

IEEE Trial Use Standard for SCADA Serial Link Cryptographic Modules and Protocol

2008

IEEE Trial Use Standard for SCADA Serial Link Cryptographic Modules and Protocol

Prepared by Working Group C6 of the
Substations Committee

Copyright © 2008 by the Institute of Electrical and Electronics Engineers, Inc.
Three Park Avenue
New York, New York 10016-5997, USA
All rights reserved.

This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to change. **USE AT YOUR OWN RISK!** Because this is an unapproved draft, this document must not be utilized for any conformance/compliance purposes. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only. Prior to submitting this document to another standards development organization for standardization activities, permission must first be obtained from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department. Other entities seeking permission to reproduce this document, in whole or in part, must obtain permission from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department.

IEEE Standards Activities Department
Standards Licensing and Contracts
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

Introduction

(This introduction is not part of IEEE Std 1711, IEEE Trial Use Standard for SCADA Serial Link Cryptographic Modules and Protocol.)

This standard applies to the retrofit of systems used to communicate between the control center SCADA master and intelligent electronic devices (IEDs) for substation integrated protection, control and data acquisition. The requirements of this standard are in addition to those contained in standards for individual devices (e.g., relays, switchgear). This standard does not apply to cryptographic modules used for the protection of dial-up access to IED maintenance ports.

This standard applies to a rapidly changing technology. It is anticipated that frequent revision may be desirable.

Participants

At the time this draft standard was completed, the WG C6 had the following membership:

David Whitehead, Chair
Dennis K. Holstein, Vice-chair

Markus Braendle	William Rush
Mike Dood	Samuel Sciacca
Chris Huntley	Yoshi Serizawa
John Kinast	John T. Tengdin
Thomas E. Kropp	Andrew Wright

Contents

1	Overview.....	1
1.1	Scope	1
1.2	Purpose	1
1.3	Background and objective.....	1
2	References	2
3	Definitions, acronyms and abbreviations	2
4	SCADA cryptographic module design requirements.....	3
4.1	Operating environment	3
4.2	SCM alarms and data reporting requirements.....	3
4.3	SCM secure enclosure and storage requirements.....	4
4.4	Communication component requirements for SCM operations	4
4.5	SCM documentation requirements	4
4.6	Quality requirements	5
4.6.1	SCADA interoperability	5
4.6.2	Scalability.....	5
4.6.3	Reliability	5
4.6.4	Availability.....	5
4.6.5	Maintainability	5
4.6.6	Flexibility and expandability	5
5	Cyber security requirements for SCADA asynchronous serial communications.....	5
5.1	SCM functional capability required.....	5
5.1.1	Identification and authentication requirements.....	5
5.1.2	CM software and firmware requirements.....	5
5.1.3	SCM management port requirements	6
5.2	Required capability with encryption enabled.....	6
5.2.1	Session establishment requirements	6
5.2.2	Operating mode requirements	6
5.2.3	Access port requirements	6
6	Test and evaluation requirements	7
6.1	Known answer test and evaluation requirements	7
6.2	Regression test requirements	7
6.3	Formal certification.....	8
7	Installation and commissioning requirements	8
7.1	Management requirements	8
7.1.1	Identity management requirements for SCM access and use control	8
7.1.2	Key management requirements.....	8
8	Performance (latency) constraints.....	8
9	SCADA cryptographic protocol specification.....	9
9.1	Octet order in messages.....	9
9.2	SSPP organizational layers	9

9.2.1	Session layer message types	10
9.2.2	Session negotiation	12
9.2.3	Session clocks.....	14
9.2.4	Session message formats	15
9.2.5	Session state machine	20
9.2.6	Broadcast messages	22
9.2.7	Forensics and intrusion detection	22
9.3	Transport layer.....	23
9.3.1	Transport layer header.....	24
9.3.2	Transport layer payload	24
9.3.3	Transport layer trailer	24
9.3.4	Transport layer sequence numbers	24
9.3.5	Transport layer error handling	25
9.4	Link layer	25
9.4.1	8-bit link layer message format	25
9.4.2	8-bit link layer error conditions during receive.....	27
9.4.3	8-bit link layer sender state machine	27
9.4.4	8-bit link layer receiver state machine	28
9.4.5	7-bit link layer.....	29
9.5	Cipher suites.....	30
9.5.1	IEEE 1711 defined cipher suites	30
9.6	Key management	36
9.6.1	Initial key loading.....	36
9.6.2	In-band transfer of keys	36
9.6.3	Revocation of keys	36
9.7	SSPP provisioning.....	36
9.8	SSPP management messages.....	36
Annex A	Bibliography	37
Annex B	Requirements traceability to cryptographic protocol specification	38
B.1	Module design (Clause 4)	38
B.1.1	Operating environment	38
B.1.2	SCM alarms and data reporting.....	38
B.1.3	SCM secure enclosure and storage.....	38
B.1.4	Communication components	38
B.1.5	SCM documentation	38
B.1.6	Quality requirements.....	38
B.2	Cyber security (Clause 5)	38
B.3	Test and evaluation (Clause 6)	38
B.4	Installation and commissioning (Clause 7)	39
B.5	Performance constraints (Clause 8)	39
Annex C	SCM implementation requirements and issues	40
C.1	Link layer mixed mode operation	40

C.2	SCM hardware handshaking	40
C.3	Dialup modem interaction	43
C.4	Timing consideration for shared connections.....	43
C.5	Design consideration for cipher suites.....	44
C.6	Side channel design considerations	44
C.7	Session clock considerations	44
C.8	Multi-thread, multi-process and other design considerations.....	45

Table of Figures

Figure 1 Notional architecture for retrofit..... 1

Table of Tables

Table 1 Session layer message types 10
Table 2 Session state machine for static sessions 20
Table 3 Session state machine for one dynamic session 21
Table 4 Eight-bit Link layer sender state machine..... 27
Table 5 Eight-bit link layer receiver state machine 28
Table 6 Seven-bit link layer ASCII requirements..... 29
Table 7 Seven-bit link layer state machine 29
Table 8 Nine common signals on the RS-232 connector 40

1 Overview

1.1 Scope

This trial use standard defines the cyber security requirements and enabling cryptographic protocol to protect existing asynchronous serial communications between control center master stations and electric power substation remote terminal units (RTU) used for supervisory control and data acquisition (SCADA). The requirements and cryptographic protocol contained herein are tailored to retrofit existing asynchronous serial communications without requiring changes to the SCADA master or RTU hardware and software. IEEE 1711 compliant SCADA cryptographic modules are designated "SCM."

1.2 Purpose

Utilities will use the cyber security requirements and cryptographic protocol as part of a procurement specification to improve cyber security over existing asynchronous serial communication links used for SCADA operations. Figure 1 shows a notional architecture for retrofitting existing SCADA communication links.

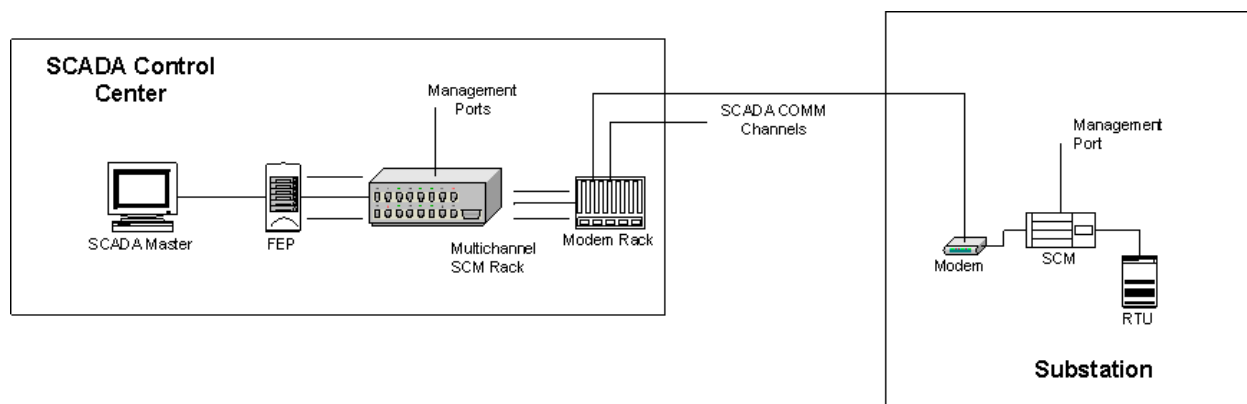


Figure 1 Notional architecture for retrofit

Security solution providers will use the cryptographic protocols to build SCADA cryptographic modules (SCM) that can be inserted between the SCADA master station or front end processors (FEP) and modem (or modem bank) on the control center side, and between the modem and RTU in the substation. If a port switch is used, the SCM is inserted between the modem and port switch in the substation.

Interoperability between IEEE 1711 compliant cryptographic modules designed by different providers is a goal, but not a requirement, of this standard.

1.3 Background and objective

The American Gas Association (AGA) planned to publish a series of reports describing the enabling technologies to improve cyber security for operations. Part 1 of the series (AGA 12-1), which has been published, addressed the general requirements (AGA, March 14, 2006).

Part 2 of the series (AGA 12-2) defined the cryptographic protocol for a retrofit solution. Part 2 reached a reasonable level of maturity and was implemented by several vendors. These early prototype cryptographic modules were submitted for limited field testing in gas, electric and water operational environments, and tested for interoperability by the Gas Technology Institute (GTI) in their laboratories.

The cryptographic protocol has also been analytically reviewed by Sandia National Laboratories (SNL). SNL suggested improvements in the protocol are incorporated in this standard.

The AGA 12 project was terminated before part 2 was finalized. The draft report is not copyrighted and there is no objection from AGA or GTI for its use by IEEE. Lessons learned from interoperability testing, field testing and the protocol analysis by SNL applicable to electric power SCADA communications were considered for development of IEEE 1711.

IEEE 1711 is a trial use standard. The objective is to update the standard in two years or less. The trial use period should provide the opportunity to gain more field experience with the cryptographic modules and protocol. The stakeholders are the engineers at electric utilities and consultants/system integrators who are seeking solutions for the existing unsecured serial links, and manufacturers who design products addressing these cyber security gaps.

2 References

- [1] ANSI. (1998). ANSI X9.69: Framework for Key Management Extensions, American National Standards for Financial Services. ANSI.
- [2] IEEE 1613-2008: Standard Environmental and Testing Requirements for Communications Networking Devices in Electric Power Substations.
- [3] IEEE 1686-2007: IEEE Standard for Substation Intelligent Electronic Devices (IEDs) Cyber Security Capabilities.
- [4] FIPS PUB 140-2: Security Requirements for Cryptographic Modules, May 25, 2001.

3 Definitions, acronyms and abbreviations

Definitions not listed are defined in The Authoritative Dictionary of IEEE Standards Terms (IEEE, Seventh Edition - 2000).

3.1

cryptographic module

The set of hardware, software and firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.

3.2

form c contact

A single pole double throw (SPDT) contact in which the common connection is to the mid-point between the normally open (NO) and the normally closed (NC) contacts.

AES	Advanced Encryption System
AGA	American Gas Association
CAPI	Cryptographic Application Programming Interface
CAR	CM Address Response
CBC	Cipher Block Chaining
CM	Cryptographic Module
CMVP	Cryptographic Module Validation Program
CTS	Clear to Send
DTE	Data Terminal Equipment
DUT	Device Under Test
ECDSA	Elliptic Curve Digital Signature Algorithm
ECB	Electronic Code Book
FEP	Front End Processor

FIPS	Federal Information Processing Standard
GTI	Gas Technology Institute
IED	Intelligent Electronic Device
KAT	Known Answer Test
MAS	Multiple Address System
MCT	Monte Carlo Tests
NC	Normally Closed
NIST	National Institute of Standards and Technology
NO	Normally Open
PKCS	Public Key Cryptographic Standard
PT	Pre-Transmission (as used in communication channels)
RBAC	Role Based Access Control
RCA	Request CM Address
RSA	Rivest, Shamir and Adleman (The algorithm was publicly described in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT)
RTU	Remote Terminal Unit
RTS	Request to Send
SCADA	Supervisory Control And Data Acquisition
SCM	SCADA Cryptographic Module
SHA	Secure Hash Algorithm
SNL	Sandia National Laboratories
SPDT	Single Pole Double Throw
SSPP	Serial SCADA Protection Protocol

4 SCADA cryptographic module design requirements

4.1 Operating environment

1. SCMs installed in field device sites shall be designed for an indoor substation environment and comply with IEEE Std™ 1613[2].
2. SCMs installed in SCADA master sites shall meet the temperature requirements of the specific environment.
3. SCM power supply shall have minimal power drain, with optional input voltage ratings of 120/240 Vac, or 48 to 125 Vdc.
4. The host and RTU or Intelligent Electronic Device (IED) ports shall have serial RS232 physical interfaces, and the management ports shall have serial RS232 and Ethernet TCP/IP interfaces.

4.2 SCM alarms and data reporting requirements

1. SCMs shall provide the capability to collect, record and make available usage and forensic data (intrusions or tampering detected, loss of power, etc.) to an external management system.
 - a. SCM alarms detected within the control center shall be recorded by the SCADA master and reported as alarms to the SCADA Operator.
 - b. SCM alarms detected in substations shall provide a Form C contact intrusion detection alarm output, which may be connected to an alarm point on the user's local RTU, or substation host, to alert the user's system operator that intrusion has been attempted, that tampering is detected, that the SCM has failed to function, or that the power supply input is lost.

- c. The SCM alarm contacts shall be Form C with alarm conditions signified by the normally closed (NC) contact in the closed position. Thus under normal conditions, the NC contact is open, and closes on loss of power or any other alarm condition.
2. SCMs shall provide the capability to monitor and terminate sessions when no activity occurs for a configurable period of time and/or event sequence.

4.3 SCM secure enclosure and storage requirements

1. SCMs shall provide the capability to generate, exchange, store, use and destroy credentials and cryptographic keying materials within its FIPS 140-2 Level 2 or higher compliant cryptographic boundary.
2. SCMs shall provide the capability to exhibit tamper evidence.

4.4 Communication component requirements for SCM operations

SCM shall provide the capability to adjust timing parameters (time-out, channel turn-on, turn-off, turn-around, and squelch times) in the sender, receiver, or both to suit the specific requirements of the communication component (modem, port switch or radio).

Note 1. Some field devices respond to requests faster than others. Typical RTUs are ready to begin a response within a few milliseconds. However, communication channel equipment may introduce additional delay. For example, it is common practice to key up a radio transmitter or wire line, wait for the receiver to open, and wait for the path to settle down before the response begins. This sometimes is referred to as the PT mark. The receiver, to synchronize to the serial channel, also uses the PT mark. PT marks often are set at 8 ms, but can be as long as 50 ms, perhaps longer when an Multiple Address System (MAS) repeater is used - for example, on a 900 MHz radio channel.

Note 2. At the end of the message, there often is a need to hold the channel at mark for a short period of time so the receiver can decide the message has ended. This is sometimes referred to as the "post mark." Post marks typically have delays based on the time to send two bytes of data, some even longer. Radios (MAS¹ and spread spectrum) need long pre-transmission (PT) marks and post marks. They also need time for the slave radio to go from transmit to receive and back again.

Note 3. RTUs and other IEDs may also include WAN cards for communications. If the PSTN is shared, each remote site needs an auto-answer modem and port switch in order to communicate with the RTU or IED. A port switch is only needed if the port is shared.

4.5 SCM documentation requirements

1. The manufacturer shall state the SCADA protocols and communication line speeds supported.
2. The manufacturer shall state the cryptographic operating modes, cryptographic functions and cipher suites supported.
3. The manufacturer shall state the status of cryptographic certification; i.e., not certified, certification in process, or certified. If certification is in progress or if the SCM has been certified, the manufacturer shall include the name of the certification authority, date submitted or certified, and the contact information of the certification authority.

¹ These are usually 900 MHz radio, which are very common on electric power distribution feeder applications. Some are being replaced with spread spectrum radios that do not require FCC licensing.

4.6 Quality requirements

Quality requirements are defined for SCADA interoperability, scalability, reliability, availability, maintainability, and flexibility and expandability. IEEE 1711 recommends that the phrase “shall not significantly degrade” used in the following subsections be quantified by the end user.

4.6.1 SCADA interoperability

The cryptographic system or its components shall not degrade the capability of IEDs to interoperate over networks on which they were designed to interoperate. Interoperability requires that cryptographic modules transcend products and networks.

4.6.2 Scalability

The cryptographic system or its components shall not significantly degrade the scalability of networks designed to operate without cryptographic protection. Scalability defines how capacity and load affect performance.

4.6.3 Reliability

The cryptographic system or its components shall not significantly degrade the reliability of networks designed to operate without cryptographic protection.

4.6.4 Availability

The cryptographic system or its components shall not significantly degrade the availability of networks designed to operate without cryptographic protection.

4.6.5 Maintainability

The cryptographic system or its components shall not significantly degrade the maintainability of networks designed to operate without cryptographic protection.

4.6.6 Flexibility and expandability

The cryptographic system or its components shall not significantly degrade the flexibility and expandability of networks designed to operate without cryptographic protection.

5 Cyber security requirements for SCADA asynchronous serial communications

5.1 SCM functional capability required

5.1.1 Identification and authentication requirements

1. SCMs shall provide the capability to be identified and authenticated without human action.
2. SCMs shall provide the capability to uniquely identify and authenticate operators accessing the management ports of SCM components using role based access control (RBAC) mechanisms defined in ANSI X9.69 (ANSI, 1998).
3. SCMs shall provide the capability to uniquely identify and authenticate a SCM that is requesting services, such as session establishment.

5.1.2 CM software and firmware requirements

Cryptographic hardware requires the use of specific drivers and software, referred to as middleware, to interface the hardware to standard applications and Internet protocols, including email, browser, encryption/decryption, and digital signing clients.

1. Industry reviewed and standards based cryptographic middleware shall be used; e.g., the industry standard Public Key Cryptographic Standard (PKCS) or Microsoft's Cryptographic Application Programming Interface (CAPI).

2. All cryptographic algorithms shall be approved by NIST.
3. Encryption shall use the Advanced Encryption System (AES) with a minimum key length of 128 bits.
4. Digital signing shall use the RSA² with a minimum key length of 1024 bits, and Elliptic Curve Digital Signature Algorithm (ECDSA) with a minimum key length of 160 bits.
5. Hashing shall use a Secure Hash Algorithm (SHA), specifically SHA-1 as defined in FIPS 140-2 shall be used.

5.1.3 SCM management port requirements

1. SCMs shall provide a management communication port for operator authentication, and SCM configuration and management.
2. SCMs shall provide the capability for secure provisioning (loading of keying materials) and management of SCMs both in-band (within the SCADA communication channel for the SCMs) and out-of-band (either remote or local update through a physically separate communication port or channel).

5.2 Required capability with encryption enabled

Encryption of SCADA messages is an optional requirement for cryptographic modules. If SCMs provide the capability to encrypt and decrypt messages, the following requirements apply.

5.2.1 Session establishment requirements

1. SCMs shall provide the capability to authorize session establishment and reestablishment.

5.2.2 Operating mode requirements

1. SCMs shall provide the capability to operate in a mixed-mode³ communication topology.
2. SCMs shall provide the capability to operate in a multipoint, multidrop, point-to-point, and cascaded topology.
3. SCMs shall provide the capability to operate in broadcast or multicast communication modes.
4. SCMs shall provide the capability to pass-through, in plaintext, communication system parameters (modem commands).

5.2.3 Access port requirements

1. SCMs shall provide the capability for the user to configure bypass mode operation via the management port. If bypass operation is enabled, the following constraints shall be in effect.
 - a. Once so configured and set in the bypass mode, the change from bypass to encryption mode shall be done either by actions via the management port or by in-band commands.
 - b. Once so configured and set in the encryption mode, the change to bypass mode shall only be possible via the management port.

² The algorithm was publicly described in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT.

³ Mixed-mode is a topology in which some IEDs on a single communication channel are protected by SCMs and others are not; for example, to facilitate sequential deployment of SCMs over time.

- c. Each encrypting SCM shall be configurable with multiple addresses for simultaneous encryption and bypass mode operation (Necessary when operating as a sub master).
2. SCMs shall provide the capability to communicate through the plaintext port to SCADA devices, including control center equipment (a SCADA Master or a Front End Processor) and substation equipment (a Remote Terminal Unit or local SCADA Master).
3. SCMs shall provide the capability to communicate through the ciphertext port to SCADA communications equipment, including modems, routers, and radios.
4. SCMs shall provide the capability to communicate through the management port for operator authentication, and provide SCM configuration and management tools.

6 Test and evaluation requirements

6.1 Known answer test and evaluation requirements

1. The SCM manufacturer shall perform Known Answer Tests (KATs) and Monte Carlo Tests (MCTs). KATs are designed for Electronic Code Book (ECB) mode implementation. MCTs are designed for ECB and Cipher Block Chaining (CBC) mode implementations⁴.
 - a. Encryption tests shall include the serial input of plaintext, serial output of ciphertext, and validation and independent verification of the encryption using an accepted source such as a trusted third-party program or National Institute of Standards and Technology (NIST) test vectors.
 - b. Decryption tests shall include the serial input of ciphertext, serial output of plaintext, and validation and independent verification of the decryption using an accepted source such as a trusted third-party program or NIST test vectors.
2. Supplying known plaintext to the plaintext interface of the Device Under Test (DUT) and comparing the output ciphertext to known ciphertext shall be used to evaluate the encryption function.
3. Supplying known ciphertext to the ciphertext interface of the DUT and comparing the output plaintext to known plaintext shall be used to evaluate the decryption function.

6.2 Regression test requirements

1. Regression testing is not one test, but a series of tests that measure critical aspects of the SCM under test.
 - a. For each new release of SCM software and hardware, regression testing shall be performed to ensure that the upgrade will function properly prior to deployment.
 - b. A regression test plan shall be published by the tester to identify which new basic test objectives should be run against each new SCM product release.
2. SCM regression testing shall be performed to verify that a hardware or software upgrade does not impact performance, reliability, or functionality of the cryptographic system.
3. Regression testing does not measure new features or capabilities.
 - a. Test data from past regression test shall be used as a baseline⁵ for the current regression test.

⁴ NIST 800-17 specifies the modes of validation (NIST, February 1998). Test values are described in the file rijndael-vals.zip, which is available from <http://csrc.nist.gov/encryption/aes/rijndael/rijndael-vals.zip>.

- b. If no current data exists, tests shall first be run against the current cryptographic system before testing the upgrade.

6.3 Formal certification

The manufacturer shall obtain SCM certification from a member of the Cryptographic Module Validation Program (CMVP). Under this program, NIST accredits internationally recognized laboratories that are qualified to certify that components of the cryptographic system conform to Federal Information Processing Standards. This assurance includes both the specifications and the adequacy with which the specification is implemented.

7 Installation and commissioning requirements

Installation, pre-operating checkout, operational turn-over, and maintenance of SCMs shall be defined by the end-user commensurate with company security policies.

7.1 Management requirements

The end-user is responsible for establishing and maintaining an acceptable level of security (determined as an outcome of risk assessment of the company) throughout the life of the cryptographic system. IEEE 1711 SCM management requirements are designed to minimize the burden imposed by key management on SCADA operations, and to minimize the inconvenience and complexity imposed on the user.

7.1.1 Identity management requirements for SCM access and use control

A secure identity management system, based on industry standards, shall be used to ensure secure access to and use of SCM network ports and functions.

7.1.2 Key management requirements

A secure key management system, based on industry standards, shall be used to manage the cryptographic keys for SCM operation.

8 Performance (latency) constraints

1. Any timing reported by the SCM firmware or software shall be independently checked for reasonableness. There are four events of interest that occur:
 - T_0 is the time at which the first bit of a message enters the encrypting SCM
 - T_1 is the time at which the last bit of a message enters the encrypting SCM
 - T_2 is the time at which the first bit of a message exits the decrypting SCM
 - T_3 is the time at which the last bit of a message exits the decrypting SCM
 - T_r is the theoretical time to transmit the message at the test data rate with no interruption
2. The overall CM latency introduced by a pair of cryptographic modules shall be defined as
$$\text{SCM Latency} = T_3 - T_0 - T_r$$
3. Jitter shall be defined as the standard deviation of at least 100 samples of a particular latency test.

⁵ Without a baseline against which to compare the SCM upgrade, it cannot be determined that the cryptographic system has been improved or regressed.

- a. Since measured latency and jitter, as defined in IEEE 1711, depend on the use of flow control applied to the messages entering the encrypting SCM, the calculated latency shall be reported with and without flow control enabled.
- b. If both hardware and software flow control options are present, the latency and jitter measurements for each configuration shall be reported.
- c. Flow control shall not be actively applied to the output of the decrypting module when measuring latency and jitter.
- d. The baseline measurement of latency and jitter shall use a hardwire connection between the ciphertext ports of the encrypting and decrypting SCMs.

9 SCADA cryptographic protocol specification

Serial SCADA Protocol Protection (SSPP) is the normative cryptographic-base communication protocol specification for IEEE 1711.

9.1 Octet order in messages

1. SSPP shall use network order for integers represented by multiple octets for consistency, interoperability, and availability.
2. Specifically, all integers that require more than one octet shall be in network octet order: the most significant octet comes first, then the less significant octets in descending order of significance (MSB LSB for two-octet integers, B3 B2 B1 B0 for four-octet integers). The highest bit (value 128) of an octet is numbered bit 7; the lowest bit (value 1) is numbered bit 0.
3. Values shall be unsigned unless otherwise noted.
4. Values explicitly noted as signed shall be represented in two's complement notation.

9.2 SSPP organizational layers

1. SSPP may be organized into 3 layers called the Session Layer, the Transport Layer, and the Link Layer.
 - a. If organized into 3 layers, the session layer shall provide the capability to distinguish different kinds of messages, and shall be responsible for session key negotiation and abstract data exchange.
 - b. If organized into 3 layers, the transport layer shall be responsible for encryption and integrity checking of SCM messages and shall guard against replay.
 - c. If organized into 3 layers, the link layer shall be responsible for formatting messages onto and from the channel in such a way as to delimit certain parts of messages and to ensure that unencrypted SCADA traffic is distinguished from encrypted traffic (when operating in mixed mode).
 - d. An implementation of the protocol not organized in this way, shall provide a functionally equivalent capability. In other words, an observer of only the external communications can conclude that the SCM is operating in this manner, independent of the actual implementation details.
2. Each layer shall be responsible for different aspects of the formatting of a message.
 - a. The session layer shall handle a SCADA message as a sequence of octets. To the transport layer, this sequence of octets is the payload.
 - b. The transport layer shall encrypt the payload, and shall associate the payload with a header and trailer.

- c. The link layer shall format the transport layer's header, payload, and trailer onto the physical communication layer by escaping the transport header, payload, and trailer, and shall add delimiters around the transport layer header, payload, and trailer, shown as:

```

Session:  msg
Transport: Header E[ msg ]           Trailer
Link:     ESC SOM X[Header E[ msg ]] ESC SOT X[Trailer] ESC EOM
    
```

Where E [...] represents encryption, and X [...] represents escaping (doubling of any occurrence of ESC; see Link Layer below). ESC, SOM, SOT, and EOM are link layer delimiters indicating Escape, Start Of Message, Start Of Trailer, and End Of Message. Note that they do not correspond to the ASCII characters of the same names, though they may be assigned the same numeric values.

9.2.1 Session layer message types

- The session layer shall format messages and the procedures to transfer data on static or dynamic sessions, and the capability to setup and teardown dynamic sessions. Session layer message types shall be designated as shown in Table 1.

Table 1 Session layer message types

Message description	Session message label	Session message type
Open message	OPN	1
Acknowledgement message	ACK	2
Data messages carry data. Depending on the session type, this may contain SCADA messages or SCM management data.	DTA	3
Clear messages are sent to terminate a session. The session to be closed is the session used to transport the CLS message.	CLS	4
Error message is issued when a SCM receives a message with an invalid sessionId.	ERR	5
Begin message	BEG	6
Request for address message is sent to get the address of an answering cryptographic module.	RCA	7
Response to a request for address message is sent by the responding cryptographic module.	CAR	8

- OPN, ACK, and BEG messages shall be exchanged to negotiate session keys for a dynamic session. In general, a session permits data transfer between two or more (i.e., broadcast sessions) SCMs. A broadcast session shall allow one-way, and one-to-many transfers.
- A SCM shall be configured with one or more static sessions for each peer SCM with which it will communicate.
 - If only one static session is defined, it shall be a session of ESTABLISHMENT type.
 - The configuration shall specify a cipher suite to use to encrypt and decrypt messages sent on that session.
 - The configuration shall also specify the session type, which may be one of the following.

ESTABLISHMENT (0): This session type shall only be used to establish dynamic sessions between SCMs. The OPN, ACK, and BEG messages shall be sent within an

ESTABLISHMENT session. ESTABLISHMENT sessions shall only operate as static sessions, and shall be established only by configuration.

DATA (1): This session type shall be used to indicate that DTA messages sent within the session contain SCADA data.

MANAGEMENT (2): This session type shall be used to indicate that DTA messages sent within the session contain management data.

BROADCAST (3): This session type shall be used to indicate that DTA messages sent within the session contain SCADA broadcast data. Broadcast sessions shall be one-way traffic only – no acknowledgement is required.

MANAGEMENT_BROADCAST (4): This session type shall be used for in-band, vendor-defined broadcast management functions. These shall also be one-way traffic – no acknowledgement is required.

4. A static ESTABLISHMENT session shall negotiate a dynamic session.
 - a. The SCM shall issue OPN, ACK, and BEG within an ESTABLISHMENT session to set up a dynamic session in response to receipt of SCADA data on its cleartext port that is destined to a remote SCM for which the sending SCM has no open session.

Note: Sessions are normally kept open for a long time, to amortize the temporal cost of session setup over many messages.

- b. A SCM shall also provide the capability to establish a session for other reasons, such as in response to an operator command received on its management port. Note: The operator commands are not defined, they are a local matter.

Caution: A static session can be used to manage data. Depending on the cipher suite, such messages may enjoy integrity protection and/or secrecy. However, static sessions provide no protection against replay or reordering. Use of static sessions for data transfer shall be limited to situations where replay or reordering is not a risk, such as with certificate exchange. Given that static session keys cannot be changed remotely (as of this writing), careful operations require minimizing use of static session messages.

5. Dynamic sessions shall only be used for data exchange. A dynamic session shall not be used to establish another dynamic session.
6. ESTABLISHMENT, DATA and MANAGEMENT sessions are bidirectional. Either SCM shall have the capability to send messages using the sessionId. BROADCAST and MANAGEMENT_BROADCAST sessions are unidirectional. Only the publisher of a broadcast message shall send data on the broadcast session.
7. A SCM receiving an OPN shall respond with an ACK, and an ACK with a BEG, subject to its configuration identifying SCMs with which it may communicate.
 - a. The exchange of OPN, ACK, and BEG shall establish a sessionId, and encryption and authentication parameters such as a shared AES session encryption key, a shared session HmacSHA key. These parameters shall be used in subsequent DTA and CLS messages.
 - b. Each SCADA message received on the cleartext port shall be encrypted and MAC'ed and transmitted in a DTA message.
 - c. Sequence numbers shall be used to prevent replay or reordering in dynamic sessions.
 - d. Either SCM shall have the capability to send DTA messages on this dynamic session containing SCADA commands and responses; that is, a DATA session is bidirectional.

- e. The sessionId shall be used to distinguish different types of traffic destined for a SCM, so that in-band management functions may be performed in parallel with SCADA data transfer.
Caution: SSPP does not ensure that data is delivered; e.g., through retrying transmission.
 - f. There is no acknowledgement of reception and there is no retransmission of messages by a cryptographic module. A dropped DTA message is treated as if it were not received; e.g., due to line noise, so that a new DTA message is sent when the SCADA system retries (or sends a new SCADA message). If a session establishment message (OPN, ACK, or BEG) is dropped, then the session will not be established. A new attempt to establish a session will be attempted when the SCADA application sends another SCADA message.
 - g. This approach was taken because SCADA applications must already deal with messages that are lost or damaged due to line noise. Some may retry, some may move to the next device in the polling cycle, some may send NACKs, etc. Retrying a cryptographic module message might result in delivery of a belated SCADA response when the master is not expecting it. Retrying cryptographic module messages might also cause congestion on the line. Leaving retry up to the SCADA application should be less likely to result in undesirable behavior.
8. SCADA data is transferred over dynamically established sessions that are generally bidirectional. For such dynamic sessions, a sequence number shall be used to prevent replay, and is used as an input to the cipher suite in encrypting or decrypting the message. In this way, one or more messages can be dropped without preventing decryption of later messages. If an appropriate clock resolution is determined during session negotiation, the sequence number may represent the transmission time of the message relative to the start of the session, and the receiver may additionally check that the message is received at approximately the correct time. This prevents an adversary from delaying messages in transit.
9. For static sessions that are used for key exchange, however, a sequence number is undesirable, as it is preferable to minimize the amount of shared state that must be available between two cryptographic modules. Otherwise synchronization problems might arise that could prevent establishment of a session. Thus, the OPN, ACK and BEG messages that are exchanged over static sessions shall carry copies of the earlier message's sequence numbers to prevent replay.

9.2.2 Session negotiation

- 1. An initiating SCM shall send an OPN message within an ESTABLISHMENT session to establish a dynamic session with a responding SCM.
 - a. The session type to be established shall be dependant on the type of information carried by the session.
 - b. The responding SCM shall respond with an ACK within the ESTABLISHMENT session, if the initiating SCM is identified by the configuration of the responding SCM as a unit with which it can communicate, and providing the OPN message is authentic.
 - c. The initiating SCM shall respond with a BEG message.
 - i. The initiating SCM shall start a timer on sending the OPN, and cancel it upon receipt of the ACK.
 - ii. If the timer expires, the initiating SCM shall discard the session.

- iii. The responding SCM shall start a timer on sending the ACK, and cancel it upon receipt of the BEG.
- iv. If the timer expires, the responding SCM shall discard the session.
- d. On receipt of the BEG, the responding SCM shall consider the session open.
 - i. The initiating SCM shall consider the session open upon transmission of the BEG.
 - ii. While waiting for an ACK or BEG, a SCM shall discard or buffer any messages received from attached SCADA units.
2. SSPP shall provide the capability to block the establishment of a dynamic session for the following conditions, and the initiating SCM shall attempt to negotiate a new session when required.
 - a. If the OPN message is damaged and does not reach the responding SCM, the initiating SCM's timer shall expire and no resources shall be reserved for the session.
 - b. If the ACK message is damaged and does not reach the initiating SCM, both timers shall expire, and no resources shall be reserved for the session.
 - c. If the BEG message is damaged, the responding SCM's timer shall expire, and no resources shall be reserved for the session.
 - i. The initiating SCM shall consider it open, but if the responding SCM sends a new OPN for the same sessionId, the new session shall replace the current one.
 - ii. If the initiating SCM sends data on the partly open session, the responding SCM shall respond with an ERR message, and the responding SCM shall then close the partially open session.

9.2.2.1 Sequence number requirements and use

1. The sequence numbers for static sessions, which transport OPN, ACK, and BEG messages, shall be 112 bits long.
 - a. They shall be generated randomly or via a persistent incremental counter.
 - b. If a secure pseudorandom generator is used, it shall be seeded with some information unique to the SCM, such as its address, and with at least 112 bits of true random information.
 - c. The seed should be persistent over power loss. If a persistent incremental counter is used, its value shall be persistent for the lifetime of the device.
2. The sequence numbers of the OPN and ACK messages shall be used to guard against replay of a session negotiation.
 - a. They shall also used by the various cipher suites to ensure uniqueness of encrypted blocks.
 - b. The ACK message shall include a copy of the sequence number of the matching OPN message.
 - c. The BEG message shall include copies of both the matching OPN and ACK messages sequence numbers.

9.2.2.2 OPN, ACK and BEG message requirements and use

1. An OPN message may contain zero or more session requests.
2. An OPN shall not contain any broadcast session requests.

3. The ACK message shall include session requests that correspond to those of the OPN message. These session requests shall fill in unspecified values of the session requests from the OPN, as specified in Clause 9.2.2.3.
4. The ACK may include session requests for additional sessions. These shall include broadcast session requests for which the sender of the ACK is the publisher.
5. The BEG corresponding to an ACK must include session requests for all the sessions from the ACK, except for broadcast session requests which shall not be repeated. These session requests shall fill in unspecified values of the session requests from the ACK, as specified in Clause 9.2.2.3.
6. The BEG may include session requests for additional sessions. These shall include broadcast session requests for which the sender of the BEG is the publisher. Session requests in a BEG shall not include unspecified values.

9.2.2.3 Rules to select selected values

If the resolution field of a session request is zero, the resolution, tolerance, base, and expiry fields of that request are unspecified. The receiver of an OPN or ACK is free to choose values for those fields as appropriate and supply them in an ACK or BEG. The following rules apply to the selection of these values.

1. The resolution field of a session request shall be greater or equal to the resolution field of the earlier session request.
2. The tolerance field of the session request shall be zero if the tolerance field of the previous session request is zero, and the resolution field was non-zero.
3. Tolerance of zero indicates that the SCM either does not support or does not wish to use session clocks. Otherwise, the tolerance field of the session request shall be greater or equal to the tolerance field of the previous session request.
4. The sequence number length field shall be large enough to accommodate the maximum possible sequence number that could be generated before the session expires.
5. The base field of the session request shall be greater or equal to that of the earlier session request.
6. If the expiry field of the session request is zero, then the session shall never expire, and the expiry field of the following session request shall be zero.
7. Otherwise, the expiry field of the following session request shall be non-zero and chosen such that the expiry multiplied by the resolution is less or equal to the earlier session request's expiry multiplied by the resolution. That is, the actual time that the expiry field represents in a session request shall not be longer than that of the earlier session request.
8. The values of resolution, tolerance, base, and expiry contained in session requests in the BEG message shall be the agreed-on values for both SCMs to use.

9.2.3 Session clocks

1. Dynamic sessions have a session clock if a non-zero tolerance was present in the session request of the BEG message that opened the session. In this case, sequence numbers for the session shall represent time since the beginning of the session in ticks.
 - a. The duration of a tick is determined during session negotiation, and shall be short enough that no more than one message can be sent during a tick.

- b. The receiver of a message shall check that the sequence number of a received message is both newer than the last received sequence number and within a tolerance, measured in ticks, of the current session time.
- c. The receiver shall measure the current session time for this check upon receipt of the beginning of the message header, or as close to that as practical.
2. The tolerance provides a window, twice the width of the tolerance, during which receipt of the message is allowed.
 - a. This tolerance, determined during session negotiation, shall not be less than the ACK/BEG timeout used during session negotiation, and shall be wide enough to accommodate clock drift up to the expiry of the session.
 - b. Most sessions should begin at time 0; that is, a message sent immediately at the beginning of the session would have sequence number 0. However, the initial time for the session may start at any value. This is particularly useful for broadcast sessions where the master SCM may have created the broadcast session when first negotiating with SCM A, and later shall ensure when negotiating with SCM B that both SCM A and SCM B have the same view of time for that session.
3. A SCM sending a BEG message may determine the beginning of session time as the time when it initiates sending of the BEG message, completes sending of the BEG message, or any time between those two events. IEEE 1711 recommends using a time half way between those two events. A SCM receiving a BEG message may determine the beginning of session time as the time when it begins receiving the BEG message, completes receiving the BEG message, or any time between the two events. A reasonable choice is half way between those two events.
4. For sessions with a non-zero agreed-on value of expiry, both SCMs shall consider the session closed when (expiry - base) clock ticks have elapsed. Either or both SCMs may send CLS messages when the session expires, but sending is not required.
5. In selecting values for the resolution, tolerance, base, and expiry fields of a session request, whether issuing an OPN or responding to an ACK, a SCM should take into account both the ACK timeout value for session negotiation and potential clock drift.
 - a. As a general rule, tolerance shall satisfy the following relation:
$$tolerance \geq AckTimeout + expiry * 2 * ClockAccuracy$$
 - b. Where AckTimeout is the OPN or ACK timeout value and ClockAccuracy is the worst-case clock drift of the SCM's clock expressed as a fraction (e.g., parts-per-million). Twice the clock drift shall be used to allow for one clock drifting fast and the other drifting slow.
6. The negotiation rules, specified in this clause (9.2.3), shall be used to ensure that two SCMs with different accuracy clocks will agree to use the worst-case tolerance.

Caution: Care should be taken to ensure consistent units are used in evaluating this formula. As an example, with an AckTimeout of 1 second, expiry of 1 day, and clock accuracy of 50 ppm, tolerance should be at least 10 seconds. If too small a tolerance is used, the two SCM's clocks could drift enough that after some time the sequence numbers may consistently fall outside the valid window and all subsequent messages will be dropped until the session expires.

9.2.4 Session message formats

9.2.4.1 OPN message

1. An OPN message shall consist of 0 or more session requests:

numberSessions - 1 octet

where the numberSessions octet indicates the number of session requests to follow in the message.

2. A session request shall consist of:

sessionType	- 1 octet
sessionId	- 1 octet
resolution	- 4 octets
tolerance	- 4 octets
seqLength	- 1 octet
base	- 8 octets
expiry	- 8 octets
cipherSuite	- length dependant on the cipher suite

a. The sessionType field shall indicate the session type, which can be one of the following values:

DATA = 1
MANAGEMENT = 2
BROADCAST = 3
MANAGEMENT_BROADCAST = 4

- i. DATA shall be used for regular SCADA data transfer
 - ii. MANAGEMENT shall be used for in-band vendor-defined management functions
 - iii. BROADCAST shall be used for SCADA data broadcasts
 - iv. MANAGEMENT_BROADCAST shall be used for in-band, vendor-defined broadcast management functions
- b. The sessionId field shall be the dynamic session id to use. The sessionId field shall not be zero.
- c. Resolution shall be the number of microseconds per session clock tick.
- i. If resolution is zero in a session request in an OPN message, the resolution, tolerance, base, and expiry fields shall be considered unspecified, and the SCM receiving the message shall choose appropriate values.
 - ii. Resolution shall not be zero in a session request in a BEG message.
- d. Tolerance shall be the maximum number of ticks that a sequence number may differ from the current session time. If zero, the tolerance check for received sequence numbers shall not be performed for messages received on this session.
- e. SeqLength shall be the length of sequence numbers in the header of messages for the session, in octets. If non-zero, seqLength shall be between 2 and 14.
- f. Base shall be the starting value for the session clock in ticks.
- i. The session shall begin this many clock ticks before the session negotiation.
 - ii. Base shall be less than expiry.
- g. Expiry shall be the time from the beginning of the session (adjusted by base) until the session expires, in clock ticks.
- i. If expiry is zero, the session shall never time out.
 - ii. If expiry is zero, tolerance shall be zero.

- iii. If tolerance is not zero, expiry shall not be zero.
- h. The content of the cipherSuite field depends on the particular cipher suite.
 - i. The first octet of the cipherSuite shall always indicate the type of cipher suite.
 - ii. If zero, the SCM receiving the OPN message shall choose the cipher suite.
 - iii. The cipherSuite field shall not be zero in a session request in an ACK message.
- 3. OPN, ACK, and BEG messages may contain more than one session request to permit multiple sessions to be established in one negotiation. For example, a DATA session, a BROADCAST session, and a MANAGEMENT session could be established simultaneously.
- 4. OPN, ACK, and BEG messages shall only be sent within a static ESTABLISHMENT session. The HMAC in the trailer shall not be truncated.
- 5. Before processing the session requests in an OPN, a SCM shall perform the following checks. These rules ensure that two SCMs that simultaneously issue OPNs to each other (on a full duplex channel) to establish a SCADA data session will establish only a single bidirectional data session.
 - a. If SCM A receives an OPN from SCM B while SCM A has an outstanding (unacknowledged) OPN to SCM B and SCM A's address is less than that of SCM B, SCM A shall ignore the OPN.
 - b. If SCM A receives an OPN from SCM B while SCM A has an outstanding OPN to SCM B and A's address is greater than that of SCM B, SCM A shall mark closed any sessions related to its outstanding OPN, cancel any related timers, and respond to the incoming OPN.
 - c. A SCM shall mark closed, without issuing a CLS message, any sessions whose session identifiers appear in session requests in the OPN. This allows reopening of a session with different expiry or other parameters.
 - d. A SCM shall not send an ERR message or other negative acknowledgement upon the reception of an invalid OPN message (e.g., invalid by comparison with received trailer). A SCM shall record this event in its diagnostics/forensics log.

9.2.4.2 ACK message

- 1. An ACK message shall consist of 0 or more session requests:
 - seqOpn - 14 octets
 - numberSessions - 1 octet
- 2. The seqOpn field shall contain the sequence number from the OPN message, and shall be used by the originator of the OPN to match the ACK against the in-progress open operation.
 - a. When issuing an OPN message, a SCM shall start a timer that shall be used to cancel the session establishment operation if no response is received, and shall mark this timer with the sequence number from the OPN.
 - b. A SCM shall ignore any received ACK message for which it does not have an unexpired timer with matching association number.
 - c. The numberSessions octet shall indicate the number of session requests to follow in the message. A session shall be established for each such session request.
 - d. An ACK message shall include a session request for each session included in the corresponding OPN message.

- e. An ACK message may include additional session requests.
- f. Session requests shall be formatted as in the OPN message, except for the following rules.
 - i. If expiry is zero, resolution and base shall also be zero.
 - ii. If either expiry or tolerance is non-zero, resolution shall not be zero.
- 3. The OPN/ACK/BEG mechanism shall provide the capability to allow a SCM to initiate a session and either generate the keys itself, by placing them in the OPN message, or request its peer generate them, by using a zero cipher suite. If a zero cipher suite is used, the peer shall either generate keys and place them in the ACK, or ignore the OPN.
- 4. An ACK message shall be sent using the same static ESTABLISHMENT session as the OPN message it is acknowledging. The HMAC in the trailer shall not be truncated.

9.2.4.3 BEG message

- 1. A BEG message shall consist of 0 or more session requests:
 - seqOpn - 14 octets
 - seqAck - 14 octets
 - numberSessions - 1 octet
- 2. The seqOpn field shall contain the sequence number from the OPN message.
- 3. The seqAck field shall contain the sequence number from the ACK message.
- 4. A SCM shall ignore any received BEG message for which it does not have an unexpired timer with matching sequence numbers.
- 5. The numberSessions octet shall indicate the number of session requests to follow in the message. A session shall be established for each such session request.
- 6. A BEG message shall include a session request for each session included in the corresponding ACK message. A BEG message may include additional session requests.
- 7. Session requests shall be formatted as in the OPN message, except for the following rules.
 - a. If expiry is zero, resolution and base shall also be zero.
 - b. If either expiry or tolerance is non-zero, resolution shall not be zero.
- 8. A BEG message shall be sent using the same static ESTABLISHMENT session as the ACK message it is acknowledging. The HMAC in the trailer shall not be truncated.

9.2.4.4 DTA message

- 1. A DTA message shall consist of:
 - data - message
- 2. A DTA message may consist of either SCADA data or management data.
- 3. A DTA message may be sent on either a static or dynamic session.
- 4. A DTA message containing SCADA data should be sent on a dynamic session.

9.2.4.5 CLS message

- 1. A CLS message shall consist of:
 - message - optional error message

2. The session that is closed is the session that shall be used to transfer the CLS message. The message field shall be an optional error message.
3. A SCM receiving a CLS message with a valid trailer shall treat the session as closed and shall not transmit any further data on the session.
 - a. A CLS message with an invalid trailer shall be silently ignored, regardless of the cipher suite that is in use.
 - b. The receiving SCM shall not use the sequence number/session clock of the CLS message to determine its validity.
4. A CLS message shall be sent within a dynamic session only.
5. A SCM receiving a CLS message for a static session shall discard it quietly, without response to the sending SCM.

9.2.4.6 ERR message

The rationale for this type of message is that it is possible as a result of line noise during session establishment or termination. For one SCM to think a particular sessionId corresponds to an open session, while the other thinks that session does not exist. Since SSPP does not send ACKs or NACKs for DTA messages, ERR ensures that a SCM will not continuously send messages on a session that the receiver thinks does not exist. Returning the trailer of the errant message in the ERR message prevents an adversary from replaying an ERR message and thereby performing a trivial denial-of-service attack.

1. An ERR message shall consist of:

destAddress	- 2 octets
srcAddress	- 2 octets
sessionId	- 1 octet
trailerLength	- 1 octet
receivedTrailer	- trailerLength octets
errorMsg	- optional error message
2. The destAddress, srcAddress, and sessionId fields shall use the corresponding fields from the transport header of the message that the ERR is issued against.
 - a. The field shall contain the trailer of a recently received DTA message for that session. The remainder of the message is an optional error message.
 - b. Any non-ERR message with a non-broadcast destAddress received on a non-open session shall be treated as an "errant" message.
 - c. An ERR message should be returned in response to receipt of any such errant message. Since the session is not open, the SCM receiving the errant message cannot decrypt it nor verify its integrity, and thus cannot distinguish a valid message from one whose sessionId was damaged by line noise or an attacker.
 - d. The SCM shall respond with ERR indicating the sessionId and shall include the trailer of the errant message.
 - e. A SCM receiving an ERR message shall check that the trailer contained in the message is one that was recently sent. If not, the ERR shall be ignored.
 - f. However, if the trailer is recent, the SCM shall close the session that is indicated by the sessionId contained in the ERR message.
 - g. In order to determine if the trailer contained in an ERR message was recently sent, SCMs shall retain the trailers of at least the last three messages transmitted.

3. An ERR message shall be sent on a static session.

9.2.4.7 RCA message

1. A RCA (Request CM Address) message shall consist of only the header and trailer; i.e., it does not contain a payload.
2. A RCA message shall be sent with 0 as the destination address, and the sending CM's address as the source address.
3. A RCA message shall be sent using a sessionId of 0 to indicate to the receiving SCM that this message should be processed; treating the payload (messageType) as unencrypted and the trailer shall contain the four most significant octets of a SHA1 hash.
4. A SCM receiving a RCA message shall respond with a CAR message.

9.2.4.8 CAR message

1. A CAR (CM Address Response) message shall consist of:
 CM address - 2 octets
2. A CAR message shall be sent using address 0 as the source address and the address of the CM that sent the RCA message as the destination address.
3. The CAR message shall be sent using a session id of 0 to indicate to the receiving SCM that this message should be processed; treating the payload (messageType and CM address) as unencrypted and the trailer shall contain the four most significant octets of a SHA1 hash.
4. A CAR message shall only be sent after the reception of a RCA message.

9.2.5 Session state machine

State diagrams are used to describe the various actions that are associated with events at the session layer.

1. An IEEE 1711 compliant implementation shall behave equivalently to the state machine defined by Table 2 which illustrates the behavior of a state machine for static sessions.

Table 2 Session state machine for static sessions

Event	Action
send OPN	3
send DTA	1
rcv OPN	3
rcv ACK	3
rcv DTA	2
rcv CLS	1
rcv ERR	3
rcv bad	1

2. The event "rcv bad" shall be used to indicate the reception of a message on a static session whose verification fails or is otherwise incorrectly formatted or cannot be decrypted.
3. An IEEE 1711 compliant implementation shall perform the following actions specified in Table 2.

- 1: do nothing
 - 2: process payload
 - 3: make appropriate transition in dynamic session state machine for session request
4. A static session shall not be closed via a CLS message. A CLS that is received on a static session shall be ignored.
5. An IEEE 1711 compliant implementation shall behave equivalently to the state machine specified in Table 3. This state machine defines the behavior for one dynamic session. Each dynamic session has an associated state machine to determine its actions.

Table 3 Session state machine for one dynamic session

Action / Event	Current State			
	closed	wait_ACK	wait_BEG	open
send OPN	wait_ACK, 3	X	X	wait_ACK, 3
send DTA	X	X	X	open, 1
send CLS	X	X	X	closed, 1
rcv OPN	wait_BEG, 5	wait_BEG, 4	wait_BEG, 8	wait_BEG, 11
rcv ACK	closed, 1	open, 6	wait_BEG, 1	open, 1
rcv BEG	closed, 1	wait_ACK, 1	open, 9	open, 1
rcv DTA	closed, 7	wait_ACK, 1	wait_BEG, 1	open, 2
rcv CLS	closed, 1	wait_ACK, 1	wait_BEG, 1	closed, 1
rcv ERR	closed, 1	closed, 10	closed, 9	closed, 1
rcv bad	closed, 1	wait_ACK, 1	wait_BEG, 1	open, 1
ACK timeout	X	closed, 1	X	X
BEG timeout	X	X	closed, 1	X

Note: X = cannot occur

6. An IEEE 1711 compliant implementation shall perform the following actions specified in Table 3:
- 1: do nothing
 - 2: process payload
 - 3: start ACK timer
 - 4: cancel ACK timer, send ACK, start BEG timer
 - 5: send ACK, start BEG timer
 - 6: cancel ACK timer, send BEG
 - 7: send ERR
 - 8: cancel BEG timer, send ACK, start BEG timer
 - 9: cancel BEG timer
 - 10: cancel ACK timer
 - 11: close current session D, send ACK, start BEG timer
7. The event "send OPN" shall be used to indicate sending an OPN message on some static session (designated S) and shall indicate a dynamic session (designated D) in the payload.
- a. The event "send CLS" shall be used to indicate sending a CLS message on session D to close the session.
 - b. The event "send DTA" shall be used to indicate sending a DTA message on dynamic session D. The event "rcv OPN" shall be used to indicate receiving an OPN message on some static session S that contains a session request for dynamic session D.

- c. The event "rcv ACK" shall be used to indicate receiving an ACK message on some static session S and indicating D in the payload.
- d. The event "rcv CLS" shall be used to indicate receiving a CLS message on session D.
- e. The event "rcv DTA" shall be used to indicate receiving a DTA message on dynamic session D.
- f. The event "rcv ERR" shall be used to indicate receiving an ERR message on some static session S and indicating dynamic session D in the payload.
- g. The event "rcv bad" shall be used to indicate receiving a message on dynamic session D whose checksum verification fails or is otherwise incorrectly formatted or cannot be decrypted.

9.2.6 Broadcast messages

1. Broadcast sessions are a special type of data session that shall be used for SCADA broadcast data.
2. A BROADCAST session can be opened between a pair of SCMs at the same time as a DATA session is opened.
 - a. However, the session shall be interpreted with respect to the Publisher's source SCM address and the broadcast SCM destination address (0xFFFF), where the Publisher shall be defined to be the SCM that placed the broadcast session request in the ACK or BEG.
 - b. That is, if the broadcast request is placed in the ACK, then the SCM sending the ACK shall be the Publisher.
3. In the case of the OPN, ACK, and BEG sequence, the DATA session shall be established to transfer SCADA messages.
 - a. If SCADA broadcast is possible (as indicated by the configuration setting), the ACK or BEG (depending on who initiated with the OPN) shall also have a session request with the parameters to be used for the encrypted broadcast session.
 - b. If the DATA session is considered active, then the associated broadcast session shall be also considered active, as are all other sessions requested in the OPN/ACK/BEG sequence.
 - c. Broadcast sessions shall be handled as unicast, and Subscriber SCMs shall not send using the broadcast session parameters.
4. The SCM assigned the role as the Publisher (usually associated with the SCADA host) shall generate the broadcast session parameters to be sent to each receiving SCM (usually in the field). A typical manner in which a SCM may operate is to dynamically generate cryptographic keys for broadcast as it issues the first OPN for a dynamic session. These same keys can then be issued other SCMs in subsequent OPN messages. Other methods are possible, as long as all Subscribers SCMs having an open dynamic data session with the Publisher SCM will also have an open broadcast session using a common set of keys for the broadcast session. Periodically (user settable) the Publisher should close all broadcast sessions and re-establish them with new keys.
5. A broadcast session shall be unidirectional. That is, the broadcast address 0xFFFF shall not be used as the source address of a message.

9.2.7 Forensics and intrusion detection

1. The reception of an OPN message that failed validation shall be identified as an item to be recorded in the diagnostics/forensics log of the cryptographic module.

2. The log shall include (but is not limited to) the sending cryptographic module address, the session which was used for the message, and the reason for failure (e.g., failed MAC, incorrect session).

9.3 Transport layer

The purpose of the transport layer is to wrap and protect a session layer message so as to preserve its integrity and confidentiality against an adversary. The transport layer ensures that an adversary cannot forge or alter messages, read messages, reorder messages, or replay old messages.

1. A transport layer message shall consist of a header, a variable length payload, and a fixed length trailer (which is cipher suite dependant).
2. The transport layer shall rely on the link layer to identify the boundaries between the header and payload, and between the payload and trailer.
3. At the sender, the transport layer shall encrypt the message received from the session layer to form the payload.
4. The transport layer shall generate a header and a trailer, and the three sections are passed to the link layer.
5. The specific method of encryption depends on the cipher suite associated with the session the message is transmitted on.
6. At the receiver, the transport layer receives the header, payload, and trailer from the link layer.
7. The transport layer shall check the validity of the header, possibly including a sequence number, prior to processing the payload.
8. The transport layer shall decrypt the payload, and may forward decrypted data to the session layer as soon as it is available, or may withhold all data until after processing the trailer, depending on the cipher suite.
9. The transport layer shall compute a cipher-suite-dependent check value of the message and shall compare it to the check value in the trailer. If the check value does not match, the SCM shall discard the message (if holding back the entire message to verify before transmitting to the attached SCADA unit), or any portion that has not been sent to the attached SCADA unit (if transferring the message as it is decrypted).
10. An IEEE 1711 compliant implementation shall format a transport layer message as follows:
header:
 type - 1 octet
 destAddress - 2 octets
 srcAddress - 2 octets
 sessionId - 1 octet
 sequence - variable octets
payload (encrypted per cipher suite):
 message - variable size message (see session layer)
 padding - dependant upon the cipher suite
trailer:
 check value - dependent upon cipher suite
11. If any of the transport layer fields of a received message are found to be invalid, the message shall be silently discarded. The SCM shall log the discard event in the diagnostic/forensic log of the cryptographic module.

9.3.1 Transport layer header

1. The type field of the header shall contain a protocol version number (currently 1) in the upper three bits (7:5), an alert flag in the fourth bit (4:4), and the message type in the lower four bits (3:0).
2. The message type shall be one of the following values:
 - OPN = 1
 - ACK = 2
 - DTA = 3
 - CLS = 4
 - ERR = 5
 - BEG = 6
 - RCA = 7
 - CAR = 8
3. The alert flag may be set in any DTA or CLS message and shall be used to indicate by the sending SCM to the receiving SCM that it has encountered a vendor-specified condition (e.g., key expiration, tampering detection). The conditions on which an alert bit is set or cleared shall be vendor defined.
4. The destAddress and srcAddress shall be the SCM addresses of the destination and source SCMs.
 - a. A SCM shall ignore any messages not containing its own address or the broadcast address 0xFFFF in the destAddress field.
 - b. A SCM shall set the srcAddress field to its own address in all messages it sends.
5. The sessionId shall be used to specify the session in which the payload is encrypted.
6. The transport layer header shall not be encrypted.

9.3.2 Transport layer payload

1. The transport layer payload shall contain a session layer message whose type is indicated in the transport layer header.
2. The transport layer payload shall be encrypted and padded as defined for the session's cipher suite.

9.3.3 Transport layer trailer

1. The transport layer trailer shall be a check value of the octets constituting the transport layer header and the transport layer payload. The function for computing the check value and the transmitted length (which may involve truncation of the value) depends on the cipher suite.
2. The transport layer trailer shall not be encrypted.

9.3.4 Transport layer sequence numbers

1. Sequence numbers shall be 112 bits (14 octets) for static sessions and variable length for dynamic sessions.
2. For static sessions, sequence numbers shall be either 112-bit random numbers, or shall be values pulled from a 112-bit counter that is persistent over the lifetime of the device (and will never repeat). If from a persistent counter, only one 112-bit persistent counter is needed per SCM; that is, all communications with all peer SCMs can share the same counter.
3. For dynamic sessions, the length of the sequence number shall be negotiated during session exchange by including an extra value in each session request.

- a. The sequence number length can vary from 16 bits to 112 bits, in multiples of 8 (2 to 14 octets).
- b. Dynamic session sequence numbers shall be either sequentially increasing counter values, or monotonically increasing clock values.
- c. The length of the sequence number shall be chosen at session negotiation so that the session will expire before the counter or clock overflows.
- d. A SCM receiving a message on a dynamic session shall determine the length of the sequence number by looking the session number up in a table - no representation of the sequence number length is present in the header.
- e. For dynamic sessions, the SCM shall also ensure that a received message has a greater sequence number than the last one received on the same session, except for CLS messages.
- f. If a session clock is in use, the SCM shall also ensure that the sequence number is within a tolerance of the current session time, except for CLS messages.

9.3.5 Transport layer error handling

1. For cipher suites that require holdback, a receiver shall verify the check value in the transport layer trailer of a message before taking any action that makes use of the decrypted payload; if the check value does not verify, the receiver shall discard the message.
2. The SCM shall log the discard event in the diagnostic/forensic log of the cryptographic module.

9.4 Link layer

The link layer is concerned with the formatting of SSPP messages as individual octets on the serial communications link. Its function is to identify the beginning and end of an SSPP message and to locate the transport layer header, payload, and trailer sections of a message. The link layer message layout is designed to permit mixing SSPP messages and most types of SCADA messages without interference on the same communications link for mixed-mode operation.

There are two link layer specifications, one for use on links with 8-bit data transfers, and one for use on links with 7-bit data transfers. The 8-bit link layer should be used whenever possible. The 8-bit link layer delimits an SSPP message with two-octet markers for the start of the message, start of the trailer, and end of the message. To avoid interference with plaintext SCADA traffic, these markers are configurable. The 7-bit link layer has no configurable markers.

9.4.1 8-bit link layer message format

1. A link layer message shall consist of three sections: header, payload, and trailer.
2. To identify the beginning and end of these elements and the overall message, a SCM shall implement the following four sequences:

ESC SOM	Start Of Message
ESC SOT	Start Of Trailer
ESC EOM	End Of Message
ESC ESC	Escape

- a. The first three sequences shall be used to identify the different parts of a message.
- b. The last sequence, ESC ESC, shall be sent when the value of the ESC octet is sent as ordinary data (see below for details; the other octet values of SOM, SOT and EOM do not need to be escaped because they only have special meaning when preceded by the ESC octet).

- c. In addition, a SCM may implement 1 or more of the following additional sequences:
 - ESC RC1 Replacement Character 1
 - ...
 - ESC RCn Replacement Character n
 - i. These sequences shall be used as to avoid transmitting particular octets that might interfere with unprotected SCADA messages.
 - ii. The octets whose presence should be avoided in an SCM message shall be denoted as SC1 through SCn (Special SCADA Character 1 through Special SCADA Character n).
- d. Each of ESC, SOM, SOT, EOM, and RC1 ... RCn shall be configurable octets and all shall be different.
 - i. If the SCMs will be operating in mixed-mode (both SCM and SCADA messages carried on the same link), these octets should be chosen to avoid octets having special meaning for the SCADA protocol.
 - ii. All SCMs sharing the same communications link shall use the same configured octets.
 - iii. Because this is a configuration issue, the configuring entity shall ensure that this requirement is met.
3. To send a message, the transport layer provides the link layer with three message sections: header, payload, and trailer. The link layer shall format and transmit these as follows:

ESC SOM header payload ESC SOT trailer ESC EOM
4. During transmission of the header, payload, or trailer, the link layer shall transmit ESC ESC for any occurrence of the ESC octet followed by any of the special characters ESC, SOM, SOT, EOM, SCi, or RCi.
5. When receiving a message, the link layer shall identify the three sections of the message and indicate them to the transport layer.
 - a. A SCM receiving ESC ESC in the header, payload, or trailer shall discard one of the ESC octets and treat the other as ordinary data.
 - b. The receiver shall separate the header from the payload by counting the first sequence length + 6 octets as header.
6. If a SCM supports 1 or more sequences ESC RC1 thru ESC RCn, it shall support an equal number of configurable octets SC1 thru SCn.
 - a. These SCADA Characters (octets) are those with special meaning to SCADA devices that shall never be sent in SSPP messages on the communications link.
 - b. If configured, SC1 thru SCn shall all be different, and shall be different from ESC, SOM, SOP, SOT, EOM, and RC1 thru RCn.
 - c. During transmission of the header, payload, or trailer, the link layer shall transmit ESC RCi for any occurrence of the SCi octet.
 - d. A SCM receiving ESC RCi in the header, payload, or trailer of an SSPP message shall behave as if SCi were received as an ordinary octet.
7. A sender should avoid leaving gaps between octets in the link layer message to simplify mixed mode operation with timing sensitive SCADA protocols such as MODBUS. Since the payload is usually encrypted by the transport using a block cipher, this may mean

delaying the ESC SOM at the beginning of the message until enough octets of the SCADA message have been received so that when a full cipher block becomes available, the last octet of the header has just been transmitted.

9.4.2 8-bit link layer error conditions during receive

1. If the link layer encounters any of the defined markers out of order, the SCM shall discard the rest of the message and return to listening for ESC SOM.
 - a. The link layer should also signal the next layer that an error occurred so that the next layer can flush any partial message.
 - b. In the event that the out-of-order marker is ESC SOM, the current message shall be discarded and a new message begun.
2. The link layer shall implement a timer that tracks the time elapsed since the last octet was received.
 - a. If this timer exceeds some configurable value, the partial message shall be discarded, and the SCM shall return to listening for ESC SOM.
 - b. Other actions may also be necessary, such as terminating a partial SCADA command that has been sent out the cleartext port.

9.4.3 8-bit link layer sender state machine

1. Table 4 describes the state transitions for a finite state machine that shall be used to send link layer messages.

Table 4 Eight-bit Link layer sender state machine

Action (char to send)	Current State				
	NotInM	InM	esclnM	InT	esclnT
StartM	InM 1	X	X	X	X
StartT	X	InT 2	InT 3	X	X
EndM	X	X	X	NotInM 4	NotInM 5
ESC	X	esclnM 6	esclnM 8	esclnT 6	esclnT 8
SOM	X	InM 6	InM 8	InT 6	InT 8
SOT	X	InM 6	InM 8	InT 6	InT 8
EOM	X	InM 6	InM 8	InT 6	InT 8
SCi	X	InM 7	InM 9	InT 7	InT 9
RCi	X	InM 6	InM 8	InT 6	InT 8
other	X	InM 6	InM 6	InT 6	InT 6

Note: X = cannot occur

2. The link layer implementation of a SCM shall operate in an equivalent manner to the state machine specified in Table 4. Given a state (column in the table) and an input character (row), Table 4 specifies the next state for the state machine and an action to take.
 - a. The actions StartM, StartT, and EndM shall be used to indicate that the sender wishes to begin a message, begin a trailer section, or end a message. These actions shall appear in the order StartM, StartT, EndM, separated by one or more characters.
 - b. The state NotInM shall be used to indicate the sender has not begun a message.
 - c. The state InM shall be used to indicate that the sender is in the message (header or payload) section.
 - d. The state InT shall be used to indicate that the sender is in the trailer section.

- e. The states `escInM` and `escInT` shall be used to indicate the sender just previously sent an ESC character while in the message (header/payload) or trailer section.
 - f. The initial state shall be `NotInM`.
 - g. In the event of any discrepancy between this transition table and the description in the preceding clauses, this transition table shall take precedence.
3. An IEEE 1711 compliant implementation shall perform the following actions:
- 1: send ESC, SOM
 - 2: send ESC, SOT
 - 3: send ESC, ESC, SOT
 - 4: send ESC, EOM
 - 5: send ESC, ESC, EOM
 - 6: send current character
 - 7: send ESC, RCi
 - 8: send ESC, current character
 - 9: send ESC, ESC, RCi

9.4.4 8-bit link layer receiver state machine

- 1. The state transitions for an eight-bit finite state machine that recognizes received link layer messages shall comply with Table 5.
 - a. The link layer implementation of a SCM shall operate in an equivalent manner to this state machine.
 - b. Given a state (column in the table) and an input octet or event (row), the table gives the next state for the state machine and an action to take.

Table 5 Eight-bit link layer receiver state machine

Received Octet (event)	Current State					
	lookSOM	escSOM	lookSOT	escSOT	lookEOM	escEOM
ESC	escSOM 1	escSOM 1	escSOT 1	lookSOT 2	escEOM 1	lookEOM 2
SOM	lookSOM 1	lookSOT 5	lookSOT 2	lookSOT 4	lookEOM 2	lookSOT 4
SOT	lookSOM 1	lookSOM 1	lookSOT 2	lookEOM 6	lookEOM 2	lookSOM 3
EOM	lookSOM 1	lookSOM 1	lookSOT 2	lookSOM 3	lookEOM 2	lookSOM 7
RCi	lookSOM 1	lookSOM 1	lookSOT 2	lookSOT 9	lookEOM 2	lookEOM 9
SCi	lookSOM 1	lookSOM 1	lookSOT 2	lookSOT 8	lookEOM 2	lookEOM 8
other	lookSOM 1	lookSOM 1	lookSOT 2	lookSOT 9	lookEOM 2	lookEOM 9
Timeout	lookSOM 1	lookSOM 1	lookSOM 3	lookSOM 3	lookSOM 3	lookSOM 3

- 2. An IEEE 1711 compliant implementation shall perform the following actions:
 - 1: do nothing
 - 2: add octet to current header/payload/trailer
 - 3: discard any partial message
 - 4: discard any partial message, start header
 - 5: start message (header/payload)
 - 6: start trailer
 - 7: end of message
 - 8: add SCi to current header/payload/trailer
 - 9: add ESC, current character to current header/payload/trailer

9.4.5 7-bit link layer

1. An implementation may provide the optional 7-bit link layer. IEEE P1711 support for this option is optional, but if implemented it shall conform to the following requirements.
 - a. The 7-bit link layer shall format an SSPP message as:
(base64-header-and-payload * base64-trailer)
 - b. The '(', '*', and ')' are those ASCII characters with decimal values 40, 42, and 41 respectively.
 - c. The sections base64-header-and-payload, and base64-trailer shall be encoded by taking each successive 6 bits of data (zero-padded to fill the last 6 bits as needed) and using it to select one of the 64 characters 'A'-'Z', 'a'-'z', '0'-'9', '+', or '/' according to Table 6.

Table 6 Seven-bit link layer ASCII requirements

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

2. When decoding a message using the 7-bit link layer, each received character in the range 'A'-'Z', 'a'-'z', '0'-'9', '+', or '/' contributes 6 bits to the current message section.
 - a. If any unexpected character is received, the current partial message shall be discarded and the receiver shall resume looking for '(' to indicate the start of a new message.
 - b. Table 7 represents the actions a receiver shall take.

Table 7 Seven-bit link layer state machine

Received Character (Event)	Current State		
	lookSOM	lookSOT	lookEOM
'('	lookSOT 5	lookSOT 4	lookSOT 4
'*'	lookSOM 1	lookEOM 6	lookSOM 3
')'	lookSOM 1	lookSOM 3	lookSOM 7
M	lookSOM 1	lookSOT 2	lookEOM 2
other	lookSOM 1	lookSOM 3	lookSOM 3
Timeout	lookSOM 1	lookSOM 3	lookSOM 3

- c. A received character M shall be used to indicate a character in the range 'A'-'Z', 'a'-'z', '0'-'9', '+', or '/'.
3. An IEEE 1711 compliant implementation shall perform the following actions:
 - 1: do nothing
 - 2: add 6 bits from character to current header/payload/trailer
 - 3: discard any partial message
 - 4: discard any partial message, start header
 - 5: start header
 - 6: start trailer
 - 7: end of message

9.5 Cipher suites

The cipher suites are an enumerated list.

1. Cipher suite numbers shall be two octets (16 bits), and shall be expressed in hexadecimal.
2. No cipher suite may be numbered 0.
3. The top-most bit, 0x8000, shall be used to indicate that the suite is vendor defined, and might not be interpreted correctly by SCMs manufactured by a different vendor. That is, IEEE 1711 does not ensure interoperability between SCMs manufactured by different vendors, nor does it ensure interoperability between versions of SCMs manufactured by the same vendor.

VENDOR_DEFINED = 0x8000 – 0xFFFF
4. The cipher suites defined by IEEE 1711 that shall be implemented are 0x0002, 0x0007, and 0x0009. The remaining suites are optional. Implementations shall not issue OPN or ACK messages using any values other than those, unless the VENDOR_DEFINED bit is set.

9.5.1 IEEE 1711 defined cipher suites

There are currently 10 cipher suites defined by IEEE 1711 for use with SCMs. The algorithms for cryptographic hashing, message authentication, and encryption draw upon the processes developed by the National Institute of Standards and Technology (NIST) and described in Federal Information Processing Standard (FIPS) Publications and Special Publications.

1. SHA1 and SHA256 refer to the first two hash algorithms defined in FIPS 180-2—Secure Hash Standard (NIST, August 1, 2002).
2. HmacSHA1 and HmacSHA256 refer to the message authentication described in FIPS 198—The Keyed-Hash Message Authentication Code, using SHA1 and SHA256, respectively (NIST, March 6, 2002).
3. AES128, AES192, and AES256 are the encryption algorithms described in FIPS 197—Advanced Encryption Standard (NIST, November 26, 2001).
4. CTR mode refers to the encryption mode (using AES) described in SP 800-38A—Recommendation for Block Cipher Modes of Operation (NIST, 2001).
5. PE mode refers to an encryption mode combining CTR mode and ECB mode (see SP 800-38A), and developed for streaming SCADA messages. It relies on the SCADA protocol's ability to detect incorrect SCADA messages. PE mode is more fully described in the Appendix of SP 800-38A.

9.5.1.1 Whitening and initialization vector calculation

Several of the suites refer to values called X, Y, S, and R that are defined as follows.

1. If the cipher suite is used to encrypt data for a static session or a broadcast session, X, Y, S, and R shall all be zero.
2. Otherwise, the cipher suite is either used to encrypt or decrypt a message on a dynamic session.
 - a. X shall be a 128-bit value formed by concatenating the 16-bit SCM address of the encrypting SCM and the 112-bit sequence number of the OPN or ACK message that the encrypting SCM sent.
 - b. The encrypting SCM sent either an OPN or an ACK depending on which SCM initiated the session.
3. Y shall be a 128-bit value formed by concatenating the 16-bit SCM address of the decrypting SCM and the 112-bit sequence number of the OPN or ACK message that the decrypting SCM sent.
4. If the cipher suite is used to encrypt, S shall be:
$$S = \text{Enc}(\text{Enc}(X) \text{ XOR } Y)$$
, where Enc shall be a single block AES encryption using the cipher suite key.
5. If the cipher suite is used to decrypt, R shall be:
$$R = \text{Enc}(\text{Enc}(Y) \text{ XOR } X)$$
6. S and R may be computed once at session negotiation and saved for the duration of the session. This is a vendor choice.

9.5.1.2 CTR mode with holdback

1. There are two cipher suites that use CTR-mode of AES for encryption, and full holdback: 0x0001 and 0x0004.
 - a. Suite 0x0001 shall use HmacSHA1 with 160 bit keys
 - b. Suite 0x0004 shall use HmacSHA256 with 256 bit keys for message authentication.
 - c. These cipher suites shall only be used for dynamic sessions.
2. Each 16-octet block of the payload shall be XORed with a 16-octet value.
 - a. This value shall be generated by encrypting the following data with the session's encryption key using AES/ECB:

block number within msg (first block is 0)	- 2 octets
sequence number or random value	- 14 octets
 - b. Then, the result shall be XORed with S. The sequence number shall be padded on the left with zeros to fill 14 octets if necessary.
3. The transport layer payload shall not be padded. If the final block of the payload is less than 16 octets, the output of the encryption step shall be truncated to the input block length.
4. For sessions using these cipher suites, the transport layer trailer shall be HmacSHA MAC constructed in the following manner.

- a. For dynamic non-broadcast sessions, the input to the MAC shall be the value X, then Y, then the unencrypted transport layer header, then the encrypted transport layer payload.
- b. For broadcast sessions, the input to the MAC shall be the unencrypted transport layer header and the encrypted transport layer payload. The MAC shall be truncated, if necessary, by dropping octets from the right.
5. For sessions using these cipher suites, the receiver shall check the MAC in the transport layer trailer of a message before taking any action that makes use of the decrypted payload. If the MAC does not verify, the receiver shall discard the message.
6. The configuration parameters for this cipher suite shall be formatted as follows for use in an OPN, ACK, or BEG message:

suiteNumber	- 2 octets = 0x0001 or 0x0004
macLength	- 1 octet
AES key	- 16 octets
HmacSHA key	- 20 or 32 octets
7. The macLength shall represent the length to which the MAC is truncated, in octets.

9.5.1.3 PE mode with no holdback

1. There are two cipher suites that use AES in PE-mode with no holdback of the message: 0x0002 and 0x0005.
 - a. Suite 0x0002 shall use HmacSHA1 with 160 bit keys
 - b. Suite 0x0005 shall use HmacSHA256 with 256 bit keys for message authentication.
 - c. These cipher suites shall be used for dynamic sessions only, not for static sessions.
 - d. They shall only be used for SCADA data sessions when the SCADA protocol is known to use a 16-bit or longer CRC checksum.
2. During encryption, a 16-octet whitener shall be constructed by encrypting the following with AES/ECB using the session encryption key:

block number within msg (first block is 0)	- 2 octets
sequence number, or zeros	- 14 octets
3. Then, the result shall be XORed with S as defined in Clause 9.5.1.1.
 - a. The sequence number shall be padded on the left if necessary with zeros to fill 14 octets.
 - b. Then each 16-octet block of the padded payload shall be XORed with the whitener.
 - c. This result shall be encrypted with AES/ECB using the session encryption key.
 - d. The output of the encryption shall again be whitened by XORing with the same whitener value.
4. The transport layer payload shall be padded with from 1 to 16 octets so that the payload and padding are a multiple of 16 octets in length.
 - a. The first octet of padding shall be 0x80, and all remaining octets of padding shall be zeros.
 - b. On receipt of the last encrypted block, the decrypted block and any portion of the previously decrypted blocks not forwarded shall be silently discarded if the SCM cannot detect valid padding.

5. For sessions using these cipher suites, the transport layer trailer shall be HmacSHA MAC constructed in the following manner.
 - a. For dynamic non-broadcast sessions, the input to the MAC shall be the value X, then Y, then the unencrypted transport layer header, then the encrypted transport layer payload.
 - b. For broadcast sessions, the input to the MAC shall be the unencrypted transport layer header and the encrypted transport layer payload.
6. The MAC shall be truncated, if necessary, by dropping octets from the right.
7. For DTA messages using these cipher suites, the receiver shall verify that the sequence number is within the tolerance of the session clock.
 - a. If not, the message shall be silently discarded.
 - b. Otherwise, the receiver should forward each block of decrypted data to the receiving SCADA equipment as soon as it is decrypted.
 - c. The receiver shall update its receive sequence number to that of a received message whenever that message's sequence number is newer and within the tolerance of the session clock, even if the MAC in the trailer is incorrect.
 - d. These cipher suites shall not be used without a session clock.
8. The configuration parameters for this cipher suite shall be formatted as follows for use in an OPN, ACK, or BEG message:

suiteNumber	- 2 octets = 0x0002 or 0x0005
macLength	- 1 octet
AES key	- 16 octets
HmacSHA1 key	- 20 or 32 octets
9. The macLength shall represent the length to which the MAC is truncated, in octets.
10. PE-mode as used in these cipher suites shall not use the MAC (HmacSHA1 or HmacSHA256) to authenticate the received, encrypted message before the SCADA message is decrypted and sent to the local, attached SCADA unit.
 - a. PE-mode shall rely on the message validation in the SCADA protocol (e.g., CRC) to identify a mangled SCADA message that would result from noise or other modification of the SSPP message.
 - b. This reliance is a trade-off between the theoretical reductions in security vs. the increased latency that occurs with other cipher suites using full holdback of messages for authentication.

9.5.1.4 Clear text with hash

1. There are two cipher suites for cleartext with hash: 0x0003 and 0x0006.
 - a. Suite 0x0003 shall use SHA1.
 - b. Suite 0x0006 shall use SHA256 for message authentication.
 - c. These cipher suites shall not be used to transfer SCADA data because they provide no security. They are intended to be used for certificate exchange by vendor-defined key management extensions.
 - d. The payload of messages using these cipher suites shall not be encrypted.
 - e. The payload of messages using these cipher suites shall not be padded.

2. For sessions using this cipher suite, the transport layer trailer shall be a SHA hash of the transport layer header and payload.
 - a. The receiver shall check the hash in the transport layer trailer of a message before taking any action that makes use of the payload.
 - b. If the hash does not verify, the receiver shall discard the message.
3. The security parameters for this cipher suite shall be formatted as follows for use in an OPN, ACK, or BEG message:
suiteNumber - 2 octets = 0x0003 or 0x0006
truncatedLength - 1 octet
4. The truncatedLength shall represent the length to which the hash must be truncated, in octets.

9.5.1.5 Clear text and MAC with holdback

1. There are two cipher suites for MAC only (no encryption/cleartext) with holdback: 0x0007 and 0x0008.
 - a. Suite 0x0007 shall use HmacSHA1 with 160 bit keys
 - b. Suite 0x0008 shall use HmacSHA256 with 256 bit keys for message authentication.
 - c. These cipher suites shall be used only for dynamic session, not for static sessions.
 - d. The payload shall not be encrypted or padded.
2. For sessions using these cipher suites, the transport layer trailer shall be HmacSHA MAC constructed in the following manner.
 - a. For dynamic non-broadcast sessions, the input to the MAC shall be the value X, then Y, then the transport layer header, then the transport layer payload.
 - b. For broadcast sessions, the input to the MAC shall be the transport layer header and the transport layer payload.
3. The MAC shall be truncated, if necessary, by dropping octets from the right.
4. For sessions using this cipher suite, the receiver shall check the MAC in the transport layer trailer of a message before taking any action that makes use of the payload. If the MAC does not verify, the receiver shall discard the message.
5. This cipher suite shall be formatted as follows for use in an OPN, ACK, or BEG message:
suiteNumber - 2 octets = 0x0007 or 0x0008
macLength - 1 octet
HmacSHA key - 20 or 32 octets
6. The macLength shall represent the length to which the MAC must be truncated, in octets.

9.5.1.6 CBC mode with holdback

1. There are two cipher suites that use AES in Cipher Block Chaining (CBC) mode with holdback of the message for verification of the HMAC before processing: 0x0009 and 0x000A.
 - a. Suite 0x0009 shall use HmacSHA1 with 160 bit keys
 - b. Suite 0x000A shall use HmacSHA256 with 256 bit keys for message authentication.
 - c. These cipher suites may be used for static or dynamic sessions.

2. The transport layer payload shall be padded with from 1 to 16 octets so that the payload and padding are a multiple of 16 octets in length.
 - a. The first octet of padding is 0x80, and all remaining octets of padding shall be zeros.
 - b. On receipt of the last encrypted block, the decrypted block and any portion of the previously decrypted blocks shall be silently discarded if the SCM cannot detect valid padding.
3. During encryption, each 16-octet block of the padded payload shall be XORed with the 16 octets of the previous cipher block.
 - a. The first block shall be XORed with 16 octets of an Initialization Vector (IV) that shall be constructed by encrypting the following data with AES/ECB using the session encryption key:

zeros	- 2 octets
sequence number	- 14 octets
 - b. Then the result shall be XORed with S (defined above).
4. For sessions using these cipher suites, the transport layer trailer shall be HmacSHA MAC constructed in the following manner.
 - a. For dynamic non-broadcast sessions, the first 32 octets of input to the MAC shall be the values X followed by Y as defined above. Following shall be the unencrypted transport layer header, then the encrypted transport layer payload.
 - b. For static and broadcast sessions, the input to the MAC shall be the unencrypted transport layer header and the encrypted transport layer payload.
5. The MAC shall be truncated, if necessary, by dropping octets from the right.
6. For sessions using these cipher suites, the receiver shall check the MAC in the transport layer trailer of a message before taking any action that makes use of the decrypted payload. If the MAC does not verify, the receiver shall discard the message.
7. The configuration parameters for this cipher suite shall be formatted as follows for use in an OPN, ACK, or BEG message:

suiteNumber	- 2 octets = 0x0009 or 0x000A
macLength	- 1 octet
AES key	- 16 octets
HmacSHA key	- 20 or 32 octets
8. The macLength shall represent the length to which the MAC is truncated, in octets.

9.5.1.7 Vendor defined cipher suites

1. The definition of a vendor-defined cipher suite may be proprietary, but shall fit within the constraints of the session, transport, and link layer operations.
2. As noted in Clause 9.5 (3), a vendor-defined cipher suite shall be assigned a value within the range of 0x8000 to 0xFFFF, i.e., the most significant bit shall be set to 1.
3. Vendors shall not be permitted to claim IEEE 1711 compliance if they specify a cipher suite that uses data-independent, XOR-based stream ciphers with no holdback. However, it is possible that stream cipher variants (e.g., with forward error propagation), may be considered adequately secure. Therefore, users/operators/owners of cryptographic modules should investigate vendor-specified cipher suites before their use, including through independent verification.

9.6 Key management

Key management for SSPP involves the establishment of static sessions with encryption and authentication keys and other parameters.

9.6.1 Initial key loading

Establishment of the encryption and authentication keys for the static session shall be done in a secure fashion designated by the manufacturer of the cryptographic modules and in accordance with the FIPS 140-2 requirements (NIST , 2001).

9.6.2 In-band transfer of keys

1. In-band transfer of keys between operational cryptographic modules shall be limited to the establishment of dynamic sessions, specifically the OPN and ACK messages that are sent using the static ESTABLISHMENT session. See the requirements for OPN and ACK messages, and the corresponding cipher suite descriptions for the format of the messages used to establish the encryption and authentication keys for dynamic sessions.
2. Keys for static sessions shall not be transferred in-band.

9.6.3 Revocation of keys

Encryption and authentication keys and other critical security parameters shall be securely destroyed upon their revocation; e.g., when the session associated with them is closed.

9.7 SSPP provisioning

1. Subsequent versions of the Serial SCADA Protection Protocol (SSPP) shall be described in later versions of IEEE 1711, or in standards that supersede IEEE 1711, or in standards that complement IEEE 1711.
2. It is intended that subsequent versions of SSPP shall be upwardly compatible; i.e., cryptographic modules supporting an earlier version of SSPP shall be able to communicate with modules with later SSPP versions.
3. It also is intended that subsequent versions of SSPP shall include the capability of querying cryptographic modules as to their supported version. Modules that do not respond to such queries are assumed to be IEEE 1711 SSPP compliant modules.

9.8 SSPP management messages

1. A message with a zero length payload received on a session of type MANAGEMENT shall be a query for a vendor-defined version string.
 - a. A SCM receiving such a message shall respond on the same session with a vendor-defined version string of length one or more.
 - b. The version string shall include at least information identifying the vendor and version number of the implementation.
 - c. The vendor identification should be chosen so as to be unlikely to conflict with the identification information of another vendor.
 - d. For example, the ScadaSafe implementation returns a string of the form:
ScadaSafe V.V.V built YYYY/MM/DD HH:MM -TZTZ
2. Subsequent versions of the Serial SCADA Protection Protocol (SSPP) shall be described in later versions of IEEE 1711, or in standards that supersede IEEE P1711, or in standards that complement IEEE 1711.

Annex A Bibliography

- AGA. (March 14, 2006). *AGA 12-1: Cryptographic Protection of SCADA Communications*. American Gas Association.
- ANSI. (1998). *ANSI X9.69: Framework for Key Management Extensions, American National Standards for Financial Services*. ANSI.
- IEEE. (Seventh Edition - 2000). *IEEE 100: The Authoritative Dictionary of IEEE Standard Terms*. Standards Information Network: IEEE Press.
- NIST. (August 1, 2002). *FIPS PUB 180-2: Secure Hash Standard*. Standard, National Institute of Standards and Technology.
- NIST. (November 26, 2001). *FIPS PUB 197: Advanced Encryption Standard*. National Institute of Standards and Technology.
- NIST. (March 6, 2002). *FIPS PUB 198: The Keyed-Hash Message Authentication Code (HMAC)*. Standard, National Institute of Standards and Technology.
- NIST. (2001). *NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation*. National Institute of Standards and Technology.
- NIST. (February 1998). *NIST SP800-17: Mods of Operation Validation System (MOVS) - Requirements and Procedures*.

Annex B Requirements traceability to cryptographic protocol specification

This informative annex provides a high-level assessment summary of the requirements for module design, cyber security, test and evaluation, installation and commissioning, and performance constraints that are addressed by the SSPP specification in Clause 9.

B.1 Module design (Clause 4)

B.1.1 Operating environment

SSPP does not address requirements for the operating environment.

B.1.2 SCM alarms and data reporting

SSPP requires logging, session establishment and message formats for reporting. SSPP does not address client responsibilities for processing logs; that is a local matter.

SSPP does not address monitoring for inactivity so as to automatically terminate a session.

B.1.3 SCM secure enclosure and storage

SSPP does not address requirements for secure enclosure and storage.

B.1.4 Communication components

SSPP address all communication requirements SCM session establishment and message exchanges.

B.1.5 SCM documentation

SSPP does not address requirements for SCM documentation.

B.1.6 Quality requirements

SSPP requirements address all quality requirements in the context of minimizing the impact on functional communication network operation and performance.

B.2 Cyber security (Clause 5)

SSPP addresses all cyber security requirements.

B.3 Test and evaluation (Clause 6)

ScadaSafe is an implementation of SSPP and is publically available at no cost.

GTI conducted extensive laboratory and field testing of the SSPP implementations. SCM manufacturers have conducted extensive factory test and field testing of SSPP implementations. Utilities have conducted extensive testing SSPP implementation using SCMs provided by manufacturers.

SNL has conducted extensive analysis of SSPP's security requirements and all recommendations are included in IEEE 1711.

SSPP does not address formal certification of a manufacturer's implementation; this is a SCM manufacturer's responsibility.

B.4 Installation and commissioning (Clause 7)

SSPP provides all functional capability needed to properly install and commission SCMs deployed in an operating environment.

B.5 Performance constraints (Clause 8)

SSPP requirements for holdback address all performance constraint in the context of minimizing the impact on functional communication network operation and performance.

Annex C SCM implementation requirements and issues

The following issues were discovered in the prototype software development, laboratory and field testing AGA 12 compliant cryptographic modules (AGA, March 14, 2006) . These lessons learned should be considered in the design, development and deployment of future IEEE 1711 compliant cryptographic modules.

C.1 Link layer mixed mode operation

The octets that are carried on the remote/ciphertext port fit into one of three categories: SCADA messages, SSPP messages, and modem control. Modem traffic alternates with the other two, based on the current state of the modem. However, a SCM needs to be able autonomously communicate both SSPP and SCADA messages to permit mixed mode operation. A SCM with this capability must be listening for two different types of messages simultaneously. As noted in consideration for selecting link layer markers, the SCM link layer markers are selected properly to avoid confusion with the SCADA traffic. In the prototype development software, this was implemented with a stack of processors, one processor for each SCADA protocol in use (usually only 1), one for ciphertext, and possibly one for modem pass-through commands. The incoming characters from the ciphertext port are input to the first processor. This processor can buffer characters until deciding that the message belongs to it, or deciding that the message is NOT for it, in which case it forwards any buffered characters down to the next link layer processor. When a link layer processor decides that a message belongs or might belong to it, it continues processing until it recognizes the end of the message, or decides to forward it to the next processor.

C.2 SCM hardware handshaking

A process is needed in which the SCM manages its interaction with and between Data Terminal Equipment (DTE), e.g., a SCADA host, and Data Communications Equipment (DCE), e.g., a radio modem.

Table 8 describes the nine signals (one signal per connector pin) that are used in most serial communications setups. The signal descriptions and directions are adopted for IEEE 1711. It should be noted that the same pins are used, but may have different labelling based on whether the connector is DTE or DCE. For example, the DTE TXD is pin 2, which means that the DCE (or modem) receives on pin 2, i.e., the terminal talks on the line to which the modem listens.

Table 8 Nine common signals on the RS-232 connector

25-Pin	9-Pin	Signal	Direction	Class
1		Protective Ground		
2	3	TXD: Transmit Data	PC to Modem	Data
3	2	RXD: Receive Data	Modem to PC	Data
4	7	RTS: Request to Send	PC to Modem	Control
5	8	CTS: Clear to Send	Modem to PC	Control
6	6	DSR: Data Set Ready	Modem to PC	Control
7	5	(Signal) GND: Ground		
8	1	DCD: Data Carrier Detect	Modem to PC	Control
20	4	DTR: Data Terminal Ready	PC to Modem	Control
22	9	Ring Indicate	Modem to PC	Control

Their frequent use (and the disuse of the others) occurred as early as the Teletype terminals and acoustic modems in the 1960s. The other pins are specified but are used in less common applications, such as synchronous serial communications (i.e., mainframe-mainframe links). These pins became exclusively used for most links as a result of the adoption of the 9-pin connector on the serial/parallel interface card released by IBM in 1984 that provided only these signals.

A simple system in which the host communicates with a CSU/DSU (leased line) potentially permits the use of only three of the lines, TXD, RXD, and GND, under the assumption that the leased line is always on and does not need to signal the state of the line.

A more complex process is involved with modems, especially radio modems. Because of this, the following discussion uses the SCADA radio modem framework. It should also be noted that intermediate devices, like a front-end communications processor, are not discussed here because their function is simply as a proxy to the primary devices that are discussed. Also noted is that this is a representative description developed from various discussions, but that individual manufacturers may develop their own uses for the signals.

There are a minimum of two variations in the management of the host-side radio modem. One is that it is treated in the same fashion as the field radios, i.e., turned on when needed. The other variation assumes that the host radio is always on. This process works only if the host-side transmitter is on one frequency (to which all field receivers are tuned), and the field transmitters are on another frequency (to which the host receiver is tuned).

Utilization of radio modems is similar to a multi-drop wired configuration, i.e., all the field units listen to the host sending a message. The field unit that was addressed by the host message develops its reply, and then transmits the response so the host radio can receive it.

In operation, the SCADA master station formulates a command, and then sends it to the radio, which transmits it. The transmitter can be always on, ready to send, because no one else is utilizing that frequency. The field device receivers pick up the message, which is sent to each of the SCADA units. The one to whom it was addressed acts on the message, while the others ignore it. When the field device sends its reply back, it needs to activate the radio transmitter before sending, then turning off the transmitter after the message has been sent. Leaving the transmitter on would jam any signal from another field device unit.

The SCADA master station can maintain its signal lines in the state that the field devices uses to transmit its message, so the details of the process will focus on the field device and its interaction with the radio modem.

When both units are powered up and ready to communicate (e.g., serial port communication parameters are set, such as speed, parity, and word length), each asserts its respective line. The host signals to the modem that it is ready with the DTR signal. The modem signals that it is ready with the DSR signal.

DCD (sometimes referred to as CD/Carrier Detect) is used to signal the DTE that the DCE is detecting a carrier signal. For radio modems, this indicates that the receiver is detecting the carrier that is being transmitted by another radio modem. Unlike phone line modems, the lack of a received carrier does not necessarily mean that the connection is broken, just that the modem at the "other end" is not getting ready to transmit. As a result, this signal may or may not be monitored by the DTE host or field device.

When the field device is ready to transmit its message, it first asserts RTS to signal to the modem. When the radio modem receives the RTS, it can activate its transmitter, and then assert CTS to signal the field unit that it can begin sending the message. The field device then sends its message, which the radio modem transmits. The field unit then completes the process by releasing RTS. The radio modem sees the RTS change, releases CTS in response, and then turns off its transmitter.

Some radio modems purportedly operating in this manner, assert CTS before the transmitter has reached a stable operating point. As a result, the hardware communicating with the radio modem needs to delay for some period of time after seeing the CTS signal before transmitting, to ensure that the data being sent is not garbled.

At the other end, the radio modem can assert DCD indicating that it is receiving a signal from a field radio modem, then receives the message and passes it on to the SCADA host. When the field radio modem stops its transmitter, the host radio modem can release DCD. The host can be set up to use DCD to signal that it is about to receive something, or can ignore it.

When the SCM is placed between a SCADA unit and its radio modem, it needs to appear to the SCADA unit as a modem/DCE, and needs to appear to the radio modem as a host/DTE or field unit/DTE. This means that the SCM reflects the changes in the signal lines between the two devices, with some additional constraints on timing. It appears as follows:

SCADA field unit / DTE [—————] DCE / SCM / DTE [—————] DCE / radio modem

Referring to the case of the field device that is ready to transmit, if these lines are used, the SCM sees the DTR from the field device and the DSR from the radio modem. To complete the connection the SCM needs to assert its DSR on the DCE side (appearing as the modem to the field device) and the SCM needs to assert its DTR on the DTE side (appearing as the field device to the modem).

When in operation, the SCM monitors the RTS line from the field unit, instead of the modem. The field device asserts its RTS line as before. When the SCM sees the asserted RTS, the SCM asserts its RTS line so that the radio modem knows that it is about to receive a message that should be transmitted. When the radio modem asserts its CTS to indicate that the DTE can send, the SCM receives this signal. The SCM needs to relay it to the field unit by asserting its CTS line, observed by the SCADA field device. The device then transmits its SCADA message, and then releases its RTS as before.

Note, that it is also possible for the SCM to assert RTS to the modem after it is ready to send, some time after the field device asserts its RTS. This would also need other changes in timing. The approach described in the previous paragraph permits the field unit to recognize that the radio modem is operating. If this did not occur, the SCM could hide the fact that the radio modem was down, because the SCM is pretending to be the modem to the field unit and everything is working on that side.

Because of the additional processing (SCM message characters, encryption, and message authentication), when the field device releases its RTS, the SCM will be in the middle of transferring the encrypted message to the radio modem. If it were to simply mirror the state of the SCADA master station signals, dropping RTS would cause the radio modem to turn off its transmitter and the last portion of the message would be cut off. At this point, the SCM keeps its RTS asserted until it is done transferring the message through the radio modem. The SCM can then release RTS (go back to reflecting the host's signal lines) so that the radio modem can deactivate its transmitter and release its CTS.

It is probable that the SCM reflecting the CTS line from the radio modem would work in this scenario. If the field device is asserted for longer than normal time, it should not interfere with anything because it is usually only monitored when the field device is preparing to send a SCADA message. However, the SCM could act more closely to the radio modem's actions, i.e., releasing CTS when it sees the SCADA master station RTS released.

C.3 Dialup modem interaction

When modems are used in support of SCADA operations, the host software addresses the modem via the serial channel to cause it to dial a specified phone number, connect with an answering modem, transfer data, and then disconnect. Modems have almost universally used the AT command set, originally developed by Hayes Computer Products for their Smart modem series (ca. 1982), and expanded by other modem manufacturers. A benefit of this is control over the modem operation and settings is through the serial connection.

To permit control of these modems via the SCADA master or field device, a SCM needs to transfer modem communications between SCADA equipment and modems untouched. Modems start in Command mode, and once connected switch to On-line mode. SCMs can track this by monitoring the return values from modems; usually the modem returns a "CONNECT" message with optional details about the connection signals that the modem has switched to On-line mode. When this occurs, the SCM needs to switch to the appropriate SSPP operation to protect the SCADA traffic.

The SCM also needs to recognize when the SCADA device wants to have the modem hang up. If run exclusively in software, the usual process is to wait 1 second, send "+++", and wait 1 more second, referred to as the Escape Code Sequence. The modem seeing this will "escape" from On-line back to Command mode, at which time it will interpret anything sent to it as a command, such as the command to hang up. To permit this process, the SCM needs to be able to recognize the Escape Code Sequence, and may need to be able to recognize alternate characters, since the modems can be set to use any ASCII character repeated three times as the sequence. Upon recognizing it, the SCM needs to switch back to its own modem communication mode, and then send the Escape Code Sequence. Not recognizing it will cause the SCM to process the Escape Code Sequence as if it were part of the SCADA traffic, encapsulating it and the hang-up command, ensuring that the modem will not disconnect.

An alternative method with hardware handshaking may be used by some systems. The serial Data Terminal Ready signal line is asserted to cause the modem to listen to its local connection in either Command or On-line mode. Once connected with a remote modem, the modem will hang up if the DTR is released. For a SCM to work in this scenario, the SCM needs to forward the DTR status it sees from the SCADA device to the modem, intelligently asserting it even if the field device drops the line if the SCM is in the middle of a longer process such as session establishment.

To cover the widest number of possibilities, a SCM would need to implement both of these processes.

C.4 Timing consideration for shared connections

SCADA systems using leased lines (or utility owned wires) often utilize multiple field devices connected to an individual line in a daisy-chain or bus topology. This permits one communications port at the SCADA master station to connect to a series of field devices. The line drivers that link the field devices to the line usually use either two or four wires. The following describes what has been observed with these types of connections.

In general, these systems operate in a fashion similar to RS-485. When four wires are used, one pair is set for messages from the SCADA master to the field device, and the other is used for messages from the field device to the SCADA master. When two wires are used, the single pair is used for messages both ways.

In the two wire case, each line driver must switch from a listener to a talker when needing to communicate, and must recognize when there is no more data to return to being a listener. One method at causing the transition is to implement a timing delay in the line driver, waiting for a silent period to ensure that the outgoing message is finished before returning to the listening mode. One effect of this is that a talker may not be ready to listen as soon as another unit is ready to talk. This situation has been observed on RS-485 lines. As a result, the new listener can miss one or more characters at the start of the message. A SCM designed to operate in this situation should have a user-settable delay to cause it to wait after receiving a message before attempting to send one. In the RS-485 case, it was observed that resetting the RS-232/485 converters to switch faster eliminated the missing characters. Putting in a response delay in the SCM would have achieved the same overall result. Note, that in the four wire case, there is no switching between talker and listener modes, so this problem would not be encountered.

In either the two-wire or four-wire case, it is possible that hardware handshaking is used in a fashion similar to that with radio modems. For example, the Request To Send (RTS) line may be asserted by the field device to cause the line driver to switch to the talker mode. The line driver may use the Clear To Send (CTS) line to signal the device it is ready to receive the data that should be communicated. Or, the field device may just simply delay for a period to allow the line driver to be in talker mode before attempting to send data. In these cases, it would be desirable for a SCM to be configured to assert the signal lines as necessary, to delay as necessary, or both. The SCM would also need to reflect the status of the line driver back to the SCADA device for its proper operation. The SCM would also need to intelligently communicate with the line driver, asserting RTS after the field device has released it if the SCM is still sending.

Note, that simply asserting the signal lines with such a configuration will cause all the line drivers to switch to talk mode, allowing no one to communicate.

C.5 Design consideration for cipher suites

Design of a cipher suite with no holdback (i.e., relying on another mechanism such as the SCADA CRC to determine the legitimacy of the message) is significantly difficult, and should be attempted only after careful study by cryptographic experts. For example, a number of attempts at providing SCADA encryption concurrent with the AGA 12 compliant field test configurations and review by the US national laboratories were subsequently shown to have easily circumvented security.

C.6 Side channel design considerations

Implementation of SSPP shall take into account the potential for side-channel attacks; e.g., determining critical security parameters from noting differences in timing. Anyone implementing an IEEE 1711 compliant cryptographic module should be conversant with the available literature on the identified and suggested attacks, to be able to include countermeasures in the design of the module.

C.7 Session clock considerations

There is the potential for use of high-resolution clocks in attacking AES implementations in cryptographic modules. To address this, it is suggested that cryptographic modules permit the use of resolutions that are in agreement with the needs of the session-relative clocks, and do not

permit use of resolutions significantly higher. An example of an excessive clock resolution is one that is an order of magnitude greater than the bit rate of the communications channel.

C.8 Multi-thread, multi-process and other design considerations

The design of a cryptographic module shall take into account the potential for conflicts within the various functions within the module. In addition to the requirements of FIPS 140-2, good programming practice should account for the potential of problems such as failed memory allocations and missing or garbled messages between modules, and the proper handling of shared resources between multiple threads or processes.