage device such as a common SATA hard drive is not necessary because the system will not need to store cryptographic material locally; it simply delivers it and forgets it. In the absence of a hard drive, the operating system should be burned onto a storage device as firmware.

The data flow of the system will start at the NIC then proceed from entropic sources through the system back out to the NIC (see Figure 5.1). The Key Generator system works as a service and starts processing after receiving a request for cryptographic material. The Key Generator Library operates as a service daemon that takes requests from the NIC. The Key Generator Library pulls the necessary data from the random data pool to generate the type of cryptographic material requested. The random data pool is fed by the variable number of entropy sources. As data is pushed to the random data pool, the statistical random test application monitors the data but does not remove it from the pool. It then runs a series of tests to determine if the entropy sources are still operating correctly. If they are not operating correctly, the statistical random test application will stop the Key Generator Library from delivering cryptographic material and instead return an error status. After stopping the delivery of cryptographic material, the statistical random test application starts sending a notification of the failure. If the entropy sources are still operating correctly, the Key Generator Library will craft the requested cryptographic material from the random data and will deliver it to the requestor via the NIC.

The Key Generator is designed as its own self-contained component. Therefore, implementations of the Key Generator component could be utilized within systems other than the Cryptographic Trust Management System that desire a centralized key generation service.

5.2 Entropy

Entropy is the foundation of all cryptography. It is the amount of randomness in a system. Without entropy, all the fundamental mathematical properties supporting cryptographic operations break down. In cryptography, entropy applies to the random data used as input to the many different cryptographic algorithms available. Whether it is generating a key, a nonce, a salt, or any other random data structure, large amounts of entropy are required. Cryptography requires that the adversary cannot discern the output of the operation, even if the cryptographic algorithm is known. Therefore, the input data for the operation must be random and undiscoverable. If an adversary can start with the same algorithm and inputs then the output will be the same.

Due to the importance of entropy in generating cryptographic material, the Key Generator component has a requirement to generate and use entropic data. A requirement for all key material is that it contains a high percentage of entropy. If a key is easily guessed, it loses its cryptographic value. For that reason, the Key Generator component relies heavily on the entropic data to perform its primary functions. The sources of entropy must provide the necessary amount of data for the Key Generator to fulfill its functions.

The Key Generator component must scale to meet the needs of the Cryptographic Trust Management System. In general, this will be based on the number of devices the CTM System is supporting, with the understanding that some devices will require more cryptographic material than others. However, the component can be implemented to support either a range of devices specifically targeting a sector of systems, or in a modular fashion allowing the addition of entropic sources to meet the needs of larger process control systems. Implementations can be executed as necessary, but the design laid out here does not put strict requirements on how much entropic data should be supported. This design document merely requires that true entropic sources are implemented in the Key Generator component and that they are solely used in performing the component's functions. An implementation is only constrained by the choices and quantity of hardware available.

Entropic sources can be based on numerous natural inputs. Most modern operating systems provide a service for collecting systemic entropic sources and combining them into a single pool of entropic data. These sources are generally based around user input devices, process metadata, disk access metrics, and memory characteristics. These sources are not by themselves one hundred percent entropic, but together they provide a sufficient amount of entropy for general operating system use. Other possible sources of entropy include audio and visual input, such as via a microphone and camera, or electromagnetic input, such as radio or light. However, these are not optimal sources of entropy. A high-quality entropic source cannot easily be manipulated by an adversary to produce less entropic data. All of the above sources of entropy utilize an external input as their source and would require physical security measures to ensure that the external inputs are not tampered with or biased. The best current sources of entropic data are from utilizing circuit thermal noise and free running oscillators. The major chip makers, Intel®, Advanced Micro Devices, and VIA™, provide implementations of these entropic sources. There are also security companies that will design and implement chips for providing entropic data to specifically designed systems. It is not important to this component which entropic sources are selected. Any of the choices discussed here or found elsewhere that meet the requirements of the randomness tests subsection (5.3) will suffice.

The design of this component is required to meet the scaling demands of a large AMI system. In order to meet this requirement, the hardware of this component must be implemented such that many of the entropic sources are stacked and pooled into one data stream that the software generating the keys can access.

5.3 Statistical Randomness Tests

The randomness of the entropic sources must be tested to ensure that they are statistically acceptable to be used within a cryptographic system. All random number generators are not created equally and their acceptability depends upon their desired function. Simulations often need random data input to imitate chaos in a system, and cryptography needs random data as described in the entropy section. However, random number generators that are acceptable for use in simulations may not meet the stringent requirements needed for cryptographic use. Therefore, random number generator test suites have been developed to test for applicability for cryptographic use.

At the time this document was written, the industry standard test suite is provided by NIST SP800-22. As part of SP800-22, NIST provides a software package, written in the C programming language, implementing their statistical test suite. The entropic sources selected must be able to pass the statistical test suite independently and combined. Passing the NIST SP800-22 test suite provides reasonable assurance that the random data produced by the tested entropic sources do not have discoverable biases that would allow discovery or estimation of the forthcoming sequence of data.

The NIST SP800-22 test suite is written and developed as an offline process that happens once at the time of system implementation. This single set of tests does not, by itself, provide adequate amounts of assurance for system operation. Entropic sources can fail silently allowing bias or repeated sequences into the stream of random data used by the Key Generator system. For that reason, it is crucial that some form of continuous testing is performed during system operation to ensure that the random data stream hasn’t malfunctioned or been compromised. At a minimum, the system must perform the continuous random number generator test defined in the FIPS 140-2 specification. Additional tests are recommended because

the tests outlined by FIPS 140-2 are rudimentary and may miss some loss of entropy. However, it is not a requirement of this design document because additional testing will require more expensive hardware and additional performance testing to ensure that the entropy testing does not impact the delivery of cryptographic material to the rest of the Cryptographic Trust Management System.

5.4 Cryptographic Material Support

At a minimum, the Key Generator shall support the generation of cryptographic material for the following cryptographic algorithms:

- Public Key / Key Agreement
  - RSA
  - DSA
  - ECDSA
  - DH
  - ECDH
  - X.509 v3
- Ciphers (in modes ECB, CBC, CFB, OFB, CTR, CCM)
  - AES
  - RC5
  - RC4
  - RC2
  - 3DES
- Digests
  - MD5/HMAC-MD5
  - SHA-1/HMAC-SHA1
  - SHA-256/HMAC-SHA256
  - SHA-512/HMAC-SHA512

A well-established cryptographic library will be used to enable the functionality required in this section. Neither the library nor the programming language selected is important because they are all tools to achieve the end solution. They all have their own dogma and quirks and are essentially interchangeable. However, using an established library over creating one for this component has the benefit of knowing it has been tested in real-world environments and numerous vulnerabilities have already been discovered and repaired.

5.5 Operating System

The choice of operating system is not significant to the operation of the Key Generator component. However, the operating system selected for an implementation of this component must work with the software library and entropy sources selected.

5.6 Interface

The Key Generator interfaces with external entities via a Hypertext Transfer Protocol (HTTP) web service. The Key Generator web service will be a simple service that allows the Cryptographic Trust Repository or any other application to request any of the supported types of cryptographic material. The request messages are defined such that the necessary data to create the cryptographic material is included. For example, a request for an X.509 certificate would include items such as the issuer information, validity date, subject information, and all the other information contained in the X.509 certification structure.

Upon receipt of a request, the web service will interface with the key software library to fulfill the request. This will initiate the retrieval of entropic data and the generation of the requested cryptographic material. When the software library has completed processing the request, the generated cryptographic material is sent to the web service. The transaction is complete when the web service delivers the cryptographic material to the requesting application and closes the connection.

The web service is secured by utilizing the lower level Transport Layer Security (TLS). TLS is a transport layer protocol which must establish a connection before the web service requests will be allowed. TLS will be configured for mutual authentication so that both communicating parties must authenticate each other before establishing a communication session. TLS message encryption and integrity services ensure that once a connection is established, the exchanged key material is secure during transmission.

5.7 Security

The security design of the Key Generator is dictated by its function. Since the Key Generator provides a service for generating and sending, not storing, cryptographic material to another component, the security mechanisms are designed tightly around those aspects. The overall component is designed to be FIPS 140-2 level 1 compliant. Any additional security mechanisms, such as tamper evident hardware, are up to the implementer. The Key Generator component should be housed within a physically secure location and should only interface with other components of the CTM System. Therefore, it does not require the RBAC requirements of FIPS 140-2 level 2.

As discussed previously, the choice of operating system is not crucial. Since this component does not require a lot of services, the operating system must be reduced to the minimum level of functionality that is needed to enable the functionality described in this section. The Key Generator component could be created as an appliance with headless, without a visual display mechanism, operation if desired. All network interface capabilities shall be disabled unless they are needed for operation of the key delivery portion of the system.

The software library used to generate the cryptographic material must be FIPS 140-2 compliant, and certified if possible. FIPS 140-2 testing ensures that the algorithms were implemented correctly and reduces the likelihood of the presence of vulnerabilities affecting the generation of the cryptographic material.

The entropic sources selected should pass the NIST SP800-22 Statistical Test Suite. Also, the component should perform the continuous random number generator test as defined in the FIPS 140-2 specification. Entropic hardware can fail silently, no longer providing the assurance of a cryptographically secure system. The continuous test defined by FIPS 140-2 provides an acceptable level of assurance that the entropy in the system is still at a sufficient level to be secure. Other statistical tests can be added to provide more assurance that the entropic sources are operating correctly. Some of the statistical tests can be slow and resource-intensive, which may require more powerful hardware to be supported without impeding delivery of cryptographic material in a near real-time manner.

The network interface must be strongly secured because it provides the cryptographic material delivery mechanism. Cryptographic material must be kept confidential or it loses its security value. Therefore, the network interface must be strongly secured to prevent the unauthorized discovery of cryptographic material. TLS configured with mutual authentication will provide the security mechanism to protect the communication with external systems. This will provide the assurance that only the appropriate system components will be able to connect to the Key Generator.

6.0 Key Manager Component

6.1 Architectural Design

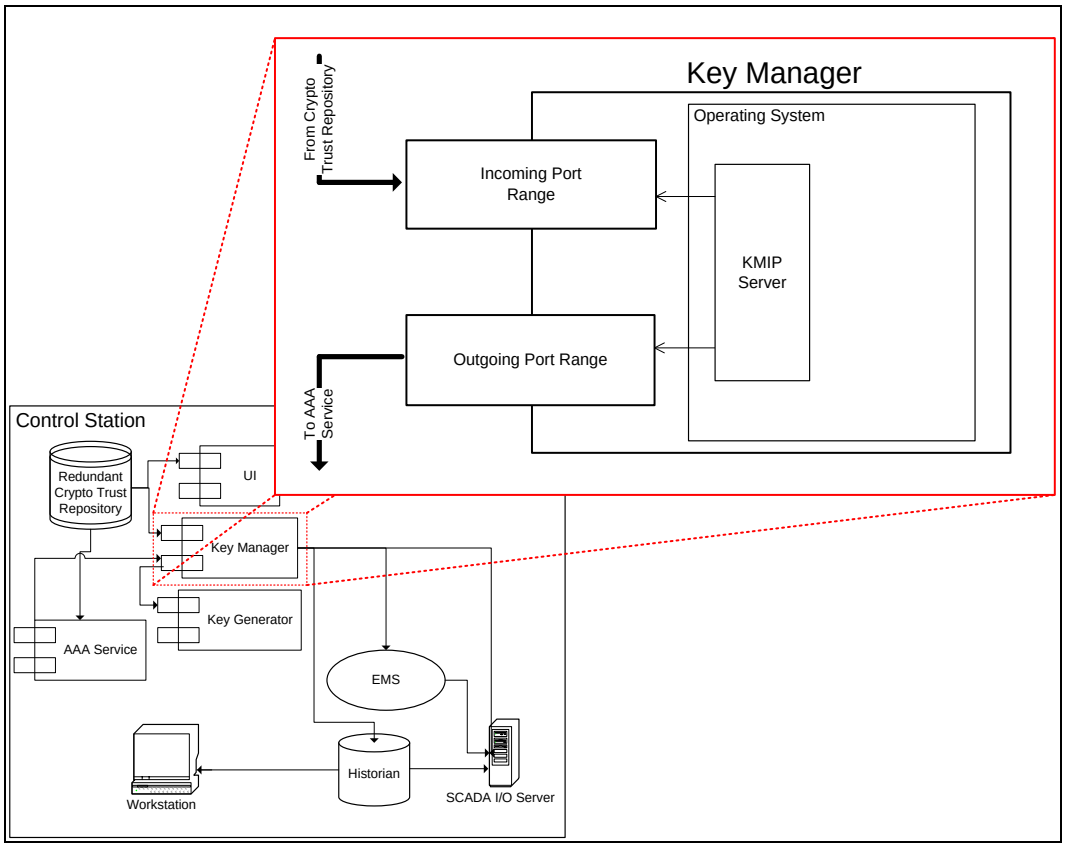

Figure 6.1. Architectural Design of the Key Manager Component

The function of the Key Manager component is to be the interface with all devices when performing automated key management functions. The Key Manager provides services to request new or in-use cryptographic material, send notifications of key expiration and lifetime percentages, and enable the distribution of keys across application domains and between different vendor equipment. The services that the Key Manager provides are not supported by the equipment in use today. But the promise of this technology is that it will reduce the operational burden of managing keys so that technical security controls can proliferate throughout critical infrastructure.

The Cryptographic Trust Management System is designed with a stop gap solution to the initial lack of device support. The CTM System provides key management procedures for creating and managing keys that are manually distributed. When the keys need to be updated, instead of performing the automated processes described in this section, the CTM System will notify security personnel so they can perform the manual update. For more information on the manual key management process, see “User Interface Component” (Section 9.0).

The Key Manager component is a Key Management Interoperability Protocol (KMIP) server. KMIP follows a client-server model. Therefore, the Key Manager component will operate as a server. A KMIP server is an application server that operates a portion of the KMIP. The underlying server technology and programming language selections for implementing the KMIP server are not important. Nor is the operating system on which they run. The server will require the use of TCP ports for the receiving and sending of KMIP messages. Depending on the scale of the control system, the Key Manager component could run on the same hardware as the Cryptographic Trust Repository or it could run on its own hardware. If the resources of the server are such that the two components operations do not impact each other then utilization of a shared resource is acceptable. The Key Manager component simply requires commercial off the shelf (COTS) server hardware to operate.

KMIP defines the cryptographic material with its associated metadata for a set of objects that are stored by KMIP servers. The Key Manager component does not store any of these objects. The responsibility of storing all of the CTM material belongs to the Trust Repository component. When the Key Manager needs access to cryptographic material it communicates with the Trust Repository. Also, the Key Manager will log all events to the Trust Repository to populate the reports that the UI component is responsible for creating.

6.2 KMIP Protocol

KMIP is an Organization for the Advancement of Structured Information Standards (OASIS) standard available for public review to enable interoperability between systems performing key management functions. KMIP was selected as the protocol to enable the key management functionality of the Cryptographic Trust Management System because it is an open standard that is being developed by a consortium of industry corporations and because it has most of the desired capabilities to automate key management among embedded devices. The only way the CTM System can become a successful system is if vendors choose to develop equipment that supports the protocols used by the system. Selecting an open protocol that already has some industry acceptance increases the likelihood of vendor adoption compared to creating a new protocol or utilizing a closed protocol.

While the current profiles of KMIP do not exactly meet all of the requirements of the CTM System the protocol does provide the components necessary. The OASIS KMIP work group has developed a document with conformance profiles. The CTM System most closely resembles the Basic Symmetric Key Store and Server KMIP Profile but it does not fit perfectly. The KMIP profile was designed more for a centralized IT-style key management system which allows for the registration of key material to the store and does not provide automated key management functionality. An explanation of how to specify a profile is also in the document therefore a new profile must be developed for this system. As the designs in this document are reduced to practice and tested in the future, it is recommended that a profile is documented and submitted to the OASIS KMIP work group for inclusion.

The KMIP specification defines a list of attributes (i.e., what has been called cryptographic metadata in this document) that can be associated with key material. KMIP uses this metadata to operate. The Cryptographic Trust Repository component has the responsibility for storing all of the metadata utilized by the KMIP. The Key Manager component will retrieve the metadata it needs to process the requests it receives.

KMIP provides a list of functional message types. These messages provide the capability to register cryptographic material, retrieve cryptographic material and metadata, and even for the server to send data to clients. The specification allows implementations to only support the subset of functionality required. Since the CTM System internally generates all cryptographic material via the Key Generator component, the functionality for external entities registering cryptographic material with the system via KMIP is unnecessary. Also, the CTM System is responsible for internally maintaining the cryptographic material policies and therefore, clients have no reason to request or receive metadata through the KMIP. Following is a list of the KMIP message types supported by the Key Manager component.

6.2.1 GET

The GET message is a request from a client to retrieve cryptographic material. From a client's and the Key Manager's perspective, there is no difference between a request for current, valid cryptographic material and updating to new cryptographic material. The Cryptographic Trust Repository will request new cryptographic material when a GET request is received and a cryptographic material expiration threshold has been reached.

6.2.2 NOTIFY

The NOTIFY message is a server notification to a client. In the CTM System, it is used to notify devices of cryptographic material expiration thresholds being reached. The system supports soft and hard expiration thresholds. A soft threshold is a configurable percentage of the key lifetime to allow a device sufficient time to update a key when it will be the least impactful to operations. A hard expiration is the threshold when a cryptographic material policy requires an update. Soon after a hard expiration the device will be flagged as violating policy and the security administrator will be notified.

6.2.3 PUT

The PUT message is for sending cryptographic material from the KMIP server to a client. The PUT command is only utilized by the CTM System to prevent synchronization problems. The CTM System manages cryptographic policy but does not enforce it. Therefore, it will not force cryptographic material updates on devices due to key expiration, but will instead rely on devices updating keys for themselves after notification. However, when devices share cryptographic material it must stay synchronized. If one device updates the cryptographic material the other devices sharing it will need to as well. Therefore, the PUT command will be used in this situation to send the newly created cryptographic material to the other devices in the group.

6.2.4 GET ATTRIBUTES

The GET ATTRIBUTES is only supported by the Key Manager and the Cryptographic Remote Trust Cache components. Devices are not allowed to request cryptographic metadata. The GET ATTRIBUTES message is utilized by the Remote Trust Cache component to retrieve the devices associated with cryptographic material from the Key Manager component. This functionality is needed to cache cryptographic material to allow for continuity of operations in the event of lost backhaul communication.

6.3 Registration Requirements

In order for the CTM System to manage the cryptographic material for devices, it must have a way to associate cryptographic material belongs to which devices. Therefore, during device provisioning a unique cryptographic identifier is created by the CTM System for each piece of cryptographic material required by the device. The identifiers are installed in the device so the device can communicate to the CTM System which cryptographic material it is requesting. Note that a single device may require multiple cryptographic keys. Within the CTM System devices are added as attributes, or metadata, to cryptographic material to keep track of associations. For a more comprehensive discussion of the provisioning and registration processes see “Device Registration and Provisioning” (Device Registration and Provisioning).

6.4 Cryptographic Material Update

When a device needs new cryptographic material, either because of expiration or loss of state (e.g., power cycle), the device will send a GET request with the cryptographic material’s unique identifier to the Cryptographic Remote Trust Cache. Either the Remote Trust Cache will have a current key and send it to the device or will forward the request to the Key Manager component. The Key Manager component will communicate with the Cryptographic Trust Repository to either retrieve current cryptographic material or to create new cryptographic material. The Key Manager will send the cryptographic material to the Remote Trust Cache component, which will cache it for a configurable time period, and then it will be forwarded to the device.

To prevent synchronization problems, the CTM System will send the cryptographic material to the rest of the devices associated with the cryptographic material. This responsibility falls upon the Remote Trust Cache component. Upon receiving a GET request or a NOTIFY message, if the Remote Trust Cache component does not have any cached information for the cryptographic unique identifier in the message, it will send a GET ATTRIBUTES KMIP request to the Key Manager component to discover all of the devices associated with the cryptographic material. The Remote Trust Cache will send PUT commands with the new cryptographic material to all of the rest of the associated devices to prevent synchronization problems from impacting operational communication.

6.5 Cryptographic Material Expiration

The Cryptographic Trust Repository maintains the lifetime of cryptographic material based on the date of creation and the governing policy’s expiration time thresholds. The Cryptographic Trust Management System also supports an optional soft expiration that allows devices to update cryptographic material when appropriate to minimize the impact upon operations. The CTM System does not force cryptographic material update because its design is governed by the concept of minimizing the impact of operational performance. Therefore, it is considered better to provide a device an update period before the cryptographic material lifetime expiration so that it may continue operation until an appropriate time to update with the least impact on operations. The soft expiration is designed to minimize the need for policy violations.

When a cryptographic material expiration threshold is reached, the Trust Repository will initiate the expiration process. The Key Manager will send a NOTIFY KMIP message to the Cryptographic Remote

Trust Cache components that are in the outstations with the devices using the cryptographic material. The Remote Trust Caches will synchronize which devices go with the cryptographic material in the message if necessary and will send NOTIFY messages to all of the devices. As mentioned above, it is up to the devices to find the right time to perform the cryptographic update.

6.6 Cryptographic Material Revocation

The Cryptographic Trust Management System provides the capability for a user to revoke cryptographic material used by devices. Since the CTM System does not enforce policy, in reality the revocation is merely a hard expiration notification. See "User Interface Component" (Section 9.0) for a description of the cryptographic material revocation process and see "Cryptographic Material Expiration" (Subsection 6.5) for further discussion of the hard cryptographic material expiration.

6.7 Security

Before any key management functionality is allowed, a device must be authenticated. See "AAA Service Component" (Section 7.0) for discussion of device authentication. After authentication, devices will utilize KMIP for key management processes. KMIP communication is protected via TLS with mutual authentication. TLS provides both message integrity and confidentiality. Both of which are necessary since cryptographic material is being transmitted.

7.0 AAA Service Component

7.1 Architectural Design

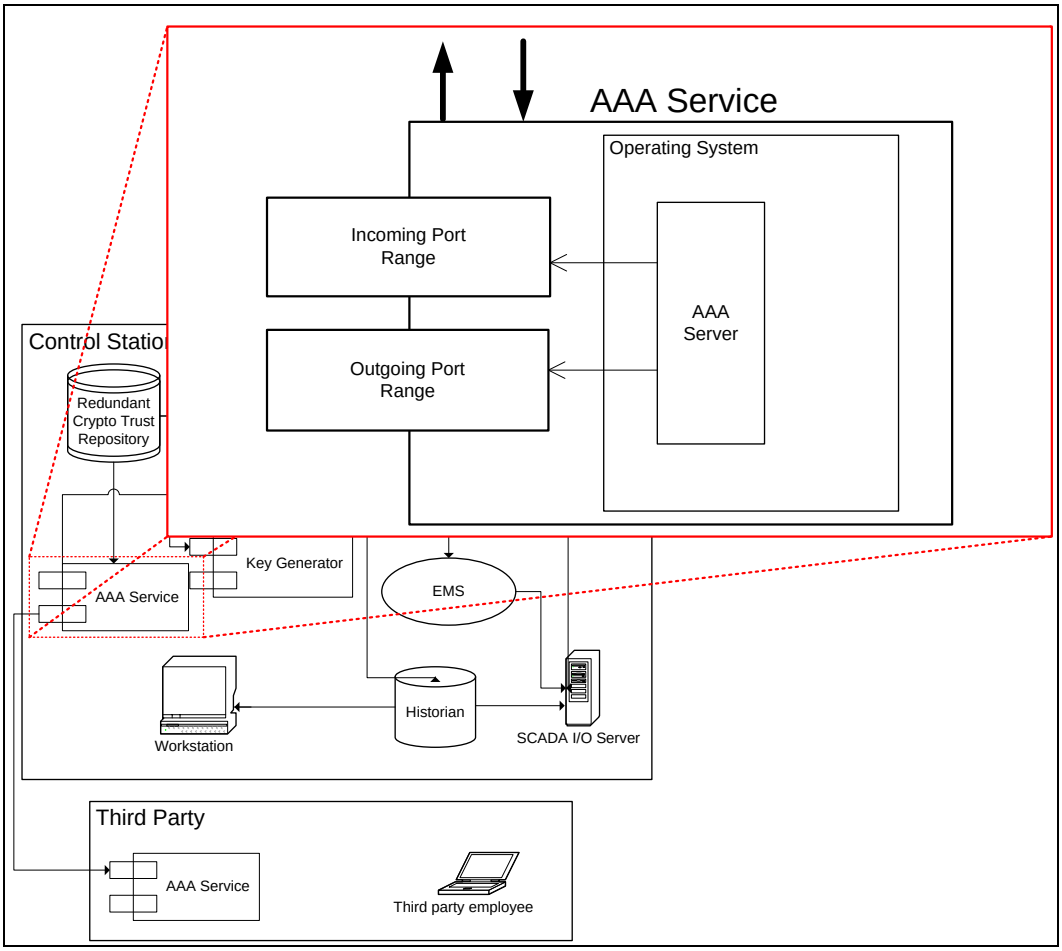

Figure 7.1. Architectural Design of the AAA Service Component

The function of the AAA Service component is to authenticate and authorize entities and devices and to provide accounting of these events. Before devices are allowed access to cryptographic management functions, and before entities are allowed access to devices and resources, they must first be authenticated by the AAA Service. Credentials for the devices and entities are transmitted to the AAA Service. The Cryptographic Trust Repository will maintain credentials and access rights or policies configured for all entities and devices. The AAA Service will query the Trust Repository to verify that the credentials received are for a valid user of the system and to determine what access privileges the user is allowed. The AAA Service will then transmit an authorization ticket back to the devices and entities authorizing them to perform the requested actions. For local authentication purposes, the Remote Trust Cache will also receive this authorization ticket.

The AAA Service also has the unique capability to negotiate trust between third parties whose employees need access to the devices and resources managed by the Cryptographic Trust Management System. It is often the case that vendors, integrators, and other third party organizations will require

access to equipment for maintenance or integration purposes. A design goal for the CTM System is to increase the security and assurance of third party connections. The best practice used today is to create, distribute, and manage credentials for third party employees locally. Handling authentication in this manner makes critical aspects of the authorization reliant on manual processes and contractual obligations. The trust negotiation capability by the AAA Service component adds real-time assurance of third party access for each connection.

In addition to authentication and authorization, the AAA Service functions by providing a complete audit trail of device access to be logged in the Cryptographic Trust Repository. The data provided by the AAA Service will be a major component of the incident response and forensics view in the User Interface component. For more discussion of this topic, see Subsection 9.3 within "User Interface Component."

Commodity hardware is the only hardware the AAA Service requires. The component consists of a common server with an operating system running networked services for performing the authentication, authorization, and third party trust functions. The services will run on top of commodity server applications waiting for and processing requests. The protocols needed (described in the next section) for the AAA Service component are all new protocols based upon well-known and tested protocols.

7.2 Authentication Process

Due to the unique requirements of the control system environment (as described in Delay Tolerant Centralized Security Management), a hybrid of two common protocols has been designed for the AAA Service component's authentication process. The environment is unique in that it has centralized management with decentralized fault tolerant outstations. Therefore, a hybrid of the EAP-IKEv2 and Kerberos protocols is used to accommodate this characteristic. The EAP provides dual authentication and the Kerberos provides a ticketing system for time-limited distributed authorization.

The AAA Service component is the centralized point for all authentications. The AAA Service must first authenticate the credentials of all entities requesting access or devices requiring key management services. It will check the roles and policies associated with the credentials in the Cryptographic Trust Repository and will authorize the request if the credentials are valid. The AAA Service component will then send a Kerberos style ticket to the Cryptographic Remote Trust Cache. The Remote Trust Cache acts as an agent sitting in all outstations providing the communication gateway to all Cryptographic Trust Management System services in the control station. The Remote Trust Cache will use the ticket to perform local authentications and authorizations for a policy-defined limited time frame. Only when that limited time frame expires does the Remote Trust Cache again forward credentials to the AAA Service component for authentication.

7.2.1 Hybrid Protocol

The hybrid authentication protocol uses EAP IKEv2 to bootstrap a Kerberos Authentication Exchange for devices installed in the field. This protocol allows devices to communicate with a centralized authentication server to verify and enforce appropriate security policies and practices. The following work flow of the protocol will highlight what actions are performed and their significance. The term "Client" in the process refers to either an entity or device. The process is interchangeable for both.

The first two phases are part of the EAP-IKEv2 protocol for authentication. IKEv2 provides dual authentication and is a common protocol used by other protocols such as IPSEC. Currently, it is recommended that public key cryptography be used for the authentication process due to its security strength and more established use. However, one of the requirements for the CTM System is to provide a mechanism to use symmetric keys for authentication to devices incapable of processing public key cryptography. The EAP-IKEv2 protocol supports the use of symmetric keys for authentication of a device. The process described in this subsection only describes how the recommended public key process operates. In future iterations of this document, the modifications for a symmetric key authentication process will be included.

Phase 1: Establish a Security Association

A security association is an agreement of security parameters and the creation of a secure communication channel to perform the subsequent phases.

1. Both parties perform the Diffie-Hellman (DH) key agreement protocol.
    a. Client sends a message to Remote Trust Cache containing DH key exchange, protocol agreement, and nonces.
    b. Remote Trust Cache sends a message to Client containing DH key exchange, protocol agreement, and nonces.
2. Both parties mutually authenticate.
    a. Client sends a message to Remote Trust Cache containing digital signature and public key, Remote Trust Cache key from first exchange, nonce authentication, and Client ID.
    b. Remote Trust Cache sends a message to AAA Service containing digital signature and public key, and nonce authentication.
    c. AAA Service sends a message to Remote Trust Cache containing verifying digital signature and public key and nonce authentication.
    d. Remote Trust Cache sends a message to Client containing digital signature and public key, Client key from first exchange, and nonce authentication.

Phase 2: Create new child Security Association

This optional phase allows the Client and Server to create multiple child security associations as needed.

*Note: If Perfect Forward Secrecy is required, a new Diffie-Hellman key exchange is required for each Phase 2.

1. Optional DH key exchange and security association setup, as in Phase 1.
    a. Client sends a message to Cryptographic Remote Trust Cache containing DH key exchange, protocol agreement, and nonces.

b. Remote Trust Cache server sends a message to Client containing DH key exchange, protocol agreement, and nonces.

After the initial EAP-IKEv2 authentication process, the authorization process of the Kerberos takes over. The rest of the phases in this protocol are influenced by the Kerberos protocol. Specifically, the ticketing system of Kerberos is used to enable the distributed authentication and authorization capability that is delay-tolerant to loss of communication to the control station.

Phase 3: Client Authentication

1. Remote Trust Cache sends a message to AAA Service containing Client ID and Client digital signature and public key.

2. AAA Service sends a message to Remote Trust Cache containing Client encrypt[session key] encrypted by the Client's public key and encrypt[Ticket Granting Ticket (TGT)] encrypted by the Remote Trust Cache's shared secret key.

3. Remote Trust Cache sends a message to device containing Client encrypt[session key] encrypted by the Client's public key.

After the completion of this phase, the AAA Service authentication and authorization are complete. The Ticket Granting Ticket and the session key are the enabling components for the Cryptographic Remote Trust Cache to perform outstation local authentication and authorization.

It is also at this stage that the third party trust negotiation process described in the Inter-Organization RBAC subsection below will be initiated if the client's credentials belong to a third party. This process must be successful before the AAA Service will send back the TGT and session key. Otherwise a failed authentication message will be sent back.

Phase 4: Client Service Authorization

1. Client decrypts encrypt[session key] with its own key.

2. Client sends a Service Server Request message to Remote Trust Cache.

3. Client sends a message to Remote Trust Cache containing encrypt[Authentication] encrypted with the shared secret session key.

4. Remote Trust Cache decrypts encrypt[Ticket Granting Ticket (TGT)] with its own shared secret key.

5. Remote Trust Cache uses the TGT to decrypt the encrypt[Authentication] message from the Client.

6. Remote Trust Cache sends a message to Client containing encrypt[Client-to-Server Ticket] encrypted by the Service Server's shared secret key.

7. Remote Trust Cache sends a message to Client containing encrypt[Device/ Server] encrypted with the shared secret session key.

Phase 5: Client Service Request

At this phase, the client is requesting access to a service. For a device, the service is with the Key Manager. For an entity, the request will be for access to a device.

1. Client decrypts encrypt[Device/ Server] with the shared secret session key.
2. Client sends a message to Service Server containing encrypt[Client-to-Server Ticket].
3. Client sends a message to Service Server containing new encrypt[Authentication] encrypted with the shared secret Client/ Server key.
4. Service Server decrypts encrypt[Client-to-Server Ticket] using its own shared secret key.
5. Service Server decrypts new encrypt[Authentication] using the decrypted shared secret Client/ Server key from previous decryption.
6. Service Server sends a message to Client containing encrypt[Timestamp] encrypted with the Client/ Server shared secret key.
7. The Client decrypts encrypt[Timestamp] with the shared secret Client/Server Key and compares timestamps.
8. Client can now use Service Server. Service Server now provides access to Device.

Phase 6: Terminate Associations

For key security, all participants should eliminate established security associations after specified information or a determined period of time has passed. Each participant will cease to use keys in expired security associations, if perfect forward secrecy is requires new nonces and/or new Diffie-Hellman Key exchanges are required for each Security Association.

7.2.2 Device Authentication

A device goes through the authentication and authorization process for the sole purpose of gaining access to the Key Manager component to request cryptographic material. The Cryptographic Remote Trust Cache will cache both authentication information and cryptographic material. Therefore, if the Remote Trust Cache has a local copy of the cryptographic material requested, the authentication process stops at phase 5. Instead of exchanging keys for the device to communicate with the Key Manager, the Remote Trust Cache will simply send the local copy to the device.

7.2.3 Entity Authentication

An entity will request access to devices for interaction such as maintenance or engineering access. In such a case, the authentication process will more closely follow the Kerberos style of service

authorization. If the entity is from a third party then a third party trust negotiation must occur to authenticate and authorize the access.

7.3 Inter-Organization Role-Based Access Control

The AAA Service component will provide the capability of assuring the validity of every third party connection. Third party device connectivity will follow device authentication but this section defines how third party employee connections are managed. In order for the Cryptographic Trust Management System to verify third party connections, there must be prior definitions and policies in place outlining communication with a third party AAA Service, what is allowed with the third party connection, and industry accepted pre-defined roles to negotiate access privileges. Using this information, the AAA Service component will check a third party's credential to determine which organization it is representing and if there is a configured trust connection with that organization. If there is a configured trust connection, the next step is to check if the third party has a role associated with the device in the access request. If there is an associated role, the AAA Service component will create a connection with the third party's AAA Service and will send the requesting employee's credentials. The third party will return a signed verification that the credential is valid and what roles the employee should fill. The local AAA Service will then verify that the roles given for the employee correspond to the role configured for this third party. Additional requirements, called trust evidence which is further defined in 7.3.3, can be configured for third party trust connections. In addition to the credentials and roles, the trust evidence must also be met. For example, the trust evidence could include a restriction for the connection to a range of IP addresses. If the connecting entity does not come from that range, the connection will be refused. If all of these requirements are met, the AAA Service will consider the third party employee authenticated and authorized and will send the Ticket Granting Ticket and session key to the Cryptographic Remote Trust Cache. If any one of these steps fails, the AAA Service will send a failure message.

The Cryptographic Trust Management System's AAA Service third party authorization process is designed to work without prior knowledge, i.e. pre-shared or pre-configured cryptographic material, of users from other companies or managing network accounts for those users. This is accomplished through the use of attribute credentials, a shared ontology of attribute names and values, trust evidence, trust negotiation, and a shared trust negotiation language. This section will explain these terms and how they are used by the Cryptographic Trust Management System to provide meaningful authorization with minimal overhead.

7.3.1 Trust Negotiation

When a third-party's employee attempts to gain access to the local company's remote station, the employee's credential is passed to the Cryptographic Remote Trust Cache, which caches Kerberos-style authorization tickets issued during a previous session. Tickets in the cache have expirations dates. If a cached, unexpired ticket exists from a previous session, the user's current session is authorized. If there is no cached, unexpired ticket, then the employee's credential is passed to the local company's AAA Service which will examine the credential to determine which company issued it. With this information and pre-configured third-party connection information, e.g. the third party's IP address, the AAA Service connects to the remote company's AAA Service and begins a process called trust negotiation. During trust negotiation, each party iteratively requests credentials from the other until trust is established between them. See Figure 7.2.

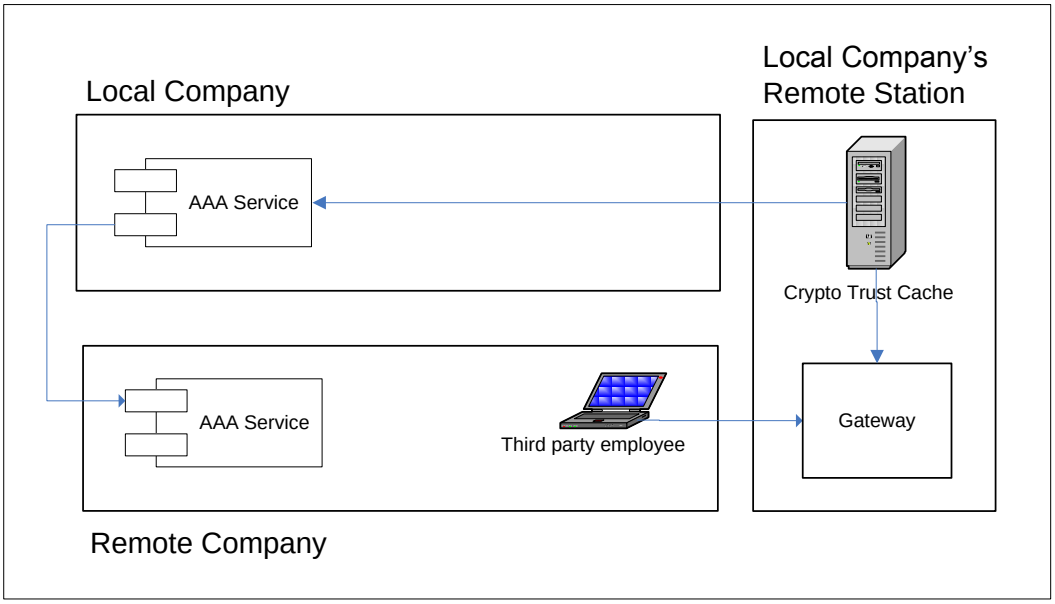

Figure 7.2. Local and Remote AAA Services Automate the Trust Negotiation Process

The credentials that are requested and the order in which they are requested are determined by policy. Cooperating companies must use a common trust negotiation language in order to carry on this dialog with each other. An example trust negotiation is as follows:

Local company	Sends user X's identity credential to remote company with request for X's role credentials.
Remote company	Requests signed AAA Service credential from local company.
Local company	Sends signed AAA Service credential to remote company.
Remote company	Compares credential to the one it has pre-installed for the local company and they match.
Remote company	Sends user X's role credentials to local company.

This trust negotiation sequence (which can include additional steps as per policy) allows the local company to ask the third party to endorse its employee and to bear the responsibility for appropriately assigning attributes (roles, etc.) for its employees and keeping their credential database up-to-date.

7.3.2 Trust Policy Language

PKI certificates bind a key to a person's credential. More recently, digitally-signed attribute credentials have been developed as a means to authorize and to delegate authorization. For example, an employee may possess a credential signed and dated by his or her company stating that the person has their company's approval to maintain customer equipment. Trust is placed in the holder of the credential so long as the company that signed the credential is trusted. Because multiple organizations will be

exchanging the attribute credentials, they must all agree on the format of the credential to be used. An example is the Simple Public Key Infrastructure/Simple Distributed Security Infrastructure (SPKI/SDSI) credential format.

To be meaningful, cooperating organizations must also agree on a list of defined, unambiguous attributes and the list of valid values for each of those attributes that may be used in the credentials. The most likely attribute is "role," referring to the role the person has been given by their own organization to fulfill in relation to external organizations. For example, the role in the credential might be Field Unit Maintenance. Agreement on this ontology of attributes and values should be negotiated by an appropriate industry group for each type of process control industry (e.g., electric, water, gas, etc.). These attributes can then be used in access control lists and other such mechanisms to provide access to networks, devices, or processes.

7.3.3 Trust Evidence

The items passed back and forth during trust negotiation or otherwise gathered and considered during the establishment of trust are termed trust evidence. Trust evidence may include items such as:

- Role credential (e.g., Field Unit Maintenance, ISO, etc.) dated and signed with the third party organization's private key.
- Current employee credential dated and signed with the third party organization's private key, testifying that the holder is a current employee in good standing.
- IP address within an expected range for that organization.
- The local organization's work order number for which the external entity is seeking access (for roles such as Field Unit Maintenance).
- The reputation of the company providing information about their employees. This can involve the local company tracking their own experiences with the company, and may also involve periodically querying other companies to determine what their experience with the remote company has been. This reputation evaluation provides a powerful incentive for the company to respond as a good citizen of the cooperative if they wish to continue as a member of the cooperative.

7.4 Cache Ticketing

Upon successful authentication and authorization, the AAA Service will send a Kerberos-like ticket to the Cryptographic Remote Trust Cache component. This ticket not only includes the session key necessary to provide local authentication and authorization, but also policy information or access privileges depending on whether it was a device or entity that was authenticated.

The first important value in the ticket is a validity period. The validity period tells the Remote Trust Cache how long it should perform local authentication and authorization services using the session key.

Once the period expires, the session key is deleted by the Remote Trust Cache and the next authentication request will start the authentication process from the beginning.

The next value in the ticket is the access rights of an entity. The access rights allow the Remote Trust Cache to provide local authorization services to allow an entity to connect to a device. These access privileges are only for local devices and are only valid as long as the ticket validity period. For more discussion of the ticketing process, see Subsection 8.2 within "Cryptographic Remote Trust Cache Component."

7.5 Security

The AAA Service component uses a hybrid of the EAP-IKEv2 and Kerberos protocols to provide centralized management of decentralized delay tolerant authentication and authorization services. EAP-IKEv2 provides mutual authentication and all connections are secured during the authentication process. Kerberos provides a ticketing system for symmetric key authorization services. Kerberos is a well understood protocol that is leveraged by most enterprise environments for authentication to a domain (for example, Microsoft's Active Directory service is a Kerberos-based authentication system).

The third party trust negotiation process provides real-time assurance of the validity of third party employee access to equipment. This capability of the AAA Service component prevents former third party employees (who perhaps are disgruntled after a termination) from accessing the system.

8.0 Cryptographic Remote Trust Cache Component

8.1 Architectural Design

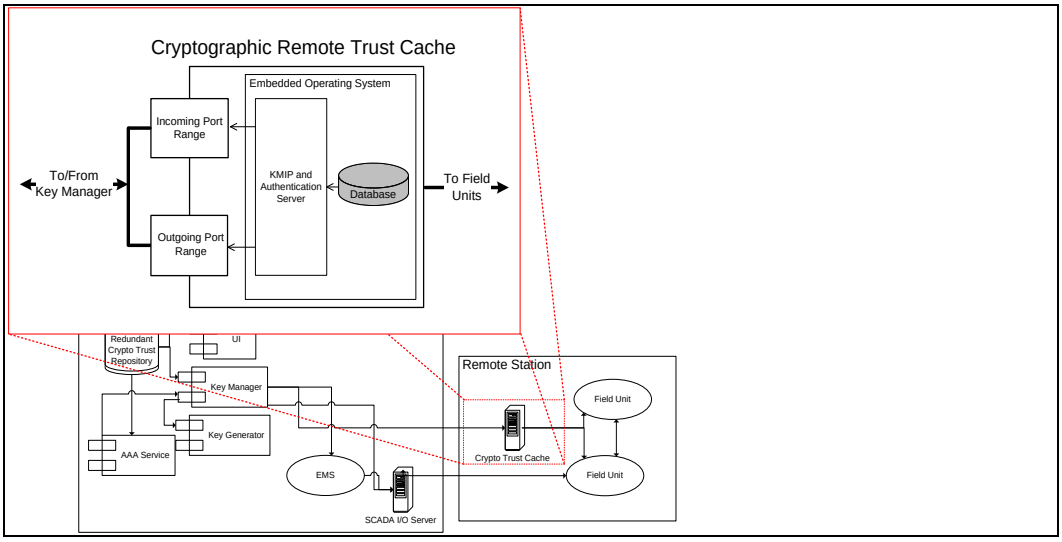

Figure 8.1. Architectural Design of the Cryptographic Remote Trust Cache Component

The Cryptographic Remote Trust Cache component is a small cryptographic trust management system contained in an outstation. The purpose of the Remote Trust Cache is to provide functionality to keep outstation equipment working in the event of loss of communication back to the control station. The Remote Trust Cache is the security communication gateway to the rest of the Cryptographic Trust Management System components for all devices and entities in an outstation and it takes in all authentication and cryptographic material management requests. It will either process these requests locally, if the necessary data is available, or will forward on the requests to the responsible CTM component in the control station.

The Remote Trust Cache component is built with similar components as the rest of the CTM System since it mimics their capabilities. The Remote Trust Cache securely stores authentication tickets so it requires an embedded database application for ease of retrieving information pertaining to a requesting device. The database is implemented such that all data at rest is secured with best practice cryptography. In front of the database are two network services for processing requests. The first is an authentication service which will take authentication requests from devices and entities and will either authenticate locally or send the request to the AAA Service in the control station. The other service is a KMIP service for receiving cryptographic material requests and sending expiration notifications and cryptographic material synchronization commands.

The Remote Trust Cache should be an environment hardened FIPS 140-2 Level 3 certified embedded device. It should not contain non-volatile storage capacity. Since the Remote Trust Cache will be storing cryptographic material for all of the devices in the substation, it will be a higher risk target. Therefore, the system should only store authentication and cryptographic material in volatile memory and requires FIPS

140-2 tamper resistant memory clearing capabilities. An embedded operating system and components should be utilized to create the Remote Trust Cache component.

8.2 Authentication

The Cryptographic Remote Trust Cache component will provide the EAP authentication gateway for devices and entities requiring access in outstations. When an entity requires access or a device requires key management services, it must first be authenticated. The Remote Trust Cache component utilizes a hybrid EAP-IKEv2 and Kerberos authentication process. A first authentication will perform an EAP authentication back to the AAA Service component. A result of the EAP authentication is a Kerberos style ticket and session key from the AAA Service. For the duration of the ticket, successive authentications will be performed locally to the outstation by the Remote Trust Cache component.

The Kerberos style local authentication functionality is provided to meet a critical requirement. That is, security applications integrated with the Cryptographic Trust Management System must continue to operate for a period of time even in the event that communication with the CTM System is lost. For more discussion on the authentication process, see subsection 7.2 of the AAA Service Component section.

8.2.1 Ticket

The ticket sent from the AAA Service is encrypted with the Remote Trust Cache's public key and includes the session key for the authenticated device, the time validity, and the access privileges of the entity or device. The Remote Trust Cache component will store the ticket and forward the session key on to the device or entity. When the entity or device needs to authenticate in the future, it will use the session key to encrypt a challenge nonce. The Remote Trust Cache will use the session key from the ticket to check the validity of the message from the authenticating device and determine if the session key is still valid according to the time period in the ticket. If both are valid, the Remote Trust Cache will authorize the device. If the time period is not valid, the device must instead authenticate at the AAA Service component.

A policy is attached during registration of cryptographic material and the creation of roles. Part of this policy is the validity period of the ticketing process that allows a device or entity to authenticate locally to the outstation and not require communication to the main CTM System. This time period should be configured for how long a device or entity should continue to have access in the event of a communication loss. It is recommended that the time period be greatest for devices, less for internal roles, and the least for external roles. This configuration ensures that devices can keep operating, but would not allow third parties to continue access for long periods without going through another full authentication and authorization process.

8.3 Key Management

After authenticating a device, the device is authorized to perform cryptographic material management requests. An initial cryptographic material request, which includes a unique identifier installed in the device during provisioning, will flow through the Cryptographic Remote Trust Cache back to the Key Manager which will retrieve information from the Cryptographic Trust Repository component. The Key Manager will send the requested cryptographic material back to the device through the Remote Trust

Cache component. The Remote Trust Cache will store the cryptographic material to fulfill further cryptographic material requests while the authentication ticket is still valid.

8.4 Authorization

When an entity requests access to a device, the AAA Service will send the access rights as part of the ticket. Using these access rights, the Cryptographic Remote Trust Cache will authorize the entity to access resources. An entity can have different levels of access determined by the role associated with their credentials in the Cryptographic Trust Repository. For more discussion on the creation of roles, see Subsection 4.4.3 within "Cryptographic Trust Repository Component."

8.5 Cache

The Cryptographic Remote Trust Cache component will cache cryptographic material as it is passed from the Key Manager component to devices. It will cache the cryptographic material to ensure that devices can access the cryptographic material for a period without requiring communication back to the control station. This speeds up delivery and ensures continued access in the case of a communication loss. The Remote Trust Cache is also responsible for synchronizing cryptographic material between devices. If one device updates its cryptographic material, all other devices utilizing the same shared cryptographic material will be out of synch until they too update. Therefore, the Remote Trust Cache will send the new cryptographic material to all devices sharing the cryptographic material to avoid synchronization problems.

Cached cryptographic material makes the Remote Trust Cache component a higher risk target. To provide mitigation of this risk, the Remote Trust Cache does not write cryptographic material to non-volatile memory. This provides the protection that if the component is stolen a large subset of cryptographic material will not be lost. A second layer of protection is FIPS 140-2 tamper resistant hardware. If the Remote Trust Cache is tampered with it will clear its memory making it even harder for the stored cryptographic material to be compromised.

8.6 Security

The Cryptographic Remote Trust Cache component is designed to be a hardened appliance that can manage cryptographic material in isolated locations with potentially weak physical security controls. The component only stores cryptographic material in volatile memory and that memory is protected by Level 3 FIPS 140-2 tamper resistant hardware. If the device senses tampering, it will wipe its memory to protect against malicious entities compromising cryptographic material. FIPS140-2 Level 3 hardware is typically used to protect mission critical federal equipment which makes it appropriately secure for the mission critical security applications in control system environments.

The Remote Trust Cache also never stores long-term authentication credentials. The Cryptographic Trust Management System uses a hybrid EAP-IKEv2 and Kerberos authentication and authorization process. Therefore, only session authentication credentials are stored locally. This provides the added benefit of allowing the continued operation of the environment for a policy-defined period in the event of

a loss of communication with the control station. The policy-based time period allows the flexibility of restricting third party connections more so than internal connections.

9.0 User Interface Component

9.1 Architectural Design

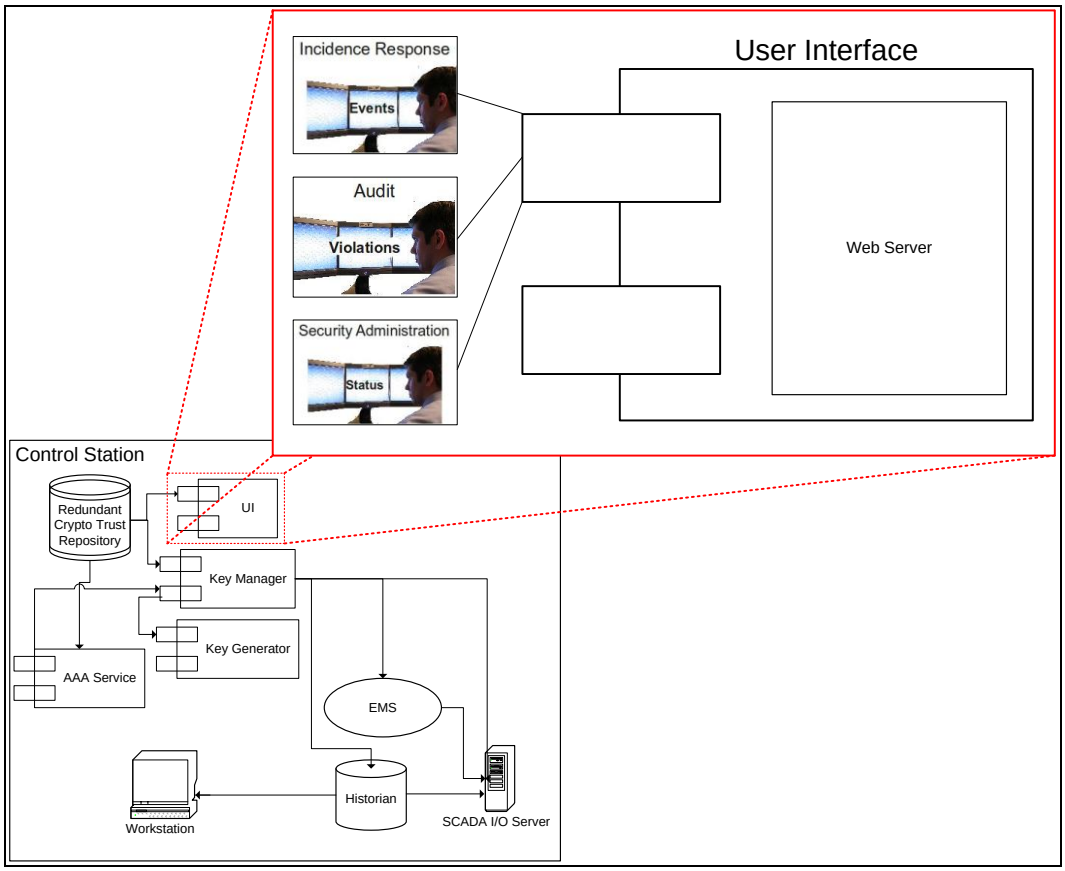

Figure 9.1. Architectural Design of the User Interface Component

The Cryptographic Trust Management System User Interface (UI) component is designed as an integrated system that provides an interface to all functions of the CTM System. These functions include system status information, reporting, system administration, and key management. The UI component is a role-based system, and as such will tailor itself to an authenticated user and their particular access level and responsibilities.

The front end of the UI will be a web-based portal that will employ standard web applications and protocols, while the back end will interface exclusively to the Cryptographic Trust Repository which will operate to configure policies, register devices, perform manual key management functionality, and retrieve reports, in addition to other capabilities. The web server and programming language selected to implement the UI component are irrelevant. All major web server applications, such as Apache Web Server or Microsoft IIS, are capable of serving the UI component. Also, the selection of a web programming language is more of a preference than a requirement. The UI component will not be handling an extreme load of connections so the common selection criteria for web-based languages do not apply.

A common set of roles will be created by default for the UI component for which users can be assigned. These roles will provide a subset of functionality and data access as is defined in the rest of this section. Each user has their role assigned by the system administrator role. The system administrator role has the ability to configure anything in the system but does not have access to cryptographic material, which is reserved for individually assigned roles. The role-based design of the system follows best practices for separation of duties.

Users will access the system by opening a web page in their browser and will be prompted for credentials which, when successfully authenticated and authorized, will bring up the main User Interface. As the UI is role-based, this initial screen will vary from role to role. All interaction with the UI component will be performed via a web browser.

9.2 Real-Time System Status

The main user role of the User Interface component is the security administrator. The security administrator is responsible for maintaining the security of all cryptographic mechanisms. The security administrator monitors the health of security systems, handles security events when they occur, and executes key management functions. Upon authentication to the UI component, the security administrator will be presented with the real-time system status view.

The real-time system status view will display all the information relevant to the cryptographic material managed by the Cryptographic Trust Management System. Central to the display are the cryptographic material identifiers. All of the metadata associated with a cryptographic material identifier can be viewed. Some of the more often used data, such as the cryptographic material and lifetime percentage, will be presented in a convenient format on the starter screens along with the identifier.

From the real-time status view, the security administrator will be notified of security events. If there are any events associated with cryptographic material, such as a policy violation or other event notification, the user will be presented with an indicative visual cue and the option of drilling down to a summary of the event. The summary will include information such as a text description of the event, the time it occurred, and the associated devices or entities. In addition, there will be an option to switch contexts to an event notifications screen where a clearing house list of events will reside.

9.2.1 Cryptographic Material Lifetime

Once cryptographic material is created, it has a finite amount of time in which it is valid. The CTM System stores policy information associated with the cryptographic material at the time cryptographic material is created. This information includes the percent of lifetime the material has left, the date and time of expiration, and a soft expiration date. The soft expiration date is a threshold, based on lifetime percentage, which the system uses to notify applications they should start trying to update keys during a down time. This threshold is used to minimize the impact of key updates on regular operations. This material can be viewed by the security administrator from the real-time system view. Also available to the security administrator from this view is information to see when keys have expired and when notifications have been sent due to soft expirations.

9.2.2 Policy Violations and Event Notifications

The policy and events screen displays a table view to the user of the occurrences of events and policy violations based on key identifiers and device associations. A policy violation can occur when a device fails to comply with the policy that is associated with it (e.g., failure to perform a key update after a key expires). If a device fails to comply with its own policy, the system will store this violation and make it available as a visual cue on the status screen and as an entry in the policy violations/notifications table view. The visual cue will be distinguished only after the security administrator proceeds to view a summary on each new violation or event. As the CTM system is not responsible for enforcement of policy, only notification of it, this type of facility is critical to the user. Event notifications can occur for any number of reasons, including authentication failures or improper access requests. The UI component will again display these events relative to key identifiers and device associations. Notification is not limited exclusively to the UI. Notification can also be done via text message or email, and can be user-based or role-based as well. Detailed summary information about the event or violation and any metadata are displayed in a separate view.

9.2.3 Device and Entity Associations

All cryptographic material must have one or more devices or entities associated with it. An association to cryptographic material denotes use of the cryptographic material for a common application. The real-time system status view will display the cryptographic associations to assist the security administrator in quickly understanding which devices need attention in the event of an emergency.

The associations screen displays the details of the devices or entities, their key identifiers, and their policies. From this view, system administrators will be able to switch contexts to edit the cryptographic material, the policies governing them, and their associations. Standard users will only be able to review this information.

9.3 Report Views

The Cryptographic Trust Management System supports a role-based reporting system. This system provides reporting capabilities for both the auditor and the forensic analyst. Data availability is based on the user's role; however, reporting is customizable to support each role's reporting requirements.

9.3.1 Audit and Compliance

The report views for auditing and compliance allows the user to see policies, produce reports on policy violations, and produce reports based on regulatory compliance. The reports on policies show a summarization of the policies configured within the CTM System. This will enable a quick audit of the implemented policies against the organizational polices. Also, the reporting system will enable the generation of a report within a specified time frame (e.g., a month or year) that shows a summary of all policy violations, such as failure to update cryptographic material according to policy, and the amount of time it took to remedy the violations. Lastly, the reporting system for audit and compliance will allow generation of reports that audit compliance against industry regulatory standards (e.g., NERC CIP series).

9.3.2 Incident Response and Forensics

The User Interface component provides a view for incident response and forensic efforts. The view for incident response and forensics is in essence a reporting system that will compile summaries of information to assist in determining sequences of events. The reports are compiled around a specified time range.

The reporting system for incident response and forensics includes reporting for policy violations, events (e.g., authentication failures), and audit logs. Audit logs include key management events such as key requests and notifications, as well as AAA Service logs which include internal and external authentications.

9.4 System Configuration

The User Interface component is the sole interface for configuring the Cryptographic Trust Management System. Through the UI, the security administrator role can register devices and establish the amount of cryptographic material that will be needed, define cryptographic policies, create roles, and establish third party trust negotiation connections.

9.4.1 Device Association

Before a device can be managed by the CTM System it must be provisioned and registered. The two processes will establish device credentials for authentication and will establish cryptographic material that the device will need to function. Once the cryptographic material is defined, the devices that will be using it are associated or attached to it. This association is how the CTM System knows which devices are using cryptographic material.

During operation, security administrators will have the ability to create new or modify current cryptographic associations. The cryptographic associations screen will present the user with a list of cryptographic material and the devices to which it is associated. Devices can be added or removed from the association list. Changes can be made to existing cryptographic material by simply clicking on a device and performing the desired action.

9.4.2 Policy Configuration

All cryptographic material must have an associated cryptographic policy. The policy configuration screen gives the system administrator the ability to create or modify these policies. A policy is a set of rules and guidelines to which cryptographic material must conform. For example, Secure DNP3 requires a shared key to operate. A policy for Secure DNP3 could be that the key must be updated every six months. The CTM System does not enforce these policies but will only alert on violations. Instead, the policies are enforced by the security administrator responsible for the devices or applications. The configuration screen contains a table view of all the current policies on the system with a detailed view below the table view.

9.4.3 User Administration and Role Definitions

The user administration screen gives the administrator the ability to add, modify, or delete users. A user has a set of credentials that uniquely identifies them. Users have metadata associated with them much like devices do. All users must have a role or roles associated with them to be provided with access to devices.

The UI's user administration screen also gives the administrator the ability to define and assign roles for each user. Roles are a set of access rights for a certain job description, such as an outstation maintenance role could be given access rights to all outstation devices but no access to devices in the control station. The primary view is a table that lists users and their respective role. From here the administrator can click on a user's name to perform desired actions. This screen also allows the administrator to define event notification rules (such as text messaging and emailing), which can be user-based, role-based, or both.

9.4.4 Third Party Trust Configurations

The CTM System will rely on third party AAA Services to authenticate and verify roles of third party employees. To enable this functionality, the connections to the third party AAA Service's server need to be configured. The UI component provides the capability for the system administrator to configure third party trust configurations.

A third party trust configuration consists of the information necessary to securely connect to the third party server. The network configuration details, such as IP address and VPN tunnel details and certificate, are necessary for the CTM AAA Service to securely communicate with the third party's AAA Service. The AAA Service provides an authorization framework allowing differing amounts of trust evidence per third party. An authorization policy must be defined and associated with every third party trust configuration. Also, each industry will need to predefine industry roles as part of the trust policy language used in the communication between third party AAA Services. A set of roles need to be defined for each third party representing the job responsibilities and access rights for the third party employees. These roles are associated with the third party trust configuration.

Configuring third party trust is the primary responsibility of the security administrator.

9.5 Manual Cryptographic Material Management

The User Interface component allows manual cryptographic material management functions for security administrators. This functionality enables forced cryptographic material management functions within the automated cryptographic material management system and to allow the Cryptographic Trust Management System to manage cryptographic material for legacy systems that cannot integrate with the automated cryptographic material processes.

9.5.1 Cryptographic Material Generation and Device Association

The security administrator is also responsible for the manual generation of keys and device associations in instances where a device is incapable of communicating directly with the CTM System.

This screen will contain a simple interface for key generation and have a visual hook into the device association interface described in “Device and Entity Associations” (9.2.3). This screen will also have options for updating manually generated keys and associations. It will also show notifications of expiration thresholds reached and will have visual hooks to the event listing screen described in “Policy Violations and Event Notifications” (9.2.2), and to a screen for listing all the manual devices needing cryptographic maintenance.

9.5.2 Key Update

The security administrator can force the expiration of cryptographic material which forces devices and entities to update their keys. When a manual cryptographic update is performed for a device, it is the same as a hard key expiration. See “Key Manager Component” (6.0). When an update is forced for a device’s or entity’s credentials, it will force an update of new credentials the next time the entity or device authenticates to the system. See “AAA Service Component” (7.0) for further discussion.

9.5.3 Key Revocation

The security administrator can revoke cryptographic material. There is not a revocation concept for device cryptographic material. In this context, a cryptographic material revocation is the same as an update which forces new cryptographic material to be exchanged. However, device and entity credentials can be revoked. A revocation of credentials is essentially removing access privileges from the device or entity. Without credentials in the CTM System, devices and entities cannot authenticate and are not able to gain access authorization.

The UI component provides the ability to delete credentials. On the device and entity associations screen described in Subsection 9.2.3, the security administrator will be given the option of reviewing and deleting devices and entities in the system.

9.6 Security

The User Interface component provides different levels of security for the Cryptographic Trust Management System. First, the web portal provides role-based access control to provide separation of duties and least privilege capabilities. The UI component only allows security administrators access to manually manage cryptographic material. The real-time status view provides a quick view of the health of the CTM System. The incident response and forensic view assists investigators when a security event occurs.

9.6.1 Role-Based Access Control

When a user logs onto the system, their role dictates what can be seen and what can be done. For instance, a security administrator will be able to view and create policies, where an auditor will only be able to view those policies. By default there are four roles for users of the User Interface; system administrator, security administrator, auditor, and forensic analyst. The system administrator has read and write access to the entire system. The security administrator has access to the real-time status view, device configuration and association, cryptographic material generation, trust negotiation between third parties, and manual cryptographic material management. An auditor can view policy violations and verify

whether current policies are compliant with organizational and regulatory policies. The forensic analyst can view audit trails and events to evaluate potential root causes of incidents.

9.6.2 Cryptographic Material Access

The UI component does not allow access to current cryptographic material unless it is associated with a manually managed device. The CTM System is designed to operate with as little human interaction as possible and therefore, should not allow users access to cryptographic material. This design decision is to provide barriers to the insider threat, preventing the misuse of cryptographic material. However, for devices that are not able to integrate with the automated cryptographic material management process, it is necessary for security administrators to have access to the cryptographic material so that they can manually install it in the devices.

The CTM System will archive cryptographic material for data retention requirements. Archived cryptographic material can be retrieved by users when needed.

9.6.3 Real-Time Status

The real-time status view provides a quick health check of the cryptographic security of the control system network. It provides notifications of critical events and notifications of which devices require attention. With the real-time status view, a security administrator can quickly ascertain what, if any, actions need to be taken to maintain a secure control system network.

9.7 Incident Response and Forensics

The incident response and forensic view is provided to accelerate the investigation of a security event in the control system network. It is a tool to be utilized to gain insight on what security events occurred during the time frame of the event in order to assist in the discovery of oddities and discrepancies. With the incident response and forensic view, investigation of the root cause of incidents will be greatly enhanced.

10.0 Device Trust Lifecycle

This section defines the trust lifecycle for devices, including all of the steps to register and authenticate devices and perform key management functions.

10.1 Device Registration

1. The user creates credentials for the device being registered.

2. The user configures the cryptographic material used by the device using the UI.

    a. A unique identifier for the cryptographic material is created.

    b. Cryptographic key type and application type. (These will be tied to a policy that states cryptographic level (e.g., AES 256), key lifetime, soft expiration percent, etc.)

    c. The device's credentials are associated with cryptographic material.

3. The user configures the communication parameters necessary for the CTM System to communicate with the device.

    a. The network interface used.

    b. Other necessary network settings (e.g., Ethernet port eth0 and IP v4 address 192.168.1.5).

10.2 Device Provisioning

1. The user installs the unique identifiers of cryptographic material into device.

2. The user installs credentials of Cryptographic Remote Trust Cache into the device(for dual authentication).

3. The user installs the device.

10.3 Authentication

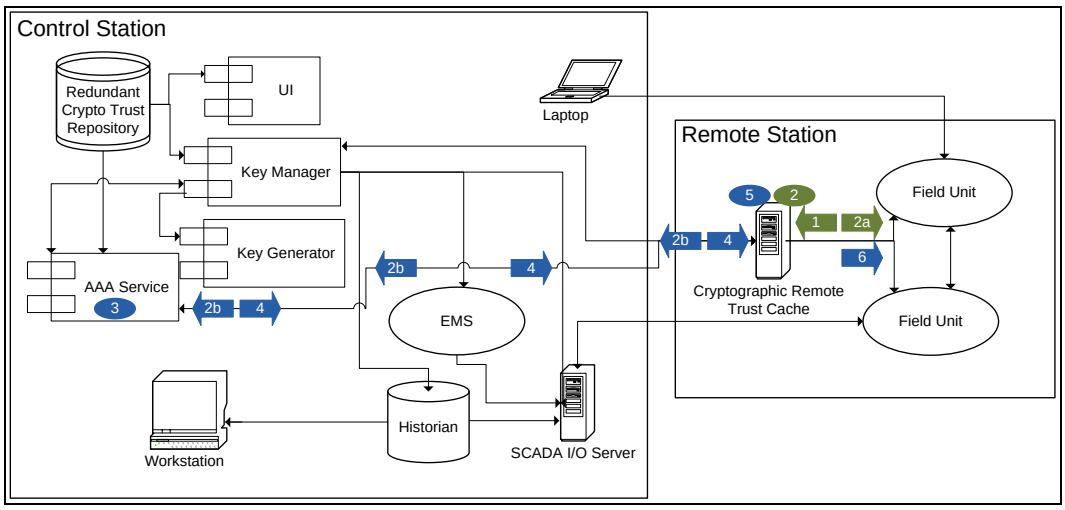

Figure 10.1. Device Authentication Process

1. The Device creates a secure connection with the Cryptographic Remote Trust Cache and sends authentication credentials.

2. The Cryptographic Remote Trust Cache checks for locally stored authentication ticket for device.

    a. If a ticket exists and hasn't expired, the Remote Trust Cache sends an authentication success message to device.

    b. If an unexpired ticket does not exist then the Cryptographic Remote Trust Cache forwards the device credentials to the AAA Service.

3. The AAA Service authenticates the device's credentials.

4. The AAA Service creates a Kerberos-like ticket and sends it to Cryptographic Remote Trust Cache.

5. The Cryptographic Remote Trust Cache stores the ticket.

6. The Cryptographic Remote Trust Cache sends an authentication success message to device.

10.4 Key Request

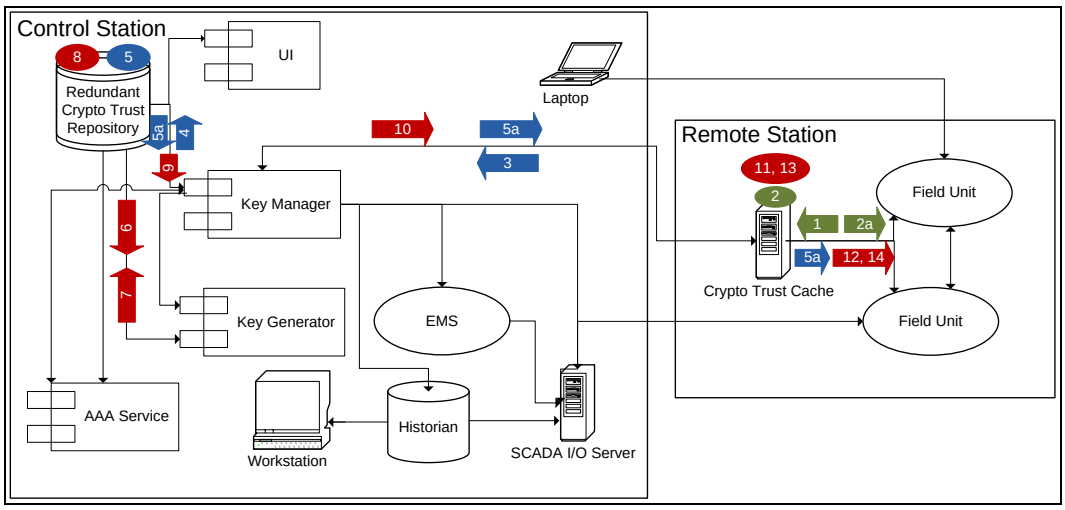

Figure 10.2. Key Request Process

1. The device sends request for key to Cryptographic Remote Trust Cache. The request includes the communication connection (i.e., the device communication partner) or the application so the system knows what key policy to use when creating the key.

2. The Cryptographic Remote Trust Cache checks for locally stored key to fulfill the device's request.

    a. If key is stored locally, Cryptographic Remote Trust Cache sends the key. (The Cryptographic Remote Trust Cache clears out keys when they are no longer valid, so if a key is found, it is still valid.)

    b. If no key exists, process continues to step 3.

3. The Cryptographic Remote Trust Cache forwards the request to the Key Manager.

4. The Key Manager requests the key from the Cryptographic Trust Repository.

5. The Cryptographic Trust Repository checks for a currently stored key that has not yet reached a soft expiration.

    a. If a current key exists, the Cryptographic Trust Repository sends the key to the Key Manager.

    b. If no current key exists, process continues to step 6.

6. The Cryptographic Trust Repository sends a request with cryptographic material specifications to the Key Generator for creation.

7. The Key Generator creates the cryptographic material and sends it to the Cryptographic Trust Repository.

8. The Cryptographic Trust Repository stores the new cryptographic material for the connection and updates its associated metadata (e.g., key lifetime, etc.).

9. The Cryptographic Trust Repository sends key material to The Key Manager.

10. The Key Manager sends the key material to the Cryptographic Remote Trust Cache.

11. The Cryptographic Remote Trust Cache stores the key associated with the connection or application.

12. The Cryptographic Remote Trust Cache sends the key to requesting device.

13. The Cryptographic Remote Trust Cache creates a secure session with the other devices associated with the connection or application.

14. The Cryptographic Remote Trust Cache sends the key to all of the associated devices so they will have the new key.

10.5 Key Expiration

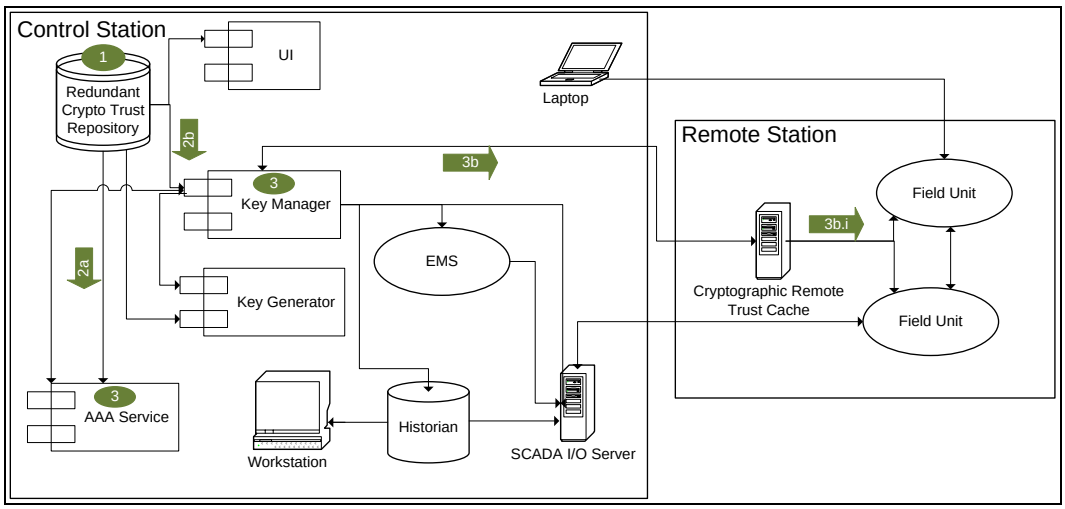

Figure 10.3. Key Expiration Process

1. A key's time threshold expires in the Cryptographic Trust Repository. This could be a soft or hard key expiration.

2. The Trust Repository notifies the appropriate service:

    a. The AAA Service is notified for entity and device credentials.

b. The Key Manager is notified for session keys.

3. The service performs the expiration process.

    a. The AAA Service will create new credentials and will distribute them on the next authentication for that entity or device.

    b. The Key Manager sends expiration notification. (The Cryptographic Trust Management System is pull-oriented, so the Key Manager simply notifies devices of key expiration but cannot enforce the key update.)

        i. The Key Manager sends expiration notice to all devices via their associated Cryptographic Remote Trust Caches. The notice includes the percent of lifetime remaining for key. This percentage is to notify the device of how soon a key update needs to occur. A soft key expiration threshold is provided so notice is given to a device with enough lead time to schedule a key update during down communication periods.

10.6 Key Revocation

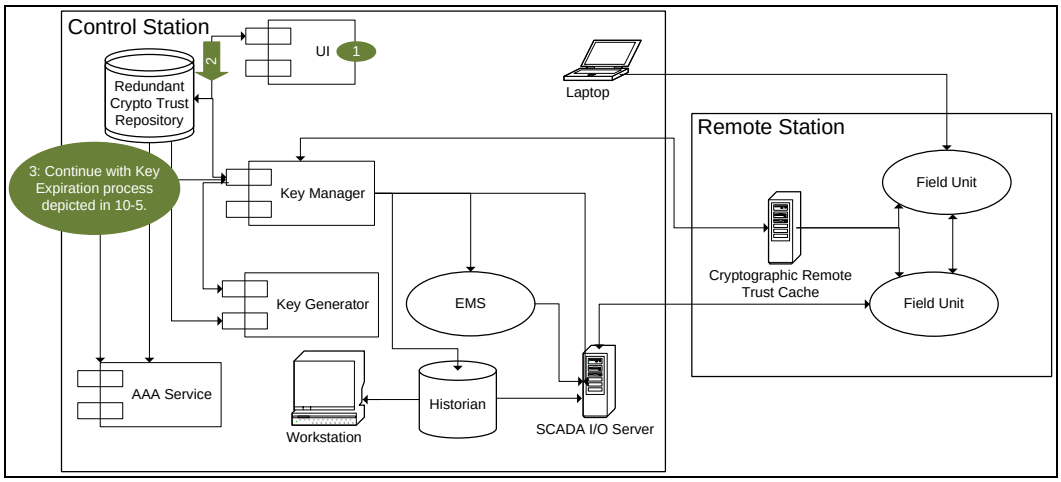

Figure 10.4. Key Revocation Process

1. A user uses the User Interface to select which key to revoke.

2. The UI communicates with the Cryptographic Trust Repository to update the metadata in order to expire the key material.

3. The process continues with Key Expiration process (10.5).

11.0 Internal Entity Trust Lifecycle

This section defines the trust lifecycle for employees of the local company. It describes all of the steps to create access roles, authenticate entities, and authorize access requests.

11.1 Role Creation

1. A user defines a role for a classification of employee.
2. A user attaches access rights for that role to devices.

11.2 Entity Credential Creation and Role Assignments

1. A user creates new employee credentials.
2. A user attaches roles to employee credentials.
3. Employee credentials are created and distributed to the entity.

11.3 Authentication and Authorization

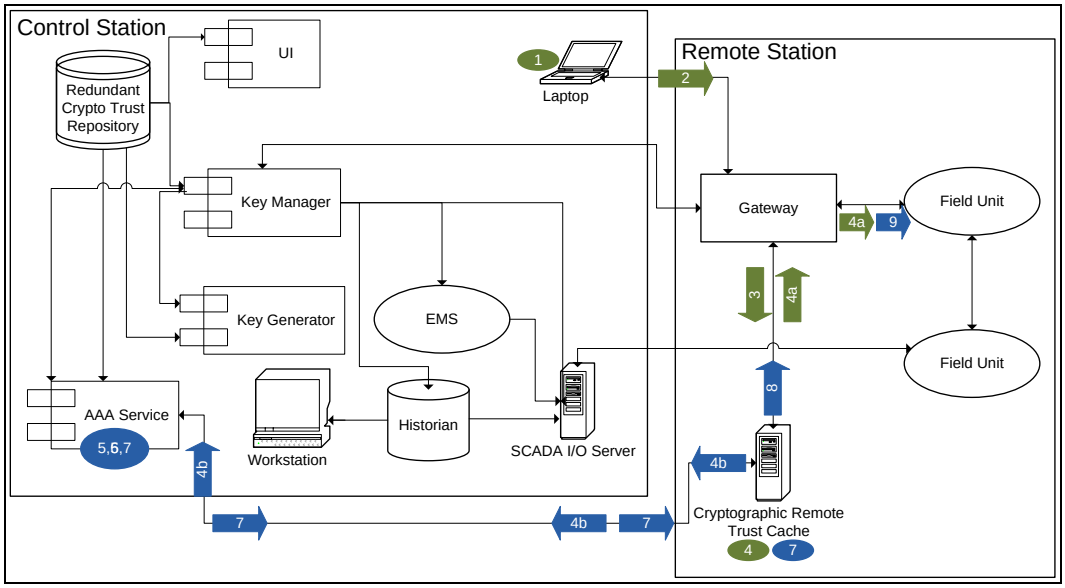

Figure 11.1. Authentication and Authorization Process

1. An entity begins connection to a device via a gateway device (e.g., modem, VPN, Terminal Server).
2. The entity passes credentials to the gateway.
3. The gateway sends credentials and access request to the Cryptographic Remote Trust Cache.
4. The Cryptographic Remote Trust Cache checks for a locally-cached authorization ticket for this credential.

a. If a ticket exists and hasn't expired, the Remote Trust Cache informs the Gateway to authorize the connection.

b. If no unexpired ticket exists, the Remote Trust Cache sends the user's identity credential to the AAA Service.

5. The AAA Service authenticates the entity credentials.
6. The AAA Service checks roles and policies to see if the entity should have access to this device.
7. The AAA Service creates the Kerberos-like ticket. The expiration-dated ticket is stored in the Cryptographic Remote Trust Cache.
8. The Cryptographic Remote Trust Cache informs the Gateway to authorize the connection.
9. Gateway opens a connection to device.

11.4 Role Change

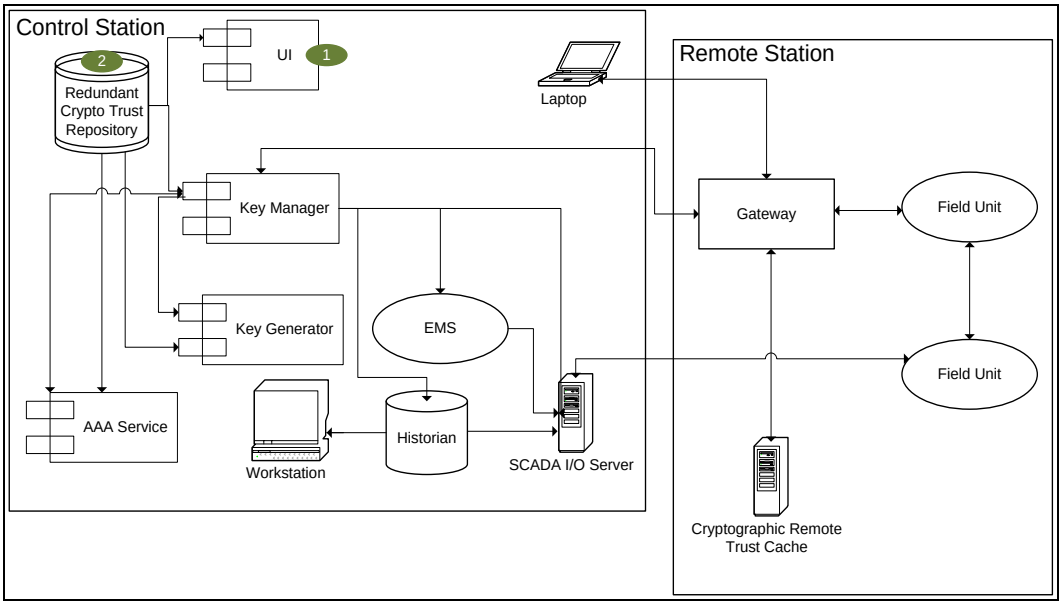

Figure 11.2. Role Change Process

1. Using the User Interface, the user changes the roles attached to entity credentials.
2. Changes are stored in the Cryptographic Trust Repository.
3. Upon a request to connect to a device during the Authentication and Authorization process (11.3), the AAA Service will not authorize the connection if the entity no longer has a role that allows access to the device.
4. An authorization failure will be communicated (pushed) back to the gateway device and the connection with the requesting party will be disconnected.

11.5 Credential Revocation

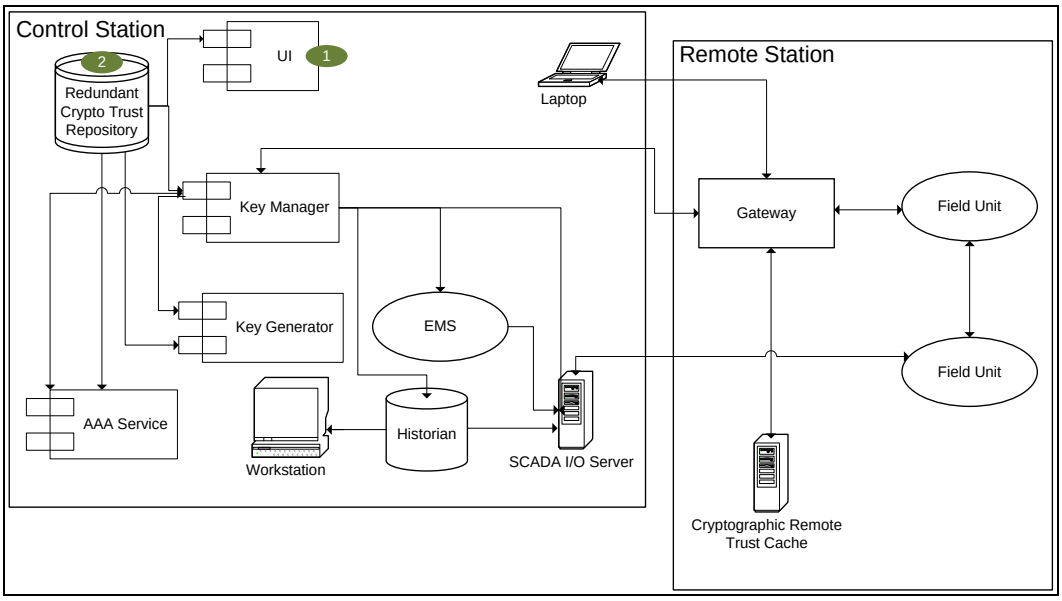

Figure 11.3. Credential Revocation Process

1. Using the User Interface, a user revokes entity credentials.
2. The AAA Service will no longer authenticate those credentials.

11.6 Credential Expire

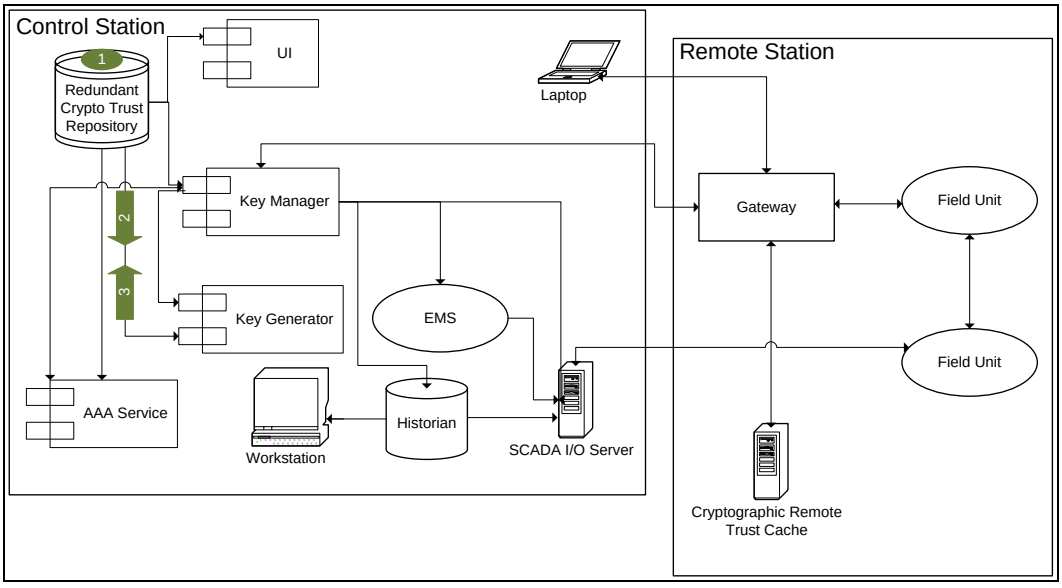

Figure 11.4. Credential Expire Process

1. The Cryptographic Trust Repository threshold is reached for a credential.

2. The Trust Repository requests new cryptographic material from the Key Generator.
3. On next authentication, the new cryptographic material is exchanged.

12.0 Third Party Trust Lifecycle

This section defines the trust lifecycle for employees from external companies. It describes all of the steps to create access roles, authenticate entities, negotiate trust, and authorize access requests.

12.1 Trust Configuration

The following steps will be performed by the local SCADA network administrator at each of the participating organizations.

1. For each third party that will be connecting to the local SCADA network, the local administrator uses the UI to configure properties related to the third party's AAA trust negotiation service.

    a. Administrator installs a public key certificate to be used for mutual authentication between the local and third party AAA trust negotiation services.

    b. Administrator configures connection details of the third party's AAA trust negotiation service: IP address, TCP port, etc.

2. Using the UI, the local administrator creates access roles (e.g. Field Unit Maintenance, ISO, etc) to be used by external entities, choosing from a set of pre-defined roles recommended by the specific process control industry.

3. The local administrator selects one or more valid access roles that will be considered valid for each third party organization.

4. The local administrator attaches access rights to devices and processes based on the access roles, for each device or process to be accessed by a third party. Rights are based on the {third party organization, access role} combination.

5. Using a trust negotiation policy language adopted by the specific process control industry (electric, water, etc), the administrator defines policies for trust negotiation, specifying the trust evidences required during trust negotiation for a given role and/or third party, the order in which the evidence must be presented, and the trust evidence that the local party is willing to share with the third party if their trust negotiation policy requires it. Trust evidence may include (among others):

    a. Role credential (e.g., Field Unit Maintenance, ISO, etc), dated and signed with the third party organization's private key

    b. Current Employee credential signed with the third party organization's private key and dated, testifying that the holder is a current employee in good standing.

    c. IP address within an expected range.

12.2 Authentication and Authorization

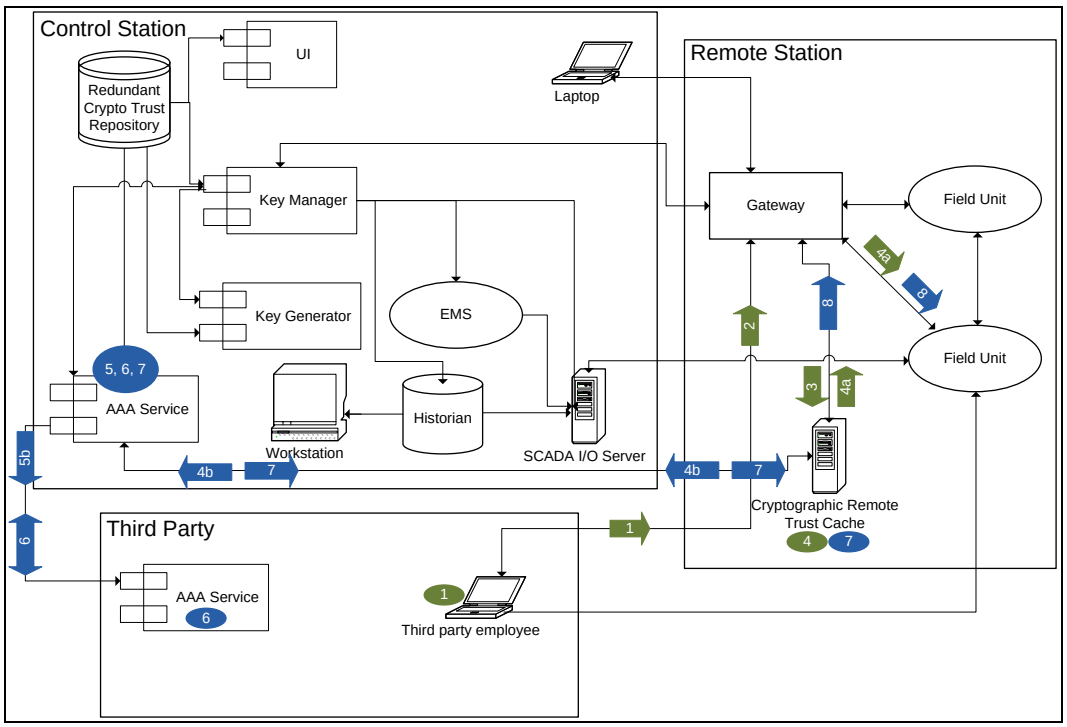

Figure 12.1. Third Party Entity Authentication and Authorization Process

1. External entity begins connection to device via a gateway device (e.g., modem, VPN, Terminal Server).

2. External entity passes their identity credential issued by their organization to the gateway.

3. Gateway sends the identity credential to the Cryptographic Remote Trust Cache.

4. Remote Trust Cache checks for a locally-cached authorization ticket for this credential.

    a. If a ticket exists and hasn't expired, the Remote Trust Cache informs the Gateway to authorize the connection.

    b. If no unexpired ticket exists, the Remote Trust Cache sends the user's identity credential to the local AAA Service.

5. The AAA Service determines whether the identity credential is local vs. third party.

    a. If the identity credential is from the local company, the AAA Service checks the Cryptographic Trust Repository to verify the employee's credential and role, and skips to step 7.

b. If the identity credential is from a third party, the AAA Service establishes a connection to the third party AAA Service.

6. The local AAA Service initiates a trust negotiation process per locally-configured policies governing access by this third party's employees. The remote AAA Service will also have policies governing its disclosure of sensitive credentials, so it may for instance require the AAA Service's signed credential before it will send any trust evidence. Example:

    a. The local AAA Service sends its own credential signed by the local company's private key.

    b. The local AAA Service sends the external user's remote identity credential to the remote AAA Service and requests a credential (time stamped within the last n days or hours) stating that the employee is a current employee of the company.

    c. Upon receipt, the local AAA Service requests all role credentials for the employee.

    d. If any of the trust negotiation or local checks fail, the access is disallowed by the Gateway.

7. The local AAA Service creates the Kerberos-like ticket. The expiration-dated ticket is stored in the Cryptographic Remote Trust Cache.

8. The Cryptographic Remote Trust Cache informs the Gateway to authorize the connection.

12.3 Role Change/Personnel Action

If an employee's role(s) change so they should no longer have access to a device or process at the local company's outstation gateway:

1. The administrator at the employee's company updates their Crypto Trust Repository to change the employee's role(s).

2. If this employee (or a masquerader) tries to connect to the local company's outstation gateway, the third party AAA Service will not provide to the local company's AAA Service a credential authenticating the employee for the former role.

3. An authorization failure will be pushed back to the Gateway and the connection with the requesting party will be disconnected.

If an employee leaves the company:

1. The administrator at the employee's company updates their identity management system to show that the person is no longer an employee.

2. If the former employee (or a masquerader) provides their old credential to the local company's outstation gateway, the third party AAA Service will not provide to the local company's AAA Service a credential authenticating the employee.

3. An authorization failure will be pushed back to the Gateway and the connection with the requesting party will be disconnected.

12.4 Credential Expire

1. Credential certificates have an expiration date.

2. All gateways and AAA Service components will check for the certificate expiration date for all credentials presented.

3. As part of the trust configuration process, a user is able to establish a time limit of how long a certificate can be allowed to exist.

    a. During the Authentication and Authorization process (7.2) at step 6 (trust negotiation), the AAA Service should check to make sure all credentials offered during the trust negotiation process comply with the local credential date policy. If not, the AAA Service rejects the certificate and sends an authorization failure.

    b. Usage of authorization tickets retrieved from the Cryptographic Remote Trust Cache shall also be constrained by the local credential date policy.

13.0 Security Network Communication Architecture

13.1 Introduction

A security network is any number of nodes connected together to produce a system that transmits and routes security related data between the interconnected nodes. It is designed to provide assurance that the mission of the facility or organization is maintained and to limit damages from accidental or intentional disruptions. While defining a security network is not within the scope of this project, it is a concept worth discussing in the context of the CTM system defined in this document. A separate and distinct security network is recommended for the operation of the Cryptographic Trust Management System. This section will define the concept of a security network and why it is valuable.

A security network, in some ways, is analogous to a fire suppression system. It is put in place as a protective measure. Fire suppression systems operate independently of the systems they are designed to protect while at the same time monitoring said system. If the fire suppression system becomes damaged or is offline, the critical processes it is protecting should not be impeded. Additionally, if there is not a fire while the system is down, there are minimal consequences. Similarly, the security network is a protective measure. It protects against accidental and intentional computer-based attacks. It too should fail without impacting the mission of the process it is in place to protect. Furthermore, the data that traverses the security network is independent of the mission critical process data. The security network should be segregated from the control system network to ensure that problems in one network do not affect the other. The security network should operate parallel to the control system network with hooks to allow it to monitor the control system.

Like fire suppression systems that carry water, halon, or other fire extinguishing chemical agents, the security network may carry many different types of data. Cryptographic key material for encryption and/or authentication, logs for audit, forensics and troubleshooting purposes, system configurations, users' rights and permissions, group password changes, and IDS/firewall alerts are all examples of types of data that may traverse a security network to help secure the control systems network. This data is distinct from control systems operational data and thus needs to be treated differently.

13.2 Security Network Value

There are three key reasons why a separate and distinct security network is needed. The first is that the security network and the process control network have different requirements and objectives. The second is that the skill set required to manage each network varies. Lastly, many of today's control system networks are simply not capable of handling the amount and types of traffic required for a security network.

The principle objective of a control system network is to assure that a certain process occurs and to provide data that will ensure its continued operation in a safe and reliable manner. The control system network provides core operational functionality. In contrast, the principle objective of the security network is to monitor the health of the control system network and to manage the security mechanisms that enable the control system network to accomplish its main objective. The security network provides

an ancillary function to support the core business processes. The security network does not contribute directly to any business process but instead tries to protect the business processes from hazards and threats.

The operational requirements and objectives of each network vary which makes putting them together on the same network difficult and imprudent. Just as a plant operator would not send oil and water through the same pipeline although they may need to go to the same area of the plant, security data and operational data should not be put on the same network. They have very different objectives and requirements. The following table highlights some of the differences.

Table 13.1. Comparison of Process Control Network and Security Network Features

	Process Control Network	Security Network
Lifespan	Short: In many systems, the valuable lifespan of the data is on the order of milliseconds to possibly minutes. Much of the process data is used to assess the status of the system at the exact moment it is collected. For example, a power system operator trying to maintain a frequency of 60 Hz at 3:35 P.M. only needs to know the frequency at 3:35 P.M.; it is irrelevant what the measurement was at 3:30 P.M.	Long: Most of the data transmitted through a security network has a useful life span of months to years. For example, cryptographic keys must remain valid throughout their replacement lifecycle. This could be six months, one year, or longer. It is also beneficial and sometimes mandated to store security logs for many years.
Tolerance for Delay	Low Tolerance: The data used to monitor and control processes is used in real-time on physical real-world processes. Any delay or out of order information will have physical repercussions.	High Tolerance: If security data is delayed (e.g., a new cryptographic key takes a few minutes or even hours longer to update than expected, or a log entry fails to reach the central repository for an extended period) there is little, if any, impact to the overall system
Tolerance for Bad or Unrecognized Data	Low Tolerance: Many devices do not know how to respond to bad or unrecognizable data. Bad data has been known to cause devices to fail or act erratically.	High Tolerance: One purpose of the security network is to detect and respond to bad and unrecognized data.
Confidentiality	Low priority: The data is useful for such a short period of time that ensuring its confidentiality is often considered unnecessary.	High Priority: The data traveling on this network is highly sensitive and confidentiality has a very high priority.

Table 13.1. (contd)

	Process Control Network	Security Network
Integrity	High Priority: Industrial control systems generally operate on an implicit trust paradigm and the data is accepted by the end devices as legitimate. Though this attribute has a higher priority than confidentiality, there are not many commercial technologies that enable devices to accurately identify themselves and verify the integrity of the messages on the network.	High Priority: The sensitive nature of the data on this network mandates assurance that the data is valid and from a trusted source.
Availability	High Priority: The availability of process data has the highest priority. Even brief outages can create significant losses. As a result, many industrial control systems are designed with complete redundancy to eliminate any single points of failure.	Low Priority: Security networks can generally recover from brief outages without suffering significant consequences.
Quantity	Low to Moderate: The quantity of data that moves through a control system is often very small when compared to IT systems.	Moderate to High: Depending on the types of applications supported, the quantity of data that moves over a security network could be moderate to high. Intermittent key management data alone would only require moderate amounts of data, but centralized logging would require high amounts of data.
Disparity of Data	Limited: Any individual link within most legacy systems carries only one data type or protocol.	Expansive: Individual links could carry many disparate data types.

The skill sets required to manage an operational control system network are different than the skills necessary to run a security network. In most cases, the IT department has the responsibility to perform the security operations on the enterprise network that are also needed to secure the control system network via the security network. Instead of further burdening the control system engineers and operators with new responsibilities, the IT department may take on many of these security responsibilities.

Finally, the need for a separate security network becomes apparent by looking at the existing networks. Many of the existing network links within control systems are point-to-point serial or 900 MHz wireless connections that are incapable of handling additional traffic. Adding additional data types or capacity to these networks is difficult to impossible.

13.3 Assumptions and Considerations

The security network must have the capability to handle multiple data types and be able to deliver that data to multiple destinations. Data must be able to get from one node in the network to any other node within the network. Thus it is assumed that the network will use routable protocols and technologies.

Local area network communications are assumed to be IP-based because of its low cost, ubiquity, and because it is well understood by many people. It is also assumed that the security network will be a physically and logically separate network from the control system network. In some cases it may be acceptable to use the same physical network (e.g., fiber optic) if there is sufficient bandwidth and logical separation such that one will not impact the other.

It is assumed that backhaul communications will utilize either the Internet or a third party communications network. The cost of planning, deploying, and maintaining a separate network for the security traffic is assumed to be too cost prohibitive to be acceptable in most cases. Given that reliability requirements for the security network are not as stringent as those of a control system network, the Internet or other public networks are practical solutions.

13.4 Security Network Architecture

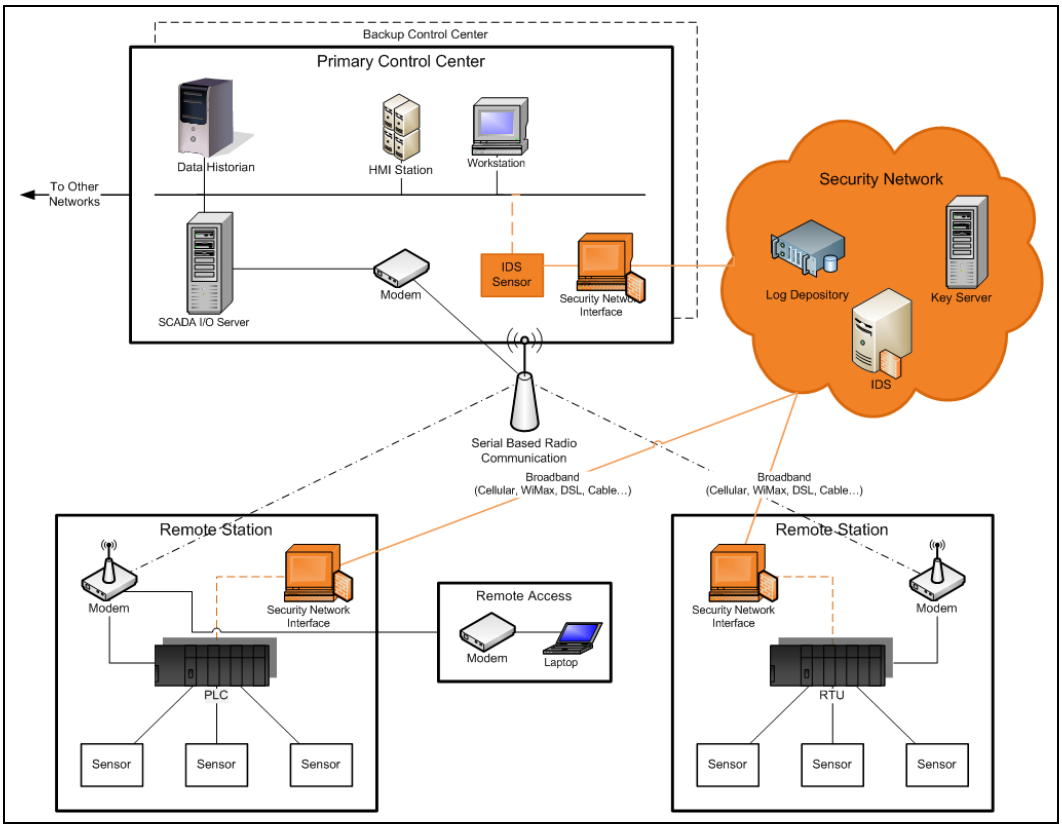

Figure 13.1. Architecture of a Security Network within a Control System Network

The security network, shown in Figure 13.1 in orange, extends into all areas of a control system network. Network links between the security network and all other locations need to be of a broadband (384 Kbps or greater) quality to provide adequate throughput of security traffic. The ability to reach into the remote locations can be accomplished using various technologies that have only become readily available and economically feasible in the past few years. WiMax, 3G/4G Cellular, and extended DSL and cable systems now provide broadband network connectivity where before only leased lines or privately built networks could reach.

13.5 Security Attributes

The purpose of the security network is to enhance the security posture of the control system network and to manage the mechanisms required to do this. Though it should never have the capability to control any devices on the control system network, the security network has full visibility into the control system network and transmits logs and cryptographic material that could be very damaging if not properly secured. As a result the security network itself must be secured.

The recommendation of this document is a defense-in-depth approach. Defense-in-depth is a methodology of layering security controls such that if one layer fails, the impact is confined and the whole network is not compromised. The following technologies will help provide the defense-in-depth.

- VPN – Virtual private networks create encrypted tunnels between two end points. They should be used for any connections traversing a network external to the organization and in some cases between end points within the organization. VPNs can also be configured to deny rerouting of traffic to other networks.
- Firewalls – By forbidding any and all traffic that isn't necessary, a properly configured firewall greatly limits the threats that can cause damage to a network.
- Mutual Network Authentication – Many systems only require server side authentication. Transport Layer Security (TLS) is used for most secure Internet transactions and generally only requires server side authentication, i.e., the server certifies that it is legitimate but the client does not have to certify itself. As such, it is susceptible to man-in-the-middle attacks (MITM). Ensuring that both nodes authenticate to each other will mitigate MITM attacks.
- MAC Address Filtering – Every network interface card has a unique MAC address. MAC address filtering allows only known MAC addresses to communicate on the network.

14.0 Device Registration and Provisioning

14.1 Authentication Credentials

The Cryptographic Trust Management System relies on the ability to uniquely identify and authenticate devices for the management of cryptographic material. Therefore, it is necessary that all devices integrating with the CTM System must have a set of credentials and capabilities to interoperate with the system. The following table details the elements needed for authentication:

Table 14.1. Elements Required to Integrate with Cryptographic Trust Management System

Element	Requirement	Recommendation
Initial Identity Credential	Device will be created and shipped with a unique, secure, and secret certificate. Cannot be modified after creation.	The device manufacturer should securely bind the Initial Identity Credential into each device. The security of the private key must be maintained at all costs.
Local Identity Credential	The device can create additional signatures for signing and security as needed. The Local Identity credential is used for authentication within the CTM System.	The Initial Identity Credential can be used to create Signature Credentials. The Initial Identity should NOT be used directly to sign communication. The private key must be stored and managed securely.
Storage	Sensitive and protected information must be protected from physical and electronic access.	Local storage must be protected by tamper resistant engineering to meet or exceed FIPS 140-2 level 2. Remotely stored information must be encrypted and stored securely.
Asymmetric Cryptography	Signatures are required to authenticate communicating parties. Asymmetric Cryptography Public Key encryption will be used for this.	Cryptography used must be at least equivalent to 2048-bit RSA. Cryptographic keys are recommended to be at least equivalent to 256-bit ECDSA.
Hashing	Hashing is used to prevent passwords and keys being sent in communication. Hashes are also used in the creation of digital signatures.	Implemented hashing algorithms must be at least cryptographically equivalent to SHA-256 (as detailed in FIPS 180-2).
Random Number Generator	Random numbers are used to ensure cryptographically unique signatures and keys. Sufficient Random Number Generators may be required for FIPS validation.	See FIPS 140-2 Annex C for details on FIPS approved Random Number Generators. Currently only deterministic number generators have been FIPS approved.
Digital Signatures & Signing	For authentication, devices and servers need to mutually authenticate. Digital signatures will be used to verify identity.	DSA or ECDSA 256-bit size keys are recommended for all signatures used (as detailed in FIPS 186-2).

14.2 Provisioning/Initialization Process

Due to the potentially unprotected locations of outstation equipment, it is recommended that devices are created following Trusted Platform Module (TPM) and the forthcoming IEEE 802.1AR standards. An endorsement key should be securely installed upon device creation. This endorsement key must be unique, exist for the lifetime of the device, and be unchangeable. This public key Initial Identity Credential must be at least cryptographically equivalent to a 2048-bit RSA key, and a 256-bit ECDSA is highly recommended. The secrecy and security of the private key portion of the Initial Identity Credential must be maintained at all times. Any device implementation must make sure that the private key portion is not readable by anyone other than the trusted portion of the device. Additionally, the private key portion must never be written to after creation.

The Initial Identity Credential should only be used to verify ownership of the device and in the creation of Local Identity Credentials. The Initial Identity Credentials should be securely stored, cryptographically unique, bound to one specific device, and difficult to duplicate or forge. The newly created Local Identity Credentials are used in the Cryptographic Trust Management System authentication process. Additional Local Identity Credential signatures can be created and used as dictated by security and use policy.

14.3 Registration Process

The Registration Process occurs when a device is installed or brought online in a network setting. The device owner must possess the Initial Identity Credential of the device to be the owner. This key must be kept completely secure at all times. If remote owner access and control is required, the Initial Identity Credential key must be encrypted, transported and stored offsite with up-to-date industry standards. This key can then be used to authenticate to the device as the owner for complete control. If owner level access is not required, the Initial Identity Credential will be used to create a Local Identity Credential (as described in the Provisioning/Initialization Process (Subsection 14.2)). This key will allow the device to communicate and authenticate with centralized servers located offsite as well as other devices locally or offsite.

To register a device with the Cryptographic Trust Management System, the security administrator responsible for the device is tasked with recording the Initial Identity Credential and Local Identity Credential keys and identifying information for an off-line or subsequent manual enrolling of the device in the network setting. Using the User Interface component's web portal, the security administrator must install the device's Local Identity Credential. For more discussion on this process, refer to Subsection 9.1 of "User Interface Component."

Alternatively, an automated solution may be possible due to the use of public key cryptography. Devices could perform a special one-time authentication process with the AAA Service component to enroll in the system. This procedure would be automated and start when a device is successfully plugged in and powered on. The device would perform an initial authentication exchange operation with the AAA Service component by providing a one-time Local Identity Credential (the Initial Identity must NOT be used in this procedure). The AAA Service component would acknowledge the device and enroll it in the Cryptographic Trust Repository. Thereafter, all subsequent authentication operations would perform as usual. For discussion of the authentication process, refer to Subsection 7.2 of "AAA Service Component."

14.4 Hardware Protections

Devices deployed at remote locations or unsecure locales should follow FIPS 140-2 Security Level 3. Devices should not only be tamper-evident, but also should prevent physical intruders from gaining access to critical security parameters. The devices should have a high probability of detecting and countering any intrusion. All devices should be validated to ensure that protected information cannot be gleaned from a physical intrusion or hardware analysis.

15.0 Third Party AAA Service Requirements

15.1 Introduction

It is not expected nor assumed that all organizations will have a complete Cryptographic Trust Management System. However, for the CTM System to work with the inter-organization role-based access control, all cooperating organizations must have a minimum set of capabilities to enable the CTM System to work. This section defines the minimum services a third party would need to interoperate with the CTM System.

15.2 AAA Service

The first service necessary is a AAA Service that can communicate with the third party trust negotiation protocol, understands the trust policy language, and can perform the authentication and authorization processes. AAA Service is needed in order to complete the trust negotiation process. This service does not have to meet the exact requirements that the AAA Service component is designed against but it does have to meet the three requirements indicated in this paragraph.

15.3 Identity Management

The second service an organization must have to work with the Cryptographic Trust Management System is an identity management system. The employees requesting access to the CTM System must have credentials to be authenticated. These credentials must also specify the organization of which they are a representative. Also, the identity management system must be able to authenticate employees of the organization. The organization does not have to use the identity management system for all of its employees but it must at a minimum provide identity management for employees requiring access at sites utilizing a CTM System.

15.4 Role Mapping Service

The last service required is a system for mapping employee identities to one of the industry-defined roles. The roles must be predefined so that every organization has an understanding of what job functions fit into a role and what access privileges can be assigned based on those roles. Since the roles are relied upon to authorize access, an organization wishing to interoperate must have a method of mapping employee identities to one or more industry-defined roles. With this capability, the organization will be able to respond to the AAA Service trust negotiation with the roles that the employee fills. Without this role mapping, the trust negotiation process will never successfully complete because the Cryptographic Trust Management System will never receive a valid role to authorize against.

16.0 Legacy Retrofitting

16.1 Introduction

The functionality and capabilities designed into the Cryptographic Trust Management System are new and advanced for current process control networks. If a commercial CTM System were available today, it would have limited usefulness because the equipment installed in process control networks is unable to leverage the more advanced aspects of the system. To make the CTM System work with current systems and start an integration path, devices will need to be created and installed that retrofit the equipment with the required capabilities.

The first and most obvious missing component is the security applications. One of the principal drivers for this system is the reluctance to integrate security applications into process control networks. It is understood that the operational burden of managing security applications is significant. The CTM System was designed to alleviate much of that burden. Therefore, the security solutions being developed for this environment need to start being deployed. The following subsections will describe possible methods of integrating security applications into a current process control network.

16.2 AAA Service

It would not be too arduous to retrofit current process control networks to enable AAA Service. The end goal is for every device with a management port to tie into the AAA Service for access authorization. However, a simple intermediary step would be to install a routable gateway device in the outstation that entities would use to connect to devices. This gateway device would be similar to a VPN or terminal services switch. When an entity connects to the gateway it will communicate with the Cryptographic Remote Trust Cache to perform authentication and authorization. After receiving authorization, the gateway would allow a connection to pass through to the port of the device the entity wants to access.

There are current devices on the market that already perform similar functions. This new type of gateway would simply require some type of protocol to interrogate a connecting entity for their credentials and what device they want to access. Also, the device would require interoperability with the AAA Services EAP-IKEv2 and Kerberos hybrid protocol.

16.3 Key Management

As was stated in the introduction of this section, the main requirement to enable key management capabilities is applications that require key management. Current security applications will work with the Cryptographic Trust Management System via the manual key management capabilities of the User Interface component. However, additional requirements for equipment are necessary for it to work with the automated key management functionality.

The simplest method of retrofitting a process control network is to install bump-in-the-wire link encryptors and authenticators. These devices sit in the path of a communication channel to add a layer of security to communication as it passes through the wire. With the manual key management process, cryptographic material can be generated and managed by the CTM System and manually installed into the

devices. The CTM System provides secure backup of the cryptographic material and notifications when the cryptographic material needs to be updated. It also provides management oversight for policy violations.

16.3.1 Automation

To enable the automated key management capabilities of the CTM System, the bump-in-the-wire devices must have the ability to communicate over a routable network and be updated to support the subset of KMIP defined in this document. The device will also need to be able to authenticate to the system via the hybrid EAP-IKEv2 and Kerberos protocols. Also, security devices located in an outstation should have additional hardware requirements to protect the integrity of the application. For recommendations on outstation equipment, refer to “Device Registration and Provisioning” (Section 14.0).

17.0 References

Adams, C., & Lloyd, S (1999). *Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations*. Indiana: New Riders Publishing.

Barker, E., Branstad, D., Chokhani, S., & Smid, M. (2010). *Cryptographic Key Management Workshop Summary – June 8-9, 2009.* (NIST Interagency Report 7609). Washington, DC: Government Printing Office.

Barker, E., Barker, W., Burr, W., Polk, W., & Smid, M. (2007). *Recommendation for Key Management – Part 1: General (Revised)* (NIST Special Publication 800-57). NIST.

Barker, E., Barker, W., Burr, W., Polk, W., & Smid, M. *Recommendation for Key Management – Part 2: Best Practices for Key Management Organization* (NIST Special Publication 800-57). NIST.

Barker, E., Burr, W., Dang, Q., Jones, A., Polk, T., & Rose, S. (2009). *Recommendation for Key Management – Part 3: Application-Specific Key Management Guidance* (NIST Special Publication 800-57). NIST.

Brown, B., et al (2008). *AMI System Security Requirements* (UCAIUG: AMI-SEC-ASAP). AMI-SEC Task Force.

The Cyber Security Coordination Task Group (2009). *Smart Grid Cyber Security Strategy and Requirements* (DRAFT NISTIR 7628). NIST.

Dierks, T. & Rescorla, E. (2006). *The Transport Layer Security (TLS) Protocol Version 1.1* (RFC 4346).The Internet Society.

Edgar, T. (2009). *Cryptographic Trust Management Requirements Specification Version 1.0.*

Eisenhauer, J., Donnelly, P., Ellis, M. & O'Brien, M. (2006). *Roadmap to Secure Control Systems in the Energy Sector*. Sponsored by U.S. Department of Energy and U.S. Department of Homeland Security.

Falco, J., Scarfone, K., & Stouffer, K. (2008). *Guide to Industrial Control Systems (ICS) Security* (NIST Special Publication 800-82). NIST.

Forsberg, D., Patil, B., Tschofenig, H., & Yegin, A. (2008). *Protocol for Carrying Authentication for Network Access (PANA)* (RFC 5191).

Housely, R., and J. Schaad (2002). *Advanced Encryption Standard (AES) Key Wrap Algorithm* (RFC 3394). The Internet Society.

The Institute of Electrical and Electronics Engineers, Inc. (2005). *IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control* (IEEE Std 802.1X™ – 2004). New York: IEEE.

Instruments, Systems, and Automation Society (2009). *Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program* (ANSI/ISA–99.02.01–2009). ISA.

Instruments, Systems, and Automation Society (2007). *Security Technologies for Industrial Automation and Control Systems* (ANSI/ISA-TR99.00.01-2007). ISA.

Rukhin, A., et al (2008). *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* (NIST Special Publication 800-22 Revision 1). NIST.

National Institute of Standards and Technology (2001). *Security Requirements for Cryptographic Modules* (FIPS PUB 140-2). Washington DC: Government Printing Office.

North American Electric Reliability Corporation Standards Committee (2009). *Loss of Control Center Functionality* (Standard EOP-008-1). NERC.

Oasis Key Management Interoperability Protocol Technical Committee (2009). *Key Management Interoperability Protocol Profiles Version 1.0* (kmip-profiles-1.0-cd-04). Oasis.

Oasis Key Management Interoperability Protocol Technical Committee (2009). *Key Management Interoperability Protocol Specification Version 1.0* (kmip-spec-1.0-cd-06). Oasis.

Oasis Key Management Interoperability Protocol Technical Committee (2009). *Key Management Interoperability Protocol Usage Guide Version 1.0* (kmip-ug-1.0-cd-05). Oasis.

Oasis Key Management Interoperability Protocol Technical Committee (2009). *Key Management Interoperability Protocol Use Cases Version 1.0* (kmip-usecases-1.0-cd-05). Oasis.

Schneier, B., (1994). *Applied Cryptography*. New York: Wiley.

Appendix A

Requirements Traceability

Table .1. Requirements Traceability Matrix

ID	Requirement ID	Functional Requirement	Architectural Component
001	1.1.1	Must provide functionality to negotiate trust with external entities to; enable external entities, down to the granularity of device and/or person, and securely communicate with internal systems and devices by exchanging the necessary session key material upon verification of external parties meeting industry/enterprise specific requirements.	AAA Service
002	1.1.1	The verification process will check, at a minimum, the validity of cryptographic material in regards to employment actions, expiration, and revocation.	AAA Service
003	1.1.2	Must automate the key management of device-to-device communication.	Key Manager
004	1.1.2	Must provide manual key management to support legacy equipment that does not support the CTM System.	Key Manager
005	1.1.3	Must provide a centralized location to manage cryptographic material and trust relationships.	Trust Repository
006	1.1.4	Must provide secure storage for cryptographic material via encryption and role based access.	Trust Repository
007	1.1.5	Must securely generate cryptographic material by meeting all applicable NIST standards.	Key Manager
008	1.1.6	Must alert when cryptographic material expires and new cryptographic material has not been distributed to all affected devices and/or people.	Trust Repository
009	1.1.6	For manually managed cryptographic material, the CTM System must also alert a user configurable percentage of the lifespan of the cryptographic material.	Trust Repository
010	1.1.7	Must provide the ability to revoke cryptographic material	Key Manager
011	1.1.8	Must archive old cryptographic material for user configured time lengths and allow for retrieval.	Trust Repository
012	1.1.9	Must support device registration, both manually and automatically, to the CTM System.	Device Registration
013	1.1.10	Must support the configuration of priority of cryptographic material for degradation of failure. If the system begins to fail or is unable to perform all of the tasks in a needed time frame, it will complete the functions involving the cryptographic material with the highest priority first.	Key Manager
014	1.1.11	Must support back-up of cryptographic material and configuration information capabilities	Trust Repository

ID	Requirement ID	Functional Requirement	Architectural Component
015	2.1.1	Must support separation of roles and role-based access for access to cryptographic material. (Access to cryptographic material shall be by role and may be configured to require multiple passwords, each representing a user from a different role.)	Key Manager
016	2.1.1	Must support separation of roles and role-based access for access to administrative configuration functionality. (Access to administrative configuration functions shall be by role and may be configured to require multiple passwords, each representing a user from a different role.)	User Interface
017	2.1.1	Must support separation of roles and role based access for access to reports. (Access to reports shall be by role and may be configured to require multiple passwords, each representing a user from a different role.)	User Interface
018	2.1.1	Provide administrative interface for configuration and assignment of roles	User Interface
019	2.1.1	Provide database support for storage of roles and users assigned to roles	Trust Repository
020	2.1.2	Must provide a web-based graphical user interface for configuration and management of key policies, device cryptographic material, cryptographic material retrieval, and key revocation.	User Interface
021	2.1.2	Must provide a web-based graphical user interface for configuration and management of external party negotiation policies.	AAA Service Requirements
022	2.1.2	Must provide a web-based graphical user interface for configuration and management of report information.	User Interface
023	2.1.3	Must utilize role-based access control for cryptographic material retrieval based on organization prescribed role.	Trust Repository
024	2.1.3	Must utilize role-based access control for configuration based on organization prescribed role.	User Interface
025	2.1.3	Must utilize role-based access control for report access based on organization prescribed role.	User Interface
026	2.1.3	Allow creation of new roles	User Interface
027	2.1.4	Construct a key metadata report for auditors that includes lifecycle information, bit strength, and devices utilizing the key	Key Manager
028	2.1.4	Without disruption of new keys, construct a policy violations report for auditors that includes keys expired, systems requests for inappropriate cryptographic material, and request for communications between devices/people without proper privileges.	User Interface
029	2.1.5	Must provide support for forensic analysis by maintaining a history of which devices have cryptographic material and when cryptographic material was distributed.	Key Manager

ID	Requirement ID	Functional Requirement	Architectural Component
030	2.1.5	Must provide support for forensic analysis by maintaining a history of which devices/people requested to communicate and to which device.	AAA Service
031	2.1.5	Must provide support for forensic analysis by maintaining a history of which devices/people requested to communicate and to which device.	Trust Repository
032	2.1.5	Provide functionality in the user interface to retrieve the forensic information based on user criteria (i.e., which device, time frame, and key material they are interested in).	User Interface
033	2.1.6	Must provide a user configurable alarming mechanism to alert user to expired cryptographic material, unknown devices, failed trust negotiations, or other erroneous events.	User Interface
034	2.1.7	Must provide a manual cryptographic management interface to support legacy devices that are not compatible with the CTM System. This interface should provide the capability to create key material and associate it with a device	Key Manager
035	2.1.8	Must provide a report system to report the cryptographic status (what devices are they communicating with and cryptographic material lifecycle metadata) of the all devices and systems managed and any system or cryptograph errors that have occurred. The report system must provide audit, forensic, system health, and system performance views	User Interface
036	2.1.9	Must support hardware interfaces for the use of hardware tokens or LUNA modules for the identification of the key authority and also for the manual distribution of key material.	AAA Service
037	2.2.1	CTM System must communicate directly via a standard protocol with devices or vendor-specific product configuration software for key management functions (key requests, communication requests, and key expiration/revocation).	Key Manager
038	2.2.2	Must interface with CTM Systems at other utilities or <provide own interface-compatible> cryptographic trust management services for trust negotiation.	AAA Service
039	2.2.3	In addition to offering custom identity management capabilities, CTM System will be able to utilize existing key management infrastructure (at a minimum the Entrust PKI) to access identity cryptographic material and metadata	Key Manager
040	2.3.1	Must operate in an automated fashion with compatible devices, after initial configuration, to maintain the lifecycle of cryptographic material (key update, distribution, request, expiration, and revocation).	Key Manager

ID	Requirement ID	Functional Requirement	Architectural Component
041	2.3.2	Both user and device cryptographic material can be managed by CTM System.	Key Manager
042	3.1.1	CTM System shall NOT be limited to a specific transmission medium	All
043	3.1.2	CTM System shall NOT be limited to a single platform or vendor.	All
044	3.1.3	CTM System shall NOT be limited to a specific cryptographic method / technique	Key Generator
045	3.1.4	CTM System components shall NOT duplicate existing commercial solutions	All
046	3.1.5	CTM System shall NOT be industry/sector specific.	All
047	3.1.6	CTM System shall be designed to meet FIPS 140-2, 180-3, 186-3 and other applicable standards. The standards identified by the NIST Interoperability Framework for the Smart Grid will be reviewed and addressed by the CTM design. However, formal validation against these standards is out of scope.	Key Manager
048	3.1.7	CTM System shall be designed to meet industry standard entropy requirements. IETF RFC 4086 shall be referenced during design and only FIPS approved pseudo-random number generators will be used. NIST 800-22 and FIPS 140-2 statistical tests shall be utilized to ensure entropy levels are met.	Key Manager
049	3.1.8	Hardware and key storage solutions must support key history for at least three years (NERC CIP-002-01 D.1.3).	Trust Repository
050	3.1.9	CTM System must scale to support as low as 10 up to 100,000,000 entities	Trust Repository
051	3.1.9	CTM System must scale to support as low as 10 up to 100,000,000 entities	Key Manager
052	3.1.9	CTM System must scale to support as low as 10 up to 100,000,000 entities	AAA Service
053	3.1.10	Asymmetric keys shall be supported for user identity.	AAA Service
054	3.1.10	Symmetric keys shall be supported for device identity.	AAA Service
055	3.1.10	Asymmetric keys shall also be supported for device identity for devices that have adequate resources	AAA Service
056	3.1.10	Asymmetric keys shall be supported for user identity.	Trust Repository
057	3.1.10	Symmetric keys shall be supported for device identity.	Trust Repository
058	3.1.10	Asymmetric keys shall also be supported for device identity for devices that have adequate resources	Trust Repository
059	3.1.10	Asymmetric keys shall be supported for user identity.	Key Manager
060	3.1.10	Symmetric keys shall be supported for device identity.	Key Manager
061	3.1.10	Asymmetric keys shall also be supported for device identity for devices that have adequate resources	Key Manager
062	6.1.1	CTM System securely and confidentially performs all key management functions	Key Manager
063	6.1.2	Key material is securely stored according to best practice methodology	Trust Repository

ID	Requirement ID	Functional Requirement	Architectural Component
064	6.1.3	CTM System will only allow a minimal attack surface through available port reduction	All
065	6.1.4	CTM System correctly performs all cryptographic functions	Key Manager
066	6.1.5	FIPS approved cryptographic algorithms will be utilized by CTM System to develop keys, protect keys, and communicate through all external <CTM> interfaces	All
067	6.1.6	CTM System will ensure unique keys are utilized across systems and networks <within a security domain>	Trust Repository
068	7.1.4	Must provide, as applicable, mechanisms and procedures for failover to redundant site	Trust Repository
069	9.1.1	The CTM System solution shall provide online help and guidance to assist in proper configuration and maintenance	User Interface
070	UC A.1	Request key material (between applications, systems, or devices)	Key Manager
071	UC A.1	Request key material	Trust Repository
072	UC A.2	Request communication key (between applications, systems, devices, and people)	Key Manager
073	UC A.3	Expire key material (for application, system, or device) through GUI or API	Key Manager
074	UC A.4	Negotiate trust session (external application, system, or device)	AAA Service
075	UC A.5	Revoke key material for user, application, system or device through GUI or API	Key Manager
076	UC A.6	Provision system -- initial configuration	User Interface
077	UC A.7	Configure third-party trust relationship through GUI or API	User Interface
078	UC A.8	Retrieve key information for auditing and forensic purposes	Trust Repository
079	UC A.8	Retrieve key information for auditing and forensic purposes	User Interface
080	UC A.9	Manual key material request (by user for an entity that cannot request it automatically)	Key Manager

Appendix B

Glossary

Accounting – Refers to the tracking of the consumption of network resources by users. This information may be used for management, planning, billing, or other purposes.

Application – An application is a set of functionalities for accomplishing some task.

Asymmetric Cryptography – Cryptographic technique that uses a different secret key for the encryption and the decryption transformations.

Audit Logs – A record of both completed and attempted accesses and service.

Authentication – Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.

Authorization – The process of granting a person, computer process, or device access to certain information, services, or functionality. Authorization is derived from the identity of the person, computer process, or device requesting access, which is verified through authentication.

Availability – The property of a system or resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system; i.e., a system is available if it provides services according to the system design whenever users request them.

Cache – A collection of data duplicating original values stored elsewhere or computed earlier, where the original data is expensive to retrieve or compute (owing to longer access time), compared to the cost of reading the cache.

Certificate – An encrypted file containing user or server identification information, which is used to verify identity and to help establish a security-enhanced link.

Control Station – A centralized location that provides monitoring and control capabilities. An example is a control center for an electrical utility.

Cryptographic Material – Cryptographic material represents keys, certificates, or any other data that is input to cryptographic algorithms.

Cryptography – The study or analysis of codes and encoding methods used to secure information. Cryptographic techniques can be used to enable and ensure confidentiality, data integrity, authentication (entity and data origin), and non-repudiation.

Decryption – Decryption is the process of reversing encryption or transforming encrypted data or ciphertext back into plaintext.

Device – A device is an embedded system that operates without the regular human intervention. RTUs, PLSs, and IEDs are considered devices.

Diffie-Hellman Key Agreement – A cryptographic protocol which allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Digital Signature – Data appended to, or a cryptographic transformation of, a data unit that allows the recipient of the data unit to prove the origin and integrity of the data unit and protect against forgery, (e.g., forgery by the recipient).

Encryption – Cryptographic transformation of data (called "plaintext") into a form (called "ciphertext") that conceals the data's original meaning to prevent it from being known or used. If the transformation is reversible, the corresponding reversal process is called "decryption," which is a transformation that restores encrypted data to its original state.

Entity – An entity represents a person with credentials.

Entropy – A measure of the uncertainty associated with a random variable.

Extensible Authentication Protocol – A universal authentication framework frequently used in wireless networks and point-to-point connections.

Failover – The capability to switch over automatically to a redundant or standby computer server, system, or network upon the failure or abnormal termination of the previously active application, server, system, or network. Failover happens without human intervention and generally without warning, unlike switchover.

Forensics – A branch of forensic science pertaining to legal evidence found in computers and digital storage media.

Hard Expiration – A hard expiration is the time threshold for which a device has to update key material or it is not compliant with policy.

Hash – A deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string—the cryptographic hash value—such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message", and the hash value is sometimes called the digest or message digest.

Incident Response – The response or actions taken against an occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits, or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies.

Integrity – Guarding against improper information modification or destruction, including ensuring information non-repudiation and authenticity.

Internet Key Exchange (IKE) – The protocol used to set up a security association (SA) in the IPsec protocol suite. IKE uses a Diffie-Hellman key exchange to set up a shared session secret, from which cryptographic keys are derived.

Kerberos – A computer network authentication protocol which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner.

Key Management – The activities involving the handling of cryptographic keys and other related security parameters (e.g., Initialization Vector and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and zeroization.

Legacy Equipment – Equipment that is using old hardware or protocols that limit its capabilities. For example, an RTU that is communicating Conitel over 1200 BAUD modems is considered a legacy device.

Metadata – Metadata is data that describes or give structure to other data. In the context of this document metadata represents a data that describes or provides state for cryptographic material.

Nonce – In security engineering, a nonce is an abbreviation of a number used once (it is similar in spirit to a nonce word). It is often a random or pseudo-random number issued in an authentication protocol to ensure that old communications cannot be reused in replay attacks.

Policy – A policy is a set of rules and guidelines to which cryptographic material must conform.

Private Key – A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and is not made public.

Provisioning – Provisioning is the process of preparing equipment for installation into a system.

Public Key – A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.

Random Number Generator – A random number generator (often abbreviated as RNG) is a computational or physical device designed to generate a sequence of numbers or symbols that lack any pattern.

Real-Time – Real-time is designated as anything within a few minutes. Real-time does not necessarily mean the exact instance an event occurs. Real-time denotes that the system will notify the user in a reasonably short time period.

Redundancy – In the context of this document redundancy refers to having multiple instantiations of services for provide high availability.

Registration – Registration is the process of entering credentials for a device or entity into the system so that they may be managed by the system.

Remote station – Remote stations are physically dispersed locations that have equipment that the control station monitors and controls. An example is a substation for an electrical utility.

Role – A predefined set of rules establishing the allowed interactions between a user and resources. A role is a symbolic category of users that share the same security privilege.

Role-Based Access Control (RBAC) – A type of authorization that uses roles to determine access rights and privileges. A role is a symbolic category of users that share the same security privilege.

Security Association – A security association is an agreement of security parameters and the creation of a secure communication channel to perform the subsequent phases.

Separation of Duties – Separation of duty, as a security principle, has as its primary objective the prevention of fraud and errors. This objective is achieved by disseminating the tasks and associated privileges for a specific business process among multiple users. This principle is demonstrated in the traditional example of requiring two signatures on a check.

Soft Expiration – A time threshold for when a device should start to perform a key update. A soft expiration represents a threshold that allows a device to update key material in a timely manner that minimizes impact upon operational functionality.

Symmetric Cryptography – A cryptographic technique that uses the same secret key for both the encryption and the decryption transformations.

Ticket – A number generated by a network server for a client, which can be delivered to itself or a different server as a means of authentication or proof of authorization, and cannot easily be forged.

Trigger – Procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for keeping the integrity of the information on the database.

Trust Evidence – Evidence or data that must be provided during a trust negotiation that will assist the communicating parties to determine if they trust the other party enough to allow access to a resource.

Trust Negotiation – An approach to gradually establishing trust between strangers online through the iterative exchange of digital credentials. In contrast to a closed system, where the interacting entities have a pre-existing relationship (often proved by typing a username and password), trust negotiation is an open system, and strangers can build trust in one another. This is done by disclosing digital credentials.