

NIST GCR 02-837

Secure Integration of Building Networks into the Global Internet

Dr. John Zachary
Dr. Richard Brooks
David Thompson
The Pennsylvania State University

NIST

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NIST GCR 02-837

Secure Integration of Building Networks into the Global Internet

Prepared for
*Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8600*

By
Dr. John Zachary
Dr. Richard Brooks
David Thompson

*Applied Research Laboratory
The Pennsylvania State University
State College, PA 16804-0030*

Contract 1W1702

October 2002



U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Contents

1	Secure Integration of Building Networks into the Global Internet	3
1.1	Introduction	3
1.2	BACnet Broadcast Management Devices (BBMDs)	3
1.3	Cryptographic Considerations in ASHRAE-135, Clause 24	4
1.4	Summary	4
2	Protection Profile for BACnet Broadcast Management Devices (BBMDs)	6
2.1	Chapter Overview	6
2.1.1	Conventions and Terminology	6
2.1.2	Document Organization	7
2.2	Introduction	7
2.2.1	PP Identification	7
2.2.2	PP Overview	7
2.2.3	Related PPs	7
2.3	TOE Description	8
2.3.1	TOE Functionality	8
2.3.2	Desired BBMD Security Capabilities	9
2.4	TOE Security Environment	9
2.4.1	Secure Usage Assumptions	10
2.4.2	Security Threats Description	10
2.4.3	Organizational Security Policies	11
2.5	Security Objectives	11
2.5.1	Security Objectives for the TOE	11
2.5.2	Security Objectives for the Environment	12
2.6	IT Security Requirements	13
2.6.1	TOE Security Functional Requirements	13
2.6.2	TOE Security Assurance Requirements	17
2.7	Rationale	24
2.7.1	Security Objective Rationale	24
2.7.2	Dependency of Requirements	29
2.7.3	Rationale for Evaluation Assurance Level 2	29
3	Addendum to Clause 24 of the BACnet Standard	32
3.1	Introduction	32
3.2	Symmetric Key Size	32
3.3	AES Symmetric Block Cipher	33
3.4	BACnet Security Protocol Analysis	34
3.4.1	Assumptions and Attacks	34
3.4.2	Freshness of SK_{AB}	35
3.5	Considerations Regarding Public-Key Cryptography	35
3.6	Conclusion	36
A	Common Security Protocols	37

Chapter 1

Secure Integration of Building Networks into the Global Internet

1.1 Introduction

Building automation and control systems (BACSs) have become increasingly innovative since the introduction of the open ANSI/ASHRAE Standard 135-1995 *BACnet - A Data Communication Protocol for Building Automation and Control Networks* standard [1]. The integration of BACSs with enterprise information technology and networking infrastructure will facilitate new customer services that increase efficiency and decrease costs. The ASHRAE Research Project 1011-1999 *Utility/Energy Management and Control System (EMCS) Communication Protocol Requirement* [7] describes nine operational information services currently considered by various working groups: (1) revenue meter reading (electricity, gas, water, heating, and cooling energy); (2) quality of service monitoring; (3) real-time pricing transmission; (4) load management services; (5) on-site generation supervisory control; (6) energy efficiency monitoring; (7) weather reporting and forecasting services; (8) indoor air quality monitoring; and (9) dynamic demand bidding into a power exchange. Each service generates and exchanges information with varying effective lifetimes, bandwidth, and security requirements.

This research project sponsored by the National Institutes of Standards and Technology (NIST) and performed at the Pennsylvania State University focused on the security aspects of these proposed operational services. We focused on two specific aspects of the BACnet standard, namely BACnet Broadcast Management Devices (BBMDs) and the Network Security clause (Clause 24) of the ANSI/ASHRAE Standard 135-1999.

1.2 BACnet Broadcast Management Devices (BBMDs)

The Common Criteria (CC) provides a uniform basis for evaluating security properties of information technology products. BBMDs are a prime target for development of a Protection Profile (PP) since they are an integral part of the BACnet networking infrastructure and they directly interact with non-BACnet devices. The BBMD PP is compatible with the following independently developed PPs (publicly available at www.iatff.net):

- *Protection Profile for Switches and Routers*,
- *Traffic Filtering Firewall for Low Risk Environments (Basic)* and
- *Traffic Filtering Firewall for Medium Robustness*

The BBMD PP supports an Evaluation Assurance Level (EAL) of 2 based on the general level of vulnerabilities associated with broadcast/multicast messaging in networked systems, the ease with which foreign devices are accepted by BBMDs into a BACnet system, and the trend by industrial developers to construct devices with multiple functionality integrated into a single package. Furthermore, the EAL2 level is augmented

with three additional assurance requirements to stress that formal testing and maintenance are necessary to protect against vulnerabilities in design, engineering, and maintenance (particular in multifunction “BBMD” devices).

1.3 Cryptographic Considerations in ASHRAE-135, Clause 24

This research focuses on *Clause 24, Network Security* of ANSI/ASHRAE Standard 135-1999 and results in an addendum to *Clause 24*. The topics addressed are:

- symmetric key length,
- the Advanced Encryption Standard (AES) symmetric block cipher, and
- a formal analysis of the session key distribution protocol in *Clause 24*.

The addendum includes a brief discussion on public-key cryptography and the impact on key lengths in BACnet systems.

Our analysis demonstrates that the current specification for a 56-bit key length in *Clause 24* is inadequate to guarantee confidential information exchange between BACnet peer devices, especially for information with a lifetime that exceed 12-24 hours. The reason is that 56-bit DES has been demonstrated to be breakable by exhaustive key search in around 20 hours. A minimum key length of 128 bits will ensure data confidentiality for a BACnet system, particularly when considered in the context of the discussion on AES.

The AES symmetric block cipher was recently deemed the new Federal standard symmetric block cipher [10]. It is meant to replace the venerable but insecure Data Encrypt Standard (DES) [9]. The replacement of DES with AES is relatively straightforward for many systems. The main differences between DES and AES are that the latter block cipher method permits three different key bit lengths (128, 192, and 256) and operates on 128 bit blocks of data (compared to 64-bit blocks in DES). Implementations of AES offer better performance, higher efficiency, and smaller memory requirements for both encryption and key scheduling than do implementations of DES. It is expected that AES will quickly replace DES in most systems, with DES used to handle legacy encrypted information.

Network security in BACnet is based entirely on symmetric block ciphers. As such, it depends on a third party key server to distribute session keys for each peer device that engages in confidential data exchange or authentication. The protocol used to initiate and distribute session keys is a general target for attackers. If an attacker is able to subvert a key distribution protocol, then they are able to learn session key values. Hence, they are able to decipher ciphertext passed between peers who assume (wrongly) that their communication channel is secure. Attackers have several attack types at their disposal, including

- Man-in-the-middle
- Type flaws
- Parallel interleaving attacks, and
- Replay (freshness) attacks

We explain that the *Clause 24* protocol for session key distribution potentially suffers from a vulnerability to replay (freshness) attacks. This vulnerability is shared by many protocols, including the Needham-Schroeder Secret Key protocol. We explain how the storage of session keys in each device’s `Device` object should include a timestamp or other time-sensitive value to indicate either when the session key was accepted or when the session key is expected to expire. An alternative is to coordinate with the trusted key server upon an initial request by a peer device to establish a secure connection for confidential data transmission.

1.4 Summary

The project investigated two aspects of BACnet system security not normally covered in the general security literature. The first contribution is a Protection Profile of BBMDs based on the *Common Criteria 2.1*

specification. The PP will ensure that BBMDs are engineered and deployed in a manner to protect broadcast information in BACnet systems. The second contribution is a careful analysis of cryptographic elements in *Clause 24, Network Security* in ANSI/ASHRAE Standard 135-1999. We covered three cryptographic topics: cryptographic key length, the AES symmetric block cipher, and the protocol specified in *Clause 24, Network Security* for establishing session keys between devices.

Chapter 2

Protection Profile for BACnet Broadcast Management Devices (BBMDs)

2.1 Chapter Overview

This Protection Profile (PP) for BACnet Broadcast Management Devices (BBMDs) was developed for the National Institutes of Standards and Technology (NIST). It was prepared by the Pennsylvania State University Applied Research Laboratory (PSU ARL) and the Pennsylvania State University Facilities Engineering Institute (PSU FEI).

The PP is the first formal submission by PSU ARL and PSU FEI to NIST on this particular concept of security standards as related to BBMDs. This PP is intended for public and private audiences.

The base set of requirements used in the PP is taken from the *Common Criteria (CC) for Information Technology Security Evaluation*, Version 2.1.

2.1.1 Conventions and Terminology

The notation, formatting, and conventions used in this PP are consistent with the Common Criteria, Version 2.1. The formatting and font style used to present the information is meant to assist the reader.

Boldface items refer to PP items, such as policies, threats, objectives, and requirements.

Italicized items are used to specify technical terms.

Bulleted lists are used throughout the text to organize information.

2.1.2 Document Organization

Section 1 provides introductory material for this PP.

Section 2 provides a description of the TOE and presents the architectural scenarios that represent the operational environment of the TOE.

Section 3 provides a description and assumptions of the TOE Security Environment. This section also defines the threats and policies addressed by the technical implementations of the TOE.

Section 4 defines the Security Objectives for the TOE based on assumptions, threats, and policies.

Section 5 provides Information Technology Security Requirements, including Functional and Assurance Requirements derived from Parts 2 and 3 of the CC.

Section 6 provides a sequence of rationales to demonstrate that the security objectives address the threats and policies. The requirements are shown to be complete with respect to the objectives.

2.2 Introduction

2.2.1 PP Identification

This PP was developed for the National Institute of Standards and Technology (NIST) by the Pennsylvania State University Applied Research Laboratory and the Pennsylvania State University Facilities Engineering Institute. It is developed with the intent to promulgate security in internetworks between BACnet building control system networks and utility providers over IP. Specifically, it focuses on promoting the security of BACnet Broadcast Management Devices (BBMDs) for broadcast and multicast functionality in the aforementioned internetworks. The intended audience is both public and private, including information security engineers, product vendors, system integrators, and product evaluators.

2.2.2 PP Overview

This PP specifies requirements for protecting broadcast information with BBMDs that are compliant with this PP. BBMDs are a component of BACnet networks that are responsible for managing broadcast messaging. Typical scenarios under which BBMDs may be operated are described in *Section 2.2, Usage Scenarios*.

BBMDs are the TOEs defined in this PP. This PP defines the assumptions, threats, and organizational policies addressed by BBMDs. It also defines implementation-independent security objectives, functional requirements, and assurance requirements. Finally, this PP provides rationale for the proposed security objectives and requirements.

2.2.3 Related PPs

This PP was written to be compatible with the following PPs:

- *Protection Profile for Switches and Routers*
- *Traffic Filtering Firewall for Low Risk Environments (Basic)*
- *Traffic Filtering Firewall for Medium Robustness*

These PPs are available from the WWW at <http://www.iafff.net>.

2.3 TOE Description

The TOE described in this document is the BACnet Broadcast Management Device (BBMD). Its purpose is to manage broadcast messages between devices on different subnets.

A BACnet/IP (B/IP) network is a (nonempty) set of IP subnetworks with an assigned BACnet network address. A B/IP device has a unique IP address, and, unless the device is a router, it does not need to know this address. B/IP devices behave like all other BACnet devices. Specifically, they can communicate directly with other peer devices, and they are able to send and receive local and remote broadcast messages.

B/IP uses the connectionless User Datagram Protocol (UDP) instead of the more ubiquitous connection-oriented Transmission Control Program (TCP) protocol. The main reason is the resource constraints associated with connection-oriented protocols. UDP requires less overhead in transmitting packets than TCP, an issue particularly sensitive for networks of embedded control devices. UDP is a well-known protocol with well-supported APIs across most operating systems, routers/switches, and firewalls.

The BACnet Virtual Link Layer (BVLL) provides an interface between the BACnet Network Layer and a physical communications layer.

Broadcast messaging is an important functional component of BACnet. However, simultaneous transmission of packets to multiple destinations is not supported by routing hardware, and the Internet Group Membership Protocol (IGMP) is not appropriate for BACnet. Hence, broadcast messaging is supported in B/IP by the introduction of an ancillary broadcast routing device called a BBMD. In this PP, it is assumed that a BBMD is owned and managed by the organization responsible for the respective IP subnet.

2.3.1 TOE Functionality

A BBMD listens for broadcast messages on a subnet and forwards them to the appropriate destination addresses. BBMD interconnections form fully connected network topologies, which makes them robust to link failures and easy to configure.

The mechanism for establishing interconnections between BBMD devices is straightforward. Each BBMD has a *Broadcast Distribution Table (BDT)* containing entries for peer BBMD B/IP addresses. A triad of messages is used to add new BBMD devices to a BACnet network and update the existing set of BBMD devices. The BVLL *Write-Broadcast-Distribution-Table* message provides the mechanism for initializing and updating a BBMD BDT. This message contains a list of BDT entries, where each entry contains a BBMD B/IP address and a broadcast distribution mask. The mask indicates how the BBMD distributes broadcast messages on its IP subnet.

There are two corresponding query messages. An initiating BBMD issues a *Read-Broadcast-Distribution-Table* message to a BBMD to request a copy of the recipient's BBMD BDT entries. The recipient responds with a *Read-Broadcast-Distribution-Table-Ack* message that contains a list of the BBMD BDT entries.

Broadcast Operation

Broadcast messages sent within a single IP subnet are not routed through BBMDs since the messages are automatically forwarded to nodes within the subnet IP using the B/IP broadcast address. In the case of sending broadcast messages across multiple IP subnets, the B/IP broadcast address is insufficient because IP routers do not automatically forward broadcast messages across different IP subnets. This deficiency is the primary role of the BBMD. This BACnet specific device can be thought of as a broadcast router.

There are two mechanisms that BBMDs can use to route broadcast messages over multiple IP subnets. The first is called "one-hop distribution" and is a form of directed broadcasting. The network address of the broadcast message contains the subnet address of the target IP subnet. The host address portion is filled with 1's. This method is quite efficient, but it requires a specific configuration the router serving the target

IP subnet.

An alternative mechanism is a “two-hop distribution” method. The BBMD of the source IP subnet sends a directed message to the BBMD of the target IP subnet. The target BBMD then uses the B/IP broadcast address to broadcast the message to all nodes within its IP subnet. This approach is easier to configure since it does not rely on an IP router configuration. Either case is susceptible to broadcast message attacks, but since the “two-hop distribution” method is simpler in terms of configuration, it is easier to analyze for vulnerabilities.

Foreign devices

The B/IP standard allows foreign BACnet devices to register with BACnet networks. Foreign devices can be permanent nodes on foreign subnets (e.g. monitor workstations) or transient nodes (e.g. workstations connected via PPP) and can send and receive broadcast messages.

Foreign devices register with a BBMD serving an IP subnet of the BACnet network. The mechanism by which the foreign device chooses a BBMD is a local matter (and a possible security issue). A BVLL *Register-Foreign-Device* message is sent to the target BBMD by the foreign device. Each BBMD maintains a *Foreign-Device-Table* where each entry contains the B/IP address of a foreign device and a timeout value after which the foreign device will be purged from the table if it does not re-register.

Integrated BBMD Devices in the Marketplace

The scope of this document is the functionality of BBMDs as defined in *BACnet Standard Annex J – BACnet/IP*. Current marketplace efforts result in devices that incorporate multiple functionality, including that of BBMDs, routing, and servicing of WWW standards. A clear delineation of services between modules of these hybrid devices is critical. Other service modules of devices typically marketed as “BBMDs” by industrial producers is not included in this document, but these services may be covered by other PPs.

2.3.2 Desired BBMD Security Capabilities

The set of desirable security capabilities of BBMDs include:

- Detection and resistance to attacks against the TOE that might result in system degradation or denial of service.
- Survivability and recovery of TOE functionality after an attack attempt or interruption in service.
- Robustness to replay attacks against the TOE.
- Resistance to attacks against information integrity.
- Resistance to attacks against information confidentiality.
- Auditing capabilities of all network activities, including TOE management.
- Authentication of peer nodes when establishing access.
- Authentication of foreign devices when establishing access.
- Resistance to foreign devices masquerading as valid TOE or BACnet devices; i.e., devices following the BACnet CC PP provided should contain functionality intended to make it difficult for non-BACnet devices (or devices that are not the device being evaluated) to make network transactions as if they were BACnet devices (or the device itself).

2.4 TOE Security Environment

This section covers the usage assumptions, threats, and policies of the TOE.

2.4.1 Secure Usage Assumptions

The conditions assumed to exist in the TOE operational environment include:

- **A.DESIGN** – The design, manufacturing, and delivery of the TOE will operate within specification limits and will comply with security requirements
- **A.DIRECT** – Human users within the physically secure boundary protecting the TOE may attempt to access the TOE from some direct connection, such as a console port, if the connection is part of the TOE.
- **A.NOGENPURPOSE** – Neither general purpose computing capabilities, such as the ability to execute arbitrary code or applications, nor storage repository capabilities exist on the TOE.
- **A.LOWEXPLOIT** – The threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
- **A.NOEVIL** – Authorized administrators are non-hostile and follow all administrator guidance; however, they are capable of error.
- **A.PEER** – Systems with which the TOE shares restricted data are assumed to operate under the same security policy constraints and have been accredited to be compliant with the proposed rule.
- **A.PHYSEC** – The TOE is located in a physically secure facility that prevents unauthorized physical access by outsiders. The TOE is located to prevent casual contact by employees.
- **A.PUBLIC** – The TOE does not host public data.
- **A.REMACCESS** – Authorized administrators may access the TOE remotely from the internal and external networks.
- **A.SECRSP** – Responsibility for ensuring that appropriate security measures are in place (including risk analysis, risk management, and development and maintenance of the security policy) and are complied with is assumed to be assigned to a specific individual or organization.
- **A.USERTRANSP** – Application of the TOE to user traffic is transparent to the users.

2.4.2 Security Threats Description

The possible threats to a PP-compliant TOE include:

- **T.ADMINERR** – An authorized administrator performs erroneous actions that compromise user and/or system resources.
- **T.ANALYSIS** – An attacker collects source and destination addresses, volume of data, and time of day that message are sent.
- **T.CAPTURE** – An attacker eavesdrops, taps into the transmission line, or otherwise captures data being transferred on a communications channel.
- **T.COMPNODE** – A node becomes compromised, altering the TOE configuration files and/or the routing behavior, causing incorrect operations of the TOE, disabling of TOE security functions, and/or re-routing of traffic, possibly through an unauthorized node.
- **T.DENIAL** – An attacker executes a command, sends excess high priority traffic, or performs other operations that cause undue burden on the network, thereby making system resources unavailable to authorized clients.
- **T.FAIL** – Failure of one or more system components or a power failure results in the loss of system-critical functionality and data.

- **T.FLAW** – Flaws in hardware, software, or firmware result in incorrect functionality, possibly leading to security-related problems.
- **T.MODIFY** – An attacker makes unauthorized modifications or manipulates protocols en-route.
- **T.NETMAP** – An attacker attempts to map the network of which the TOE is a member, thus obtaining addresses of nodes, routing information, and physical locations.
- **T.SPOOF** – An attacker masquerades as an authorized node or administrator in order to obtain access to TOE resources, possibly by using a valid network address.

2.4.3 Organizational Security Policies

Organizational security policies that support the security objectives presented in Section 2.5 include:

- **P.ACCOUNTABILITY** – Organizations that use the TOE for transmitting information, persons in network management roles, and developers shall be held accountable for their actions.
- **P.AUTHENTICATION** – The TOE shall support authentication of network operators, administrators, and peer nodes.
- **P.AVAILABILITY** – The TOE will provide timely and reliable access to information and system resources to meet operational requirements.
- **P.DATA** – The network will be configured to route all applicable data through the TOE to ensure that the TSP will be met.
- **P.DEFAULTCONF** – The default TOE configuration will have all functions that weaken or break TOE security disabled. TOE functions that enhance TOE security will be enabled by default.
- **P.INSTALL** – Documentation for the secure installation, configuration, and maintenance of the BBMD will be provided with the BBMD.
- **P.INTEGRITY** – The TOE will provide controls to protect the integrity of information and system resources, including maintaining the integrity of the TSF in the event of a system failure (e.g, fail-safe mode).
- **P.INTEROPERABILITY** – The TOE shall be interoperable with BBMDs manufactured by other vendors according to the BACnet specification. Standardized and nonproprietary protocols that adhere to the BACnet standard shall be implemented in the TOE. Proprietary protocols may be implemented to the extent that they do not interfere with the TOE standardized protocols.
- **P.NOTIFY** – The TOE and the TSE will be capable of generating alerts in the event of a component, firmware, hardware, or software failure.
- **P.PEER** – Secure nodes will have the ability to accept traffic from trusted and untrusted nodes. Traffic may be filtered between trusted and untrusted nodes to protect information.
- **P.SURVIVE** – The TOE shall be able to recover from hostile attempts against security. The TOE shall be able to recover from errors that may occur during transmission. The network must be resistant or able to recover in a reasonable time period from hardware, software, or firmware failures. Partial recoveries shall be documentable.

2.5 Security Objectives

2.5.1 Security Objectives for the TOE

The following are IT security objectives for the TOE:

- **O.ACCESSCONTROL** – Implement an access control policy based on but not limited to the following: the TOE’s role, TOE identity, source and destination addresses, filtering at the port level, acceptance and authentication of foreign devices, etc.
- **O.ALARM** – The TOE will be capable of detecting any failure or error with any component, hardware, software, or firmware. The TOE will be capable of generating an alarm for security related events and failure.
- **O.CONFINTEGRITY** – All information pertaining to the configuration and operation of the TOE will retain content integrity. The TOE may not be responsible for storing this information.
- **O.CONFMANAGE** – A management plan for capturing and retaining configuration and connection information for each BBMD will be developed and implemented. The plan should ensure storage integrity, system connectivity, and identification of system components.
- **O.CTRLAUTH** – Connectivity will be provided between peer TOEs only upon the identification, authentication, and authorization of the source and destination addresses in accordance with the access control policy.
- **O.DETECT** – The TOE will be able to discriminate between authorized and unauthorized connections.
- **O.LIFECYCLE** – The TOE will be managed and maintained such that its security functions are implemented and preserved across its operational lifecycle. Hardware, software, and/or firmware upgrades will preserve or enhance current security features without negatively affecting any other TSF.
- **O.PATCHES** – The TOE will have the latest patches and security fixes installed.
- **O.PROTECTADDR** – The TOE will protect the confidentiality and integrity of the source and destination entity address in accordance with the access control policy. The TOE will correctly interpret both source and destination addresses.
- **O.PROTOCOLS** – Standardized protocols will be implemented in the TOE to achieve interoperability with BBMDs and other BACnet devices from other vendors. Protocols will enable reliable transport and error detection.
- **O.TEST** – Test plans and procedures will be documented and followed to test the TOE and the TSF, including vulnerability testing to determine if and how the TOE security policy might be violated.
- **O.TRUSTEDRECOVERY** – Ensure the recovery to a secure state without security compromise after discontinuity of operations.
- **O.UNUSEDFIELDS** – Unused field values will be set properly to ensure correct and secure TOE operation.
- **O.VALIDATION** – Ensure the integrity and correct installation of all hardware, software, and/or firmware.

2.5.2 Security Objectives for the Environment

The following security objectives are achieved largely through application of procedural or administrative measures, and not through implementation of functions in the TOE hardware, software, and/or firmware:

- **OE.ACCESSMGMT** – Network administrators will manage the access control policy within the network management system to grant authorized network management personnel the necessary functional abilities. Network management personnel can assume their privileged role(s) within the network management system only upon proper identification and authentication.

- **OE.DOCUMENTATION** – Minimize installation, configuration, and operating errors by providing guidance documentation covering these aspects on a timely basis. This documentation will assist personnel in the maintenance of the TOE and its security functions.
- **OE.ENVIRONMENT** – Resources shall be developed to provide protection against environmental threats, such as fire, earthquake, and power loss.
- **OE.PERSONNEL** – Trustworthy and competent personnel will be hired. Possible procedures that may be used to ensure these characteristics include security lectures, monitoring, misuse detection, and testing.
- **OE.PHYSEC** – Resources shall be physically protected to prevent malicious attacks, unauthorized modification, destruction, and theft.

2.6 IT Security Requirements

The functional and assurance requirements provided in this section must be satisfied by a PP-compliant TOE. The components of the functional requirements are taken from Part 2 of the CC. The assurance components are taken from Part 3 of the CC.

2.6.1 TOE Security Functional Requirements

Class FAU: Security Audit

- **FAU_GEN.1: Audit Data Generation**

The TSF shall be able to generate an audit record of the following auditable events: (a) start-up and shutdown of the audit functions; and (b) all auditable events for the *basic* level of audit (*FAU_GEN.1.1*).

The TSF shall record within each audit record at least the following information: date and time of the event, type of event, subject identity, and the outcome (success or failure) or the event (*FAU_GEN.1.2*).

Dependencies:

- **FPT_STM.1: Reliable Time Stamps**

- **FAU_SAR.1: Audit Review**

The TSF shall provide network administrators with the capability to read the list of all audit data from the audit records (*FAU_SAR.1.1*).

The TSF shall provide the audit records in a manner suitable for the user to interpret the information (*FAU_SAR.1.2*).

Dependencies:

- **FAU_GEN.1: Audit Data Generation**

- **FAU_SEL.1: Selective Audit**

The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes: object identity, host identity, and event type (*FAU_SEL.1.1*).

Dependencies:

- **FAU_GEN.1: Audit Data Generation**
- **FMT_MTD.1: Management of TSF Data**

Class FDP: User Data Protection

- **FDP_ACC.1: Subset Access Control**

The TSF shall enforce the access control policy on communication requests. (*FDP_ACC.1.1*).

Dependencies:

- **FDP_ACF.1: Security Attribute Based Access Control**

- **FDP_ETC.1: Export of User Data without Security Attributes**

The TSF shall enforce the access control policy when exporting user data, controlled under the SFP, outside the TSC (*FDP_ETC.1.1*).

The TSF shall export the user data without the user data's associated security attributes (*FDP_ETC.1.2*).

Dependencies:

- **FDP_ACC.1: Subset Access Control**
- **FDP_IFC.1: Subset Information Flow Control**

- **FDP_IFC.1: Subset Information Flow Control**

The TSF shall enforce the information flow control SFP on control information received from foreign devices (*FDP_IFC.1.1*).

Dependencies:

- **FDP_IFF.1: Simple Security Attributes**

- **FDP_IFF.1: Simple Security Attributes**

The TSF shall enforce the information flow control SFP based on the following types of subject and information security attributes: (a) access control policy; (b) the state of the source of control information i.e. trusted or untrusted (a trusted source is identified, authenticated, and verified); and (c) identity of the originating source of the message (*FDP_IFF.1.1*).

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: the source of the message is able to be identified and authenticated by the recipient. (*FDP_IFF.1.2*).

The TSF shall enforce the information flow control policy (*FDP_IFF.1.3*).

The TSF shall provide the following: the ability to configure the TSF to implement a policy decision for accepting data from an untrusted source (foreign device), the ability to auditing the receipt of data from an untrusted source (foreign device), and the ability to accept data from an untrusted source (foreign device) (*FDP_IFF.1.4*).

The TSF shall explicitly authorize an information flow only if management information through a trusted path to the network management station(*FDP_IFF.1.5*).

The TSF shall explicitly deny an information flow in accordance with the security policy setting denying receipt of data and/or control information from untrusted sources (*FDP_IFF.1.6*).

Dependencies:

- **FDP_IFC.1: Subset Information Flow Control**
- **FMT_MSA.3: Static Attribute Initialization**

- **FDP_ITC.1: Import of User Data without Security Attributes**

The TSF shall enforce the access control policy and/or information flow control policy when importing user data, controlled under the SFP, from outside the TSC (*FDP_ITC.1.1*).

The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC (*FDP_ITC.1.2*).

Dependencies:

– **FMT_MSA.3: Static Attribute Initialization**

• **FDP_UIT.1: Data Exchange Integrity**

The TSF shall enforce the access control policy and/or information flow control policy to be able to transmit and receive user data in a manner protected from modification, deletion, insertion, and replay errors (*FDP_UIT.1.1*).

The TSF shall be able to determine on receipt of user data whether modification, deletion, insertion, or replay has occurred (*FDP_UIT.1.2*).

Dependencies:

– **FDP_ACC.1: Subset Access Control**

Class FIA: Identification and Authentication

• **FIA_UAU.2: User Authentication Before Any Action**

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user (*FIA_UAU.2.1*).

Dependencies:

– **FIA_UID.1: Timing of Identification**

• **FIA_UID.2: User Identification Before Any Action**

The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user (*FIA_UID.2.1*).

Dependencies: None

Class FMT: Security Management

• **FMT_MOF.1: Management of Security Functions Behavior**

The TSF shall restrict the ability to: determine the behavior of, disable, enable, or modify the behavior of the functions of: (a) the TOE at the time of installation and throughout its lifecycle; (b) TOE security fixes/patches; and (c) TOE configuration and maintenance (*FMT_MOF.1.1*).

Dependencies:

– **FMT_SMR.1: Security Roles**

• **FMT_MSA.1: Management of Security Attributes**

The TSF shall enforce the access control policy and/or information control policy to restrict the ability to change the default settings, create, modify, and delete the security attributes of the following: selecting auditable events, managing audit logs, system access control lists and accounts, and user access control lists and accounts to network administrators (*FMT_MSA.1.1*).

Dependencies:

– **FDP_ACC.1: Subset Access Control**

• **FMT_MSA.3: Static Attribute Initialization**

The TSF shall enforce the access control policy and/or information flow control policy to provide restrictive default values for security attributes that are used to enforce the SFP (*FMT_MSA.3.1*).

The TSF shall allow the network administrator to specify alternative initial values to override the default values when an object or information is created (*FMT_MSA.3.2*).

Dependencies:

– **FMT_MSA.1: Management of Security Attributes**

– **FMT_SMR.1: Security Roles**

• **FMT_MTD.1: Management of TSF Data**

The TSF shall restrict the ability to change defaults as well as the ability to query the TOE audit data to network administrators (*FMT_MTD.1.1*).

Dependencies:

– **FMT_SMR.1: Security Roles**

Class FPT: Protection of the TSF

• **FPT_AMT.1: Abstract Machine Testing**

The TSF shall run a suite of tests during initial start-up and at the request of authorized network administrators to demonstrate the correct operation of the security assumptions provided by the abstract machine underlying the TSF (*FPT_AMT.1.1*).

Dependencies: None

• **FPT_FLS.1: Failure with Preservation of State**

The TSF shall preserve a secure state when the following types of failures occur: hardware component failure, short-term power interruptions (*FPT_FLS.1.1*).

Dependencies:

– **ADV_SPM.1: Informal TOE Security Policy Model**

• **FPT_PHP.1 Passive Detection of Physical Attack**

The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF (*FPT_PHP.1.1*).

The TSF shall provide the capability to determine whether physical has occurred (*FPT_PHP.1.2*).

Dependencies:

– **FMT_MOF.1: Management of Security Functions Behavior**

• **FPT_STM.1: Reliable Time Stamps**

The TSF shall be able to provide reliable time stamps for its own use (*FPT_STM.1.1*).

Dependencies: None

• **FPT_TDC.1: Inter-TSF Based TSF Data Consistency**

The TSF shall provide the capability to consistently interpret access, audit, control, and security parameter information when shared between itself and another trusted IT product (*FPT_TDC.1.1*).

The TSF shall use authenticated protocols (specified by the TOE developer) when interpreting the TSF data from another trusted IT product (*FPT_TDC.1.2*).

Dependencies: None

• **FPT_TST.1: TSF Testing**

The TSF shall run a suite of self tests during initial start-up, at the request of an authorized network administrator, and after a short-term power interruption to demonstrate correct operations (*FPT_TST.1.1*).

The TSF shall provide authorized users with the capability to verify the integrity of TSF data (*FPT_TST.1.2*).

The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code (*FPT_TST.1.3*).

Dependencies:

– **FPT_AMT.1: Abstract Machine Testing**

Class FRU: Resource Utilization

- **FRU_FLT.1: Degraded Fault Tolerance**

The TSF shall ensure the operation of backup-components switching, secure information transfers, proper broadcasting of messages, proper internal processing of network traffic, preservation of foreign device tables, and continuing network operations while event auditing when the following failures occur: hardware failure, software error, modification of control and management information, extreme network congestion, and short power interruption (*FRU_FLT.1.1*).

Dependencies:

- **FPT_FLS.1: Failure with Preservation of Secure State**

- **FRU_PRS.1: Limited Priority of Service**

The TSF shall assign a priority to each subject in the TSF (*FRU_PRS.1.1*).

The TSF shall ensure that each access to routing information and foreign device tables shall be mediated on the basis of the subjects assigned priority (*FRU_PRS.1.2*).

Dependencies: None

Class FTA: TOE Access

- **FTA_TSE.1: TOE Session Establishment**

The TSF shall be able to deny session establishment based on identity, authentication data, untrustworthiness of source, roles, address, time of access, or security status (*FTA_TSE.1.1*).

Dependencies: None

Class FTP: Trusted Path and Channels

- **FTP_TRP.1: Trusted Path**

The TSF shall provide a communication path between itself and remote trusted devices that is logically distinct from other communication paths. It shall also provide assured identification of its end points and protection of the communicated data from modification or disclosure (*FTP_TRP.1.1*).

The TSF shall permit the TSF and remote trusted devices to initiate communication via the trusted path (*FTP_TRP.1.2*).

The TSF shall require the use of the trusted path for the transmission of control information and security attributes (*FTP_TRP.1.3*).

Dependencies: None

2.6.2 TOE Security Assurance Requirements

Class ACM: Configuration and Management

- **ACM_CAP.2: Configuration Items**

The developer shall provide a reference for the TOE (*ACM_CAP.2.1D*).

The developer shall use a CM system (*ACM_CAP.2.2D*).

The developer shall provide CM documentation (*ACM_CAP.2.3D*).

The reference for the TOE shall be unique to each version of the TOE (*ACM_CAP.2.1C*).

The TOE shall be labeled with its reference (*ACM_CAP.2.2C*).

The CM documentation shall include a configuration list (*ACM_CAP.2.3C*).

The configuration list shall describe the configuration items that comprise the TOE (*ACM_CAP.2.4C*).

The CM documentation shall describe the method used to uniquely identify the configuration items (*ACM_CAP.2.5C*).

The CM system shall uniquely identify all configuration items (*ACM_CAP.2.6C*).

Dependencies: None.

Class ADO: Delivery and Operation

- **ADO_DEL.1: Delivery Procedures**

The developer shall document procedures for delivery of the TOE or parts of it to the user (*ADO_DEL.1.1D*).

The developer shall use the delivery procedures (*ADO_DEL.1.2D*).

The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site (*ADO_DEL.1.1C*).

Dependencies: None.

- **ADO_IGS.1: Installation, Generation, and Start-Up Procedures**

The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE (*ADO_IGS.1.1D*).

The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE (*ADO_IGS.1.1C*).

Dependencies:

- **AGD_ADM.1: Administrator Guidance**

Class ADV: Development

- **ADV_FSP.2: Fully Defined External Interfaces**

The developer shall provide a functional specification (*ADV_FSP.2.1D*).

The functional specification shall describe the TSF and its external interfaces using an informal style (*ADV_FSP.2.1C*).

The functional specification shall be internally consistent (*ADV_FSP.2.2C*).

The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing **complete** details of **all** effects, exceptions, and error messages (*ADV_FSP.2.3C*).

The functional specification shall completely represent the TSF (*ADV_FSP.2.4C*).

The functional specification shall include rationale that the TSF is completely represented (*ADV_FSP.2.5C*).

Dependencies:

- **ADV_RCR.1: Informal Correspondence Demonstration**

Remarks: Augmented from **ADV_FSP.1: Informal Functional Specification** because the BBMD interface is simple and well-defined.

- **ADV_HLD.1: Descriptive High-Level Design**

The developer shall provide the high-level design of the TSF (*ADV_HLD.1.1D*).

The presentation of the high-level design shall be informal (*ADV_HLD.1.1C*).

The high-level design shall be internally consistent (*ADV_HLD.1.2C*).

The high-level design shall describe the structure of the TSF in terms of subsystems (*ADV_HLD.1.3C*).

The high-level design shall describe the security functionality provided by each subsystem of the TSF (*ADV_HLD.1.4C*).

The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software (*ADV_HLD.1.5C*).

The high-level design shall identify all interfaces to the subsystems of the TSF (*ADV_HLD.1.6C*).

The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible (*ADV_HLD.1.7.C*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**
- **ADV_RCR.1: Informal Correspondence Demonstration**

- **ADV_RCR.1: Informal Correspondence Demonstration**

The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided (*ADV_RCR.1.1D*).

For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation (*ADV_RCR.1.1C*).

Dependencies: None.

Class AGD: Guidance Documents

- **AGD_ADM.1: Administrator Guidance**

The developer shall provide administrator guidance addressed to system administrative personnel (*AGD_ADM.1.1D*).

The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE (*AGD_ADM.1.1C*).

The administrator guidance shall describe how to administer the TOE in a secure manner (*AGD_ADM.1.2C*).

The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment (*AGD_ADM.1.3C*).

The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE (*AGD_ADM.1.4C*).

The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate (*AGD_ADM.1.5C*).

The administrator guidance shall describe each type of secure-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF (*AGD_ADM.1.6C*).

The administrator guidance shall be consistent with all other documentation supplied for evaluation (*AGD_ADM.1.7C*).

The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator (*AGD_ADM.1.8C*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**

- **AGD_USR.1: User Guidance**

The developer shall provide user guidance (*AGD_USR.1.1D*).

The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE (*AGD_USR.1.1C*).

The user guidance shall describe the use of user-friendly security functions provided by the TOE (*AGD_USR.1.2C*).

The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment (*AGD_USR.1.3C*).

The user guidance shall clearly present all user responsibility necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of the TOE security environment (*AGD_USR.1.4C*).

The user guidance shall be consistent with all other documentation supplied for evaluation (*AGD_USR.1.5C*).

The user guidance shall describe all security requirements for the IT environment that are relevant to the user (*AGD_USR.1.6C*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**

Class ALC: Life Cycle Support

- **ALC_FLR.3: Systematic Flaw Remediation**

The developer shall document the flaw remediation process (*ALC_FLR.3.1D*).

The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws (*ALC_FLR.3.2D*).

The developer shall designate one or more specific points of contact for user reports and inquiries about security issues involving the TOE (*ALC_FLR.3.3D*).

The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE (*ALC_FLR.3.1C*).

The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw (*ALC_FLR.3.2C*).

The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws (*ALC_FLR.3.3C*).

The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections, and guidance on corrective actions to TOE users (*ALC_FLR.3.4C*).

The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users (*ALC_FLC.3.5C*).

The procedures for processing reported security flaws shall provide safeguards such that any corrections to these security flaws do not introduce new flaws (*ALC_FLC.3.6C*).

The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaws reports and the associated corrections to registered users who might be affected by the security flaw (*ALC_FLC.3.7C*).

Dependencies: None.

Remarks: Not required to satisfy EAL2. Augmented assurance level to ensure that flaws are tracked and corrected, and that information concerning security flaws is disseminated.

Class AMA: Maintenance of Assurances

- **AMA_AMP.1: Assurance Maintenance Plan**

The developer shall provide an AM plan (*AMA_AMP.1.1D*).

The AM plan shall contain or reference a brief description of the TOE, including the security functionality it provides (*AMA_AMP.1.1C*).

The AM plan shall identify the certified version of the TOE, and shall reference the evaluation results (*AMA_AMP.1.2C*).

The AM plan shall reference the TOE component categorization report for the certified version of the TOE (*AMA_AMP.1.3C*).

The AM plan shall define the scope of changes to the TOE that are covered by the plan (*AMA_AMP.1.4C*).

The AM plan shall describe the TOE life-cycle, and shall identify the current plans for any new releases of the TOE, together with a brief description of any planned changes that are likely to have a significant security impact (*AMA_AMP.1.5C*).

The AM plan shall describe the assurance maintenance cycle, stating justifying the planned schedule of AM audits and the target data of the next evaluation of the TOE (*AMA_AMP.1.6C*).

The AM plan shall identify the individual(s) who will assume the role of developer security analysis for the TOE (*AMA_AMP.1.7C*).

The AM plan shall describe how the developer's security analysis role will ensure that the procedures documented or referenced in the AM plan are followed (*AMA_AMP.1.8C*).

The AM plan shall describe how the developer security analyst role will ensure that all developer actions involved in the analysis of the security impact of changes affecting the TOE are performed correctly (*AMA_AMP.1.9C*).

The AM plan shall justify why the identified developer security analyst(s) have sufficient familiarity with the security target, functional specification, TOE high-level design (where appropriate), and with the evaluation results and all applicable assurance requirements for the certified version of the TOE (*AMA_AMP.1.10C*).

The AM plan shall describe or reference the procedures to be applied to maintain the assurance in the TOE, which as a minimum shall include the procedures for configuration management, maintenance or assurance evidence, performance of the analysis of the security impact of changes affecting the TOE, and flaw remediation (*AMA_AMP.1.11C*).

Dependencies:

- **ACM_CAP.2: Configuration Items**
- **ALC_FLR.2: Basic Flaw Remediation**
- **AMA_CAT.1: TOE Component Categorization Report**

Remarks: 1. Maintenance of Assurance is not required to achieve any EAL. **AMA_AMP.1** is included to ensure developer disclosure of current plans for new TOE releases and TOE satisfaction of security targets. 2. **AMA_CAT.1** is not included as a requirement since **AMA_AMP.1** is not a requirement of EAL2. Categorization of TOEs is left to the discretion of the TOE developer.

• **AMA_EVD.1: Evidence of Maintenance Report**

The developer security analyst shall, for the current version of the TOE, provide a security impact analysis that covers all changes affecting the TOE as compared to the certified version (*AMA_EVD.1.1D*).

The security impact analysis shall identify the certified TOE from which the current version of the TOE was derived (*AMA_EVD.1.1C*).

The security impact analysis shall identify all new and modified TOE components that are categorized as TSP-enforcing (*AMA_EVD.1.2C*).

The security impact analysis shall, for each change affecting the security target or TSF representations, briefly describe the change and any effects it has on lower representation levels (*AMA_EVD.1.3C*).

The security impact analysis shall, for each change affecting the security target or TSF representations, identify all IT security functions and all TOE components categorized as TSP-enforcing that are affected by the change (*AMA_EVD.1.4C*).

The security impact analysis shall, for each change which results in a modification of the implementation representation of the TSF or the IT environment, identify the test evidence that shows, to the required level of assurance, that the TSF continues to be correctly implemented following change (*AMA_EVD.1.5C*).

The security impact analysis shall, for each applicable assurance requirement in the configuration management (ACM), life cycle support (ALC), delivery and operation (ADO), and guidance documents (AGD) assurance classes, identify any evaluation deliverables that have changed, and provide a brief description of each change and its impact on assurance (*AMA_AMP.1.6C*).

The security impact analysis shall, for each applicable assurance requirement in the vulnerability assessment (AVA) assurance class, identify which evaluation deliverables have changed and which have not, and give reasons for the decision taken as to whether or not to update the deliverable (*AMA_AMP.1.7C*).

Dependencies:

- **AMA_AMP.1: Assurance Maintenance Plan**
- **AMA_SIA.1: Sampling of Security Impact Analysis**

- **AMA_SIA.1: Sampling of Security Impact Analysis**

The developer security analyst shall, for the current version of the TOE, provide a security impact analysis that covers all changes affecting the TOE as compared to the certified version (*AMA_SIA.1.1D*).

The security impact analysis shall identify the certified TOE from which the current version of the TOE was derived (*AMA_SIA.1.1C*).

The security impact analysis shall identify all new and modified TOE components that are categorized as TSP-enforcing (*AMA_SIA.1.2C*).

The security impact analysis shall, for each change affecting the security target or TSF representations, briefly describe the change and any effects it has on lower representation levels (*AMA_SIA.1.3C*).

The security impact analysis shall, for each change affecting the security target or TSF representations, identify all IT security functions and all TOE components categorized as TSP-enforcing that are affected by the change (*AMA_SIA.1.4C*).

The security impact analysis shall, for each change which results in a modification of the implementation representation of the TSF or the IT environment, identify the test evidence that shows, to the required level of assurance, that the TSF continues to be correctly implemented following the change (*AMA_SIA.1.5C*).

The security impact analysis shall, for each applicable assurance requirement in the configuration management (ACM), life cycle support (ALC), delivery and operation (ADO), and guidance documents (AGD) assurance classes, identify any evaluation deliverables that have changed, and provide a brief description of each change and its impact on assurance (*AMA_SIA.1.6C*).

The security impact analysis shall, for each applicable assurance requirement in the vulnerability assessment (AVA) assurance class, identify which evaluation deliverables have changed and which have not, and give reasons for the decision taken as to whether on not to update the deliverable (*AMA_SIA.1.7C*).

Dependencies:

- **AMA_CAT.1: TOE Component Categorization Report**

Remarks: **AMA_CAT.1** is not included as a requirement since **AMA_SIA.1** is not a requirement of EAL2. Categorization of TOEs is left to the discretion of the TOE developer.

Class ATE: Tests

- **ATE_COV.1: Evidence of Coverage**

The developer shall provide evidence of the test coverage (*ATE_COV.1.1D*).

The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification (*ATE_COV.1.1C*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**
- **ATE_FUN.1: Functional Testing**

- **ATE_FUN.1: Functional Testing**

The developer shall test the TSF and document the results (*ATE_FUN.1.1D*).

The developer shall provide test documentation (*ATE_FUN.1.2D*).

The test documentation shall consist of test plans, test procedure descriptions, expected test results, and actual test results (*ATE_FUN.1.1C*).

The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed (*ATE_FUN.1.2C*).

The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests (*ATE_FUN.1.3C*).

The expected test results shall show the anticipated outputs from a successful execution of the tests (*ATE_FUN.1.4C*).

The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified (*ATE_FUN.1.5C*).

Dependencies: None.

- **ATE_IND.2: Independent Testing – Complete**

The developer shall provide the TOE for testing (*ATE_IND.2.1D*).

The TOE shall be suitable for testing (*ATE_IND.2.1C*).

The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF (*ATE_IND.2.2C*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**
- **AGD_ADM.1: Administrator Guidance**
- **AGD_USR.1: User Guidance**
- **ATE_FUN.1: Functional Testing**

Class AVA: Vulnerability Assessment

- **AVA_MSU.1: Examination of Guidance**

The developer shall provide guidance documentation (*AVA_MSU.1.1D*).

The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation (*AVA_MSU.1.1C*).

The guidance documentation shall be complete, clear, consistent, and reasonable (*AVA_MSU.1.2C*).

The guidance documentation shall list all assumptions about the intended environment (*AVA_MSU.1.3C*).

The guidance documentation shall list all requirements for external security measures, including external procedural, physical, and personnel controls (*AVA_MSU.1.4C*).

Dependencies:

- **ADO_IGS.1: Installation, Generation, and Start-Up Procedures**
- **ADC_FSP.1: Informal Functional Specification**
- **AGD_ADM.1: Administrator Guidance**
- **AGD_USR.1: User Guidance**

Remarks: Augmentation of EAL2 assurance components because of the trend to construct “hybrid” BBMD devices with additional functionality (e.g., web services).

- **AVA_SOF.1: Strength of TOE Security Function Evaluation**

The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim (*AVA_SOF.1.1D*).

For each mechanism with a strength of TOE security function claim the strength of TOE security functional analysis shall show that it meets or exceeds the minimum strength level defined in the PP (*AVA_SOF.1.1C*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**
- **ADV_HLD.1: Descriptive High-Level Design**

- **AVA_VLA.1: Developer Vulnerability Analysis**

The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE (*AVA_VLA.1.1C*).

The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP (*AVA_VLA.1.1D*).

The developer shall document the disposition of obvious vulnerabilities (*AVA_VLA.1.2D*).

Dependencies:

- **ADV_FSP.1: Informal Functional Specification**
- **ADV_HLD.1: Descriptive High-Level Design**
- **AGD_ADM.1: Administrator Guidance**
- **AGD_USR.1: User Guidance**

2.7 Rationale

This section presents evidence to justify requirements set forth in this PP.

2.7.1 Security Objective Rationale

This section provides mappings between policies/threats and security objectives in order to demonstrate coverage.

Threat	Security Objective
T.ADMINISTRATOR T.ANALYSIS	O.CONFMANAGE O.CTRLAUTH O.DETECT O.PROTECTADDR
T.CAPTURE	O.CTRLAUTH O.DETECT
T.COMPNODE	O.PROTECTADDR O.TRUSTEDRECOVERY
T.DENIAL	O.CTRLAUTH
T.FAIL	O.ALARM O.TRUSTEDRECOVERY O.VALIDATION
T.FLAW	O.LIFECYCLE O.PATCHES O.TEST O.VALIDATION
T.MODIFY	O.TRUSTEDRECOVERY O.VALIDATION
T.NETMAP	O.CTRLAUTH O.DETECT O.PROTECTADDR
T.SPOOF	O.CONFINTEGRITY O.DETECT O.PROTECTADDR

Table 2.1: Relationship of TOE Threats to Objectives.

Threats

- **T.ADMINERR: Abuse of Network Administrative Privileges**
 - O.CONFMANAGE ensures the capture and storage of TOE configuration information to allow for the recovery of TOE and network functionality.
 - OE.PERSONNEL ensures that trustworthy network management personnel are retained to manage TOE resources.
- **T.ANALYSIS: Traffic Analysis**
 - O.CTRLAUTH ensures that source and destination peer addresses are authenticated.
 - O.DETECT provides the ability of the TOE to detect unauthorized connections.
 - O.PROTECTADDR protects the integrity of source and destination addresses to prevent data re-routing.
- **T.CAPTURE: Unauthorized Access to Captured Data**
 - O.CTRLAUTH ensures that source and destination peer addresses are authenticated.
 - O.DETECT provides the ability of the TOE to detect unauthorized connections.
- **T.COMPNODE: Compromised Node**
 - O.PROTECTADDR ensures that integrity of stored addresses.
 - O.TRUSTEDRECOVERY ensures the recovery of the TOE to a secure state after a disruption of operations.

Policy	Security Objective
P.ACCOUNTABILITY	O.CTRLAUTH O.LIFECYCLE
P.AUTHENTICATION	O.ACCESSCONTROL O.CTRLAUTH
P.AVAILABILTY	O.ACCESSCONTROL O.ALARM O.TRUSTEDRECOVERY
P.DATA	O.CONFINTEGRITY O.CONFMANAGE
P.DEFAULTCONF	O.CONFMANAGE O.TRUSTEDRECOVERY
P.INTEGRITY	O.CONFINTEGRITY O.CONFMANAGE
P.INTEROPERABILITY	O.PROTOCOLS
P.NOTIFY	O.ALARM
P.PEER	O.ACCESSCONTROL O.CTRLAUTH O.PROTECTADDR
P.SURVIVE	O.ALARM O.CONFMANAGE O.LIFECYCLE O.TEST O.TRUSTEDRECOVERY O.VALIDATION

Table 2.2: Relationship of TOE Policies to Objectives.

Threat	Security Objective of the Environment
T.ADMINERR	OE.PERSONNEL
T.FAIL	OE.ENVIRONMENT OE.PHYSEC
T.MODIFY	OE.PERSONNEL
T.NETMAP	OE.PHYSEC

Table 2.3: Relationship of Threats to Security Objectives in the Environment.

Policy	Security Objective of the Environment
P.ACCOUNTABILITY	OE.DOCUMENTATION OE.PERSONNEL
P.AUTHENTICATION	OE.ACCESSMGMT
P.INSTALL	OE.DOCUMENTATION
P.SURVIVE	OE.ENVIRNMENT OE.PHYSEC

Table 2.4: Relationship of Policies to Security Objectives in the Environment.

- **T.DENIAL: Denial of Service**

- O.CTRLAUTH ensures that source and destination peer addresses are authenticated.

- **T.FAIL: Component of Power Failure**

- O.ALARM allows a quick response to correct faults, errors, and security breaches.
- OE.ENVIRONMENT allows the TOE to be protected from environment threats such as fire, power outages, etc.
- OE.PHYSEC protects TOE resource against harmful physical attack.
- O.TRUSTEDRECOVERY ensures the TOE will recover to a secure state following a fault, error, or security breach.
- O.VALIDATION ensures that all network components are correctly installed, configured, and functioning.

- **T.FLAW: Flaws in hardware, software, and/or firmware**

- O.LIFECYCLE preserves the proper operation of TOE security functions through the operational lifetime of the TOE.
- O.PATCHES ensures that recent fixes and patches are correctly applied to the TOE hardware, software, and/or firmware.
- O.TEST discovers flaws that may lead to faults, errors, and/or security breaches prior to deployment.
- O.VALIDATION ensures the integrity, installation, and functioning of all TOE components and their interoperability with network resources.

- **T.MODIFY: Modification of Protocols**

- OE.PERSONNEL increases the likelihood that trustworthy network management personnel will not make unauthorized modifications or manipulate configuration files pertaining to installed protocols.
- O.TRUSTEDRECOVERY ensures that in the event that T.MODIFY causes TOE disruption, then the TOE and the network can be restored to a secure and functioning state.
- O.VALIDATION validates the integrity and operation of all hardware, software, and/or firmware prior to TOE deployment in an operational network.

- **T.NETMAP: Network Mapping**

- O.CTRLAUTH ensures that source and destination peer addresses are authenticated.
- O.DETECT ensures the detection of unauthorized connections.
- OE.PHYSEC reduces the threat of an unauthorized connection with the intent to capture network information.
- O.PROTECTADDR ensures integrity of source and destination addresses.

- **T.SPOOF: Spoofing Attack**

- O.CTRLAUTH ensures that authentication increases the difficulty for an intruder to access the TOE.
- O.DETECT detect unauthorized or suspect connections.
- O.PROTECTADDR ensures that that source and destination addresses are not compromised, thus minimizing an attacker's ability to obtain legitimate addresses.

Policies

- **P.ACCOUNTABILITY: Individual Accountability**
 - O.CTRLAUTH provides for authentication in accordance with the access control policy to ensure organizations can be held accountable for their actions.
 - OE.DOCUMENTATION ensures developers are held accountable for providing proper installation and usage documentation.
 - O.LIFECYCLE ensures developers are accountable for the preservation or enhancement of security features when releasing TOE upgrades to hardware, software and/or firmware.
 - OE.PERSONNEL ensures organization and network administrative personnel are held accountable on the grounds of proper training, configuration, and maintenance.
- **P.AUTHENTICATION: Authentication of Operators and Nodes**
 - O.ACCESSCONTROL ensures that authentication must occur in order for an access control policy to be effective.
 - OE.ACCESSMGMT ensures that the chain of secure authentication for network administrators is intact.
 - O.CTRLAUTH allows connectivity only after authentication.
- **P.AVAILABILITY: Network Availability**
 - O.ACCESSCONTROL allows access to network resources to authorized users only.
 - O.ALARM ensures network availability be alerted to failures, errors, and security breaches to enable a quick response to correct the problem.
 - O.TRUSTEDRECOVERY ensure that network availability is restored to the state prior to the fault.
- **P.DEFAULTCONF: Default Configuration**
 - O.CONFMANAGE ensures that the configuration management plan captures and stores default TOE configuration values.
 - O.TRUSTEDRECOVERY supports default configuration by ensuring recovery to a secure operational state after a failure if the recovery is to revert to a default setting.
- **P.INSTALL: Proper Installation**
 - OE.DOCUMENTATION ensures that developer documentation delivered to end users is properly aligned with correct installation procedures.
- **P.INTEGRITY: Content Integrity**
 - O.CONFINTEGRITY ensures that stored information related to the TOE retains content integrity.
 - O.CONFMANAGE ensures that stored network management information retains content integrity.
- **P.INTEROPERABILITY: Interoperability**
 - O.PROTOCOLS ensures that protocols are correctly implemented in the TOE to enable predictable cross-device interoperability according to standards.
- **P.NOTIFY: Notification of Failure**
 - O.ALARM ensures the TOE will be capable of detecting failures or errors in any component and alerting network management personnel.

- **P.PEER: Peer Nodes**

- O.ACCESSCONTROL ensures that only those peers with proper authorization can access the node.
- O.CTRLAUTH ensure that connectivity will only be provided when entities have been properly authorized in accordance with the access control policy.
- O.PROTECTADDR ensures that traffic will be transmitted between trusted TOEs by providing integrity of the transmitting and receiving entity addresses.

- **P.SURVIVE: Network Survivability and Recovery**

- O.ALARM ensures survivability by providing notification of failures, errors, and security breaches to network management personnel to enable a quick response.
- O.CONFMANAGE ensures retention of configuration information as well as content integrity of this stored information to enable quick re-creating of network state.
- OE.ENVIRONMENT ensures that physical facilities are adequate to enable quick recovery from environmental threats, such as fire, power loss, etc.
- O.LIFECYCLE ensures that security functions are maintained throughout the life of the TOE so that resources can retain the ability recover from faults, errors, and security breaches.
- OE.PHYSEC ensures that physical facilities are adequate to enable quick recovery from faults, errors, and security breaches.
- O.TEST ensures that the network can quickly recover from faults, errors, and security breaches.
- O.TRUSTEDRECOVERY provides assurance that the network can quickly recover from faults, errors, and security breaches.
- O.VALIDATION ensures that all hardware, software, and firmware are correctly installed and functional so that the network can quickly recover from faults, errors, and security breaches.

2.7.2 Dependency of Requirements

This section presents coverage of dependencies for functional requirement and assurance requirements.

2.7.3 Rationale for Evaluation Assurance Level 2

Evaluation Assurance Level 2 was chosen for this PP, including a few augmentations. This level was decided upon after considering the level of vulnerability, documented threats, and possible risks associated with broadcast messaging in BACnet networks. The current BACnet Annex J standard is well-defined and broadly accepted. EAL2 was chosen over EAL1 for two primary reasons: the acceptance of foreign devices by BBMDs and the trend to construct multifunctional devices including BBMDs, web services, and other networking functionality.

EAL2 requires the cooperation of the developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than is consistent with good commercial practice. As such, it should not require a substantially increased investment of cost or time. EAL2 is therefore applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.

EAL2 has been augmented to include additional assurance requirements and, hence, a higher assurance level than EAL3. EAL3 requires more design description and life cycle support in order to decrease the possibility of TOE tampering during development. Additionally, EAL3 requires a higher level of independent validation than is necessary for the TOE addressed in this PP. The non-maintenance assurance augmented assurance levels are:

Functional Requirement	Dependency List
FAU_GEN.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1, FIA_UID.1
FAU_SAR.1	FAU_GEN.1
FAU_SEL.1	FAU_GEN.1, FMT_MTD.1
FDP_ACC.1	FDP_ACF.1
FDP_ETC.1	FDP_ACC.1, FDP_IFC.1
FDP_IFC.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1, FMT_MSA.3
FDP_ITC.1	FMT_MSA.3
FDP_UIT.1	FDP_ACC.1
FIA_UAU.2	FIA_UID.1
FIA_UID.2	None
FMT_MOF.1	FMT_SMR.1
FMT_MSA.1	FDP_ACC.1
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1
FMT_MTD.1	FMT_SMR.1
FPT_AMT.1	None
FPT_FLS.1	ADV_SPM.1
FPT_PHP.1	FMT_MOF.1
FPT_STM.1	None
FPT_TDC.1	None
FPT_TST.1	FPT_AMT.1
FRU_FLT.1	FPT_FLS.1
FRU_PRS.1	None
FTA_TSA.1	None
FTP_TRP.1	None

Table 2.5: Functional Requirements and Associated Dependencies

- **ADV_FSP.2: Fully Defined External Interfaces**

ADV_FSP.1: Information Functional Specification is required by EAL2. It is upgraded to **ADV_FSP.2** to ensure that TOE security functional requirements are completely described. This will facilitate thorough testing of the TOE. Given the completeness of BACnet Annex J, this upgrade is reasonable.

- **ALC_FLR.3: Systematic Flaw Remediation**

This is not required of EAL2 (or any other EAL). This will ensure that flaws are tracked, corrected, and security-related information distributed to TOE users.

- **AVA_MSU.1: Examination of Guidance**

This is not required of EAL2. This is included to ensure that vulnerabilities to BBMDs in multipurpose device commonly marketed as “BBMDs” will be examined appropriately.

Maintenance of assurance is designated by **AMA_AMP.1**, **AMA_EVD.1**, and **AMA_SIA.1**. TOE Categorization Reports are not specifically given in this PP and are left to the discretion of the developer.

Assurance Requirement	Dependency List
ACM_CAP.2	None
ADO_DEL.1	None
ADO_IGS.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1
ADV_HLD.1	ADV_FSP.1, ADV_RCR.1
ADV_RCR.1	None
AGD_ADM.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1
ALC_FLR.3	None
AMA_AMP.1	ACM_CAP.2, ALC_FLR.2, AMA_CAT.1
AMA_EVD.1	AMA_AMP.1, AMA_SIA.1
AMA_SIA.1	AMA_CAT.1
ATE_COV.1	ADV_FSP.1, ATE_FUN.1
ATE_FUN.1	None
ATE_IND.2	ADC_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1
AVA_MSU.1	ADO_IGS.1, ADC_FSP.1, AGD_ADM.1, AGD_USR.1
AVA_SOF.1	ADV_FSP.1, ADV_HLD.1
AVA_VLA.1	ADV_FSP.1, ADV_HLD.1, AGD_ADM.1, AGD_USR.1

Table 2.6: Assurance Requirements and Associated Dependencies

Chapter 3

Addendum to Clause 24 of the BACnet Standard

3.1 Introduction

This article focuses on Clause 24 of the ASHRAE-135 BACnet Standard [1] (hereafter referred to as *Clause 24*) that deals with network security. The security properties of *Clause 24* are consistent with this document. The security properties are: data confidentiality, data integrity, and authentication of peer entities, data origin, and operators. The topics addressed herein are:

- symmetric key length,
- the Advanced Encryption Standard (AES) symmetric block cipher,
- formal analysis of the current *Clause 24* security protocol and
- public-key encryption and BACnet systems.

It is intended that this document will serve as a basis of discussion for active BACnet developers, including the recently formed SSPC Network Security Working Group (NS-WG).

3.2 Symmetric Key Size

Currently, *Clause 24* specifies two separate 56-bit Data Encryption Standard (DES) keys. One key is for device authentication (PK_x), and the other is for exchanging session data (SK). The SK s are encrypted using the PK_x of device x . Generation and distribution of the PK_x s are considered “local matters”, whereas the SK s are generated and distributed by a central key server.

The capacity of a 56-bit key length is $2^{56} = 7.2 \times 10^{16}$ possible unique keys. Assuming (conservatively) that current technology allows 10^6 decryptions per microsecond ¹, a 56-bit DES key can be broken by brute force exhaustive search in 20 hours. This fact was exposed by the Electronic Frontier Foundation (EFF) when they developed a special purpose “DES cracker” machine for less than \$250,000 in July 1998 [5]. Before then, a report by Blaze et al. [2] in January 1996 highlighted the weakness of 56-bit DES keys. When DES was first developed in 1977, a 56-bit key length was sufficient. Because of Moore’s law, a 56-bit key is no longer sufficient.

Current key lengths are commonly set to a minimum size of 128 bits, which gives a capacity of $2^{128} = 3.4 \times 10^{38}$ unique keys. Under the same performance assumptions above, a 128-bit symmetric block cipher key can be cracked in 1.8×10^{19} years by exhaustive search.

From this perspective, *Clause 24* should be updated to specify a minimum key length of 128 bits. However, the context in which the encryption is being used should be considered as well. The time and cost associated

¹This figure is feasible with faster processors coupled with distributed computing environments.

with breaking a cipher should be compared to the lifetime and value of the protected data in BACnet systems, which depends on the operational scenarios of those systems. The ASHRAE RP-1011 [7] defines a set of operational information services and the data/communications requirements necessary to enable them. The services defined are (1) revenue meter reading (electricity, gas, water, heating and cooling energy), (2) quality of service (QoS) monitoring, (3) real-time pricing (RTP) transmission, (4) load management service, (5) on-site generation supervisory control, (6) energy efficiency monitoring, (7) weather reporting and forecasting services, (8) indoor air quality (IAQ) monitoring, and (9) dynamic demand bidding (DDB) into a power exchange. Each service generates and exchanges information with varying effective lifetimes, bandwidth, and security requirements (Table 3.1).

Service	Eff. Lifetime	Bandwidth	Security
Revenue meter reading	24 hours	High (96 data items/meter)	High
QoS	Immediate	Same as (1)	High
RTP	24 hours	Low	High (bilateral contracts)
Load management	Immediate	Low	High (plant control systems)
On-site generation	Immediate	Dependent on of control variables	Same as (4)
Weather information	24 hours	Low	Low
Energy efficiency monitoring	Immediate	Low to High	Unknown
IAQ monitoring	24 hours	Same as (7)	Same as (7)
DDB to Power Exchange	Strict timing req. (short intervals)	High (on demand)	High

Table 3.1: RP-1011 operational services compared to their respective effective lifetimes, bandwidth requirements, and data security requirements [7].

In some cases, information is valuable within a span of minutes; in others, information has an effective lifetime of 24 hours. The majority of services require a high degree of data security. If information is encrypted using a 56-bit key DES cipher, then the information should have a lifetime of considerably less than one or two days. Otherwise, the information is not adequately protected against cryptanalysis. Therefore, a 128-bit key length is suggested as a minimum.

3.3 AES Symmetric Block Cipher

The National Institutes of Standards and Technology (NIST) recognized the need for a successor to DES, partly based on the 56-bit key length vulnerability. The main selection criteria were security, performance, and flexibility. The Advanced Encryption Standard (AES) [10] is the result of that process and has the following properties: [8]:

- The AES algorithm must be a symmetric block cipher.
- The algorithm must allow three possible key bit lengths: 128, 192, and 256.
- The algorithm must work on 128 bit blocks of data (as compared to the 64-bit block length of DES).

The resultant implementation is based on the symmetric block Rijndael algorithm [4], which uses ten rounds of four operations. A positive attribute to BACnet systems is that Rijndael is quite amenable to hardware implementations, including Smart Cards and Field Programmable Gate Arrays (FPGAs).

Much work has been undertaken to understand the performance attributes of the Rijndael algorithm. We cite a study by Schneier and Whiting [12]² that compared the five AES algorithm candidates in terms of

²Many studies are available, and we believe [12] to be credible and objective.

performance on both 32-bit and 64-bit processors. The 32-bit processor space spanned high-end microprocessors as well as embedded smart card processors. Across 32-bit and 64-bit microprocessors represented by Pentium, Pentium II, HP PA8200, and IA 64, the Rijndael algorithm takes approximately between 120 and 320 clock cycles for 128-bit encryption in assembly language. For the same processor set, the same operation coded in C takes between 250 and 900 clock cycles.

For 8-bit embedded processors, memory requirements are generally more of a system constraint than performance. Fortunately, the Rijndael algorithm requires much less memory than DES in memory constrained environments [6, 12]. Keating [6] uses a Motorola 6802 simulator to show the maximum RAM requirement of AES is 37 bytes compared to 96 bytes for DES. For ROM implementations, AES require 553 bytes of memory. DES, however, requires 680 bytes of ROM memory for encryption only, and 1036 bytes for encryption and key scheduling operations. This characteristic appears to be consistent across other embedded processor architectures [12].

AES will replace DES and should be considered a black-box function in the context of BACnet systems. The replacement of DES with AES in *Clause 24* has no technically adverse effects to the security of the protocol. In fact, the replacement of DES with AES in many IETF protocols is considered a trivial undertaking and is already underway. DES should be viable for backward compatibility for a short time period. A key size of 128 bits is appropriate for most BACnet systems.

3.4 BACnet Security Protocol Analysis

3.4.1 Assumptions and Attacks

Security protocols exist to distribute cryptographic keys or apply cryptographic operations to information. Key distribution protocols facilitate private and authenticated communication between two peers by providing the peers with secure cryptographic keys. Authentication provides assurance to a peer that a message was sent by another peer. The goal of an attacker is the subversion of these operations.

Analysis of protocol vulnerabilities is based on two assumptions. An attacker is assumed to be capable of eavesdropping on messages and modifying them. However, cryptographic algorithms are assumed to be strong enough such that the attacker cannot decrypt messages without the encryption key. Protocol attacks with the highest impact on BACnet network security are described in the following list.

1. **Man-in-the-middle** attacks involve an intruder interjecting himself into the communication path between two target peers and masquerading to each as the other peer. Simple protocols without peer authentication, such as Diffie-Hellman, tend to be easily subverted in this manner.
2. **Type flaws** result from sequences of bits in a well-formed message being transposed to cause the message's recipient to misinterpret the contents. Nonces³ and key values are typically the target of these attacks.
3. **Parallel interleaving attacks** use messages from one protocol run to form messages in another concurrent protocol run. That is, the attacker causes messages from two distinct protocol runs to *overlap*.
4. **Replay (freshness) attacks** occur when message information from a previous protocol run is used by an intruder to provide message content in a future run. This attack is possible when a message recipient is unable to determine the freshness of message information. For example, if an attacker cracks an earlier message exchange between two devices *A* and *B*, then the session key SK_{AB} may be reused.
5. **Implementation dependent flaws**, such as interactions between protocols and encryption algorithms, predictable generators for random numbers and nonces, and inadequate coverage of testing, are possible avenues for attackers to exploit. These attacks are hard to predict and analyze. They can be alleviated by better software engineering and testing practices.

³The term "nonce" is used to denote a token value that has an immediate and temporary usefulness.

This list is not exhaustive and does not include cryptanalysis, timing, or power analysis ⁴ attacks. This list represents the most common attacks that a BACnet network is likely to experience in real world implementations. All attacks excluding the replay attack compromise cryptographic key information by subverting traffic patterns of current protocol runs. It is a true that it is impossible to develop a protocol to prevent all attacks. However, protocols have been developed to prevent attacks that are easily launched (see Appendix A). A relevant example is that the *replay* attack can be defended against with a well-designed protocol.

3.4.2 Freshness of SK_{AB}

In *Clause 24*, secure communication between two peer devices A and B depends on each device receiving a session key SK_{AB} from a trusted key server S . The protocol for obtaining a session key is ⁵:

$$A \rightarrow S : RequestKey(A, B) \tag{3.1}$$

$$S \rightarrow B : \{SK_{AB}.A\}_{PK_B} \tag{3.2}$$

$$B \rightarrow S : n_B \tag{3.3}$$

$$S \rightarrow A : \{SK_{AB}.B\}_{PK_A} \tag{3.4}$$

$$A \rightarrow S : n_A \tag{3.5}$$

Each device receives SK_{AB} and places it in its `Device` object in the `List_Of_Session_Keys` field. From the specification, it does not appear that SK_{AB} is given a finite lifetime. An intruder can request a secure communications exchange with one of the devices using an older session key. It can also spoof the Message Initiation Authentication since it has the session key. It is recommended that during the course of requesting an enciphered session (Section 24.3.1 of *Clause 24*), the server device (B) check the freshness of the session key in addition to issuing a Message Initiation Authentication procedure. Timestamps that represent a creation date and a well-defined procedure for session key expiration have been successful in thwarting similar replay attacks in other security protocols (see the discussion on NSSK in the Appendix).

3.5 Considerations Regarding Public-Key Cryptography

Currently, *Clause 24* specifies the use of private key or symmetric cryptography. This type of cryptography uses a single key to encrypt and decrypt information, and the assumption is that the key is known only to those peers involved in the encryption and decryption process. In the 1970's, a new method was developed called public-key or asymmetric encryption. This method uses two keys per peer, one key is kept private by the peer and the other is made public to all other peers. The idea is that the public key is used to encrypt plaintext information, and the private key is used to decrypt the ciphertext back to plaintext.

A benefit of public-key cryptography is that it theoretically negates the requirement for a third-party key distribution server, thereby reducing the complexity of the network topology and the number of communication rounds necessary to establish a secure session between two devices. However, in practice, this is not as easily achieved since the public keys of each device require an infrastructure to publicize and distribute them. Furthermore, asymmetric encryption requires a key *pair*, not a single key as is the case with symmetric encryption. The bit sizes are different between the two methods. For example, a 128 bit key length in AES is comparable to a RSA key size of 2560 bits. Initial key distribution is an issue as is the ramification of key discovery by intruders.

A hybrid approach, where device public-keys are used to exchange session keys, may be useful for securing group communication. If a device's public key is known to an entire group of BACnet devices, then it is possible to encrypt the session key and data using the hybrid encryption method, broadcast them to multiple devices, and expect each device to be able to decrypt and respond to messages from the initiating device. This is another advantage of the hybrid method over the method described in *Clause 24*, which does not enable secure broadcast messaging.

⁴This is different from the concept of power monitoring with which the BACnet community may be familiar.

⁵The notation $\{M\}_K$ denotes plaintext M is encrypted using key K . The notation $P \rightarrow Q : M$ indicates peer P sends message M to peer Q . The message M may be a composition of messages, i.e. $M \equiv M_1.M_2$.

3.6 Conclusion

This addendum highlights current aspects of network security relevant to BACnet systems specified in *Clause 24*. The issue of increasing cryptographic key lengths from 56 bits to 128 bits is considered concomitant to migrating away from the DES block cipher algorithm in favor of the AES algorithm. Also, we discuss likely attacks against the protocol specified in *Clause 24*. The NSSK protocol is discussed and compared with the Denning-Sacco and Yahalom protocols in light of forward secrecy attacks. Finally, we discuss public-key encryption algorithms and reasons against adopting them in BACnet systems.

We stop short of making an explicit recommendation of which protocol to use because there are many variables to simultaneously consider, such as the time-value of the information, the number of communication rounds of the protocol, and the sensitivity of the information.

Appendix A

Common Security Protocols

Needham-Schroeder Secret Key Protocol

The classic security protocol is the Needham-Schroeder Secret-Key (NSSK) protocol [3, 11]. This approach depends on a trusted third party authority to authenticate device identity and generate a session key for data exchange between two devices. The NSSK protocol establishes authenticated key distribution between two peers. Using notation from *Clause 24* and [11], device A establishes secure communication with device B with the help of a trusted server S in the following manner:

$$A \rightarrow S : A.B.n_A \tag{A.1}$$

$$S \rightarrow A : \{n_A.B.SK_{AB}.\{SK_{AB}.A\}_{PK_B}\}_{PK_A} \tag{A.2}$$

$$A \rightarrow B : \{SK_{AB}.A\}_{PK_B} \tag{A.3}$$

$$B \rightarrow A : \{n_B\}_{SK_{AB}} \tag{A.4}$$

$$A \rightarrow B : \{n_B - 1\}_{SK_{AB}} \tag{A.5}$$

The nonces values (n_k) correspond to the pseudo random numbers generated as part of the *Authenticate Service* (Section 24.5 of *Clause 24*).

The NSSK protocol is vulnerable to replay attacks because device B cannot determine the freshness of the session key SK_{AB} from the information contained in message (A.3). An attacker that captures messages from a previous protocol run between devices A and B compromises the session key SK_{AB} can then masquerade as device A at a later time. Device B will accept the session key SK_{AB} from the masquerading device, believing that it is communicating with device A .

This vulnerability has been addressed by two different protocols, the Denning-Sacco protocol and the Yahalom protocol.

Denning-Sacco Protocol

The *Denning-Sacco protocol* (DS) [3] relies on timestamps instead of nonces to ensure that recipient device B can check the staleness of received key information. Using the same notation as the NSSK protocol and T as a timestamp, the DS protocol is:

$$A \rightarrow S : A.B \tag{A.6}$$

$$S \rightarrow A : \{B.SK_{AB}.T.\{A.SK_{AB}.T\}_{PK_B}\}_{PK_A} \tag{A.7}$$

$$A \rightarrow B : \{A.SK_B.T\}_{PK_B} \tag{A.8}$$

The DS protocol reduces the number of communication rounds between device A and device B from five to three. Device B is capable of judging the timeliness of message (A.3) thereby negating the need for additional exchanges of messages (A.4) and (A.5) in the NSSK protocol.

Yahalom Protocol

The *Yahalom protocol* protects against replay attacks by using nonces in a completely different way than the NSSK protocol. The Yahalom protocol is formally described as:

$$A \rightarrow B : A.n_A \tag{A.9}$$

$$B \rightarrow S : B.\{A.n_a.n_B\}_{PK_B} \tag{A.10}$$

$$S \rightarrow A : \{B.SK_{AB}.n_A.n_B\}_{PK_A}.\{A.SK_{AB}\}_{PK_B} \tag{A.11}$$

$$A \rightarrow B : \{A.SK_{AB}\}_{PK_B}.\{n_B\}_{SK_{AB}} \tag{A.12}$$

Both devices A and B submit their nonce values before a request to generate a session key is sent to the trusted server. The intervention of a nonce value from device B before the server request allows both devices A and B to check the staleness of the session key SK_{AB} upon receipt. This protocol also reduces the number of communication rounds necessary to establish secure communication between two devices.

Appendix B

Acronyms

BBMD	BACnet Broadcast Management Device
BDT	Broadcast Distribution Table
BVLC	BACnet Virtual Link Control
BVLCI	BACnet Virtual Link Control Information
BVLL	BACnet Virtual Link Layer
B/IP	BACnet Internet Protocol
B/IP-M	B/IP Multicast
CC	Common Criteria
EAL	Evaluation Assurance Level
FDT	Foreign Device Table
IP	Internet Protocol
PP	Protection Profile
PPP	Point-to-Point Protocol
SF	Security Function
SFP	Security Function Policy
SLIP	Serial Line Internet Protocol
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSE	TOE Security Environment
TSF	TOE Security Functions
TSP	TOE Security Policy
UDP	User Datagram Protocol

References

- [1] ASHRAE Standard 135. BACnet: A data communication protocol for building automation and control networks, 1995.
- [2] Matt Blaze, Whitfield Diffie, Ronald Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener. Minimal lengths for symmetric ciphers to provide adequate commercial security: A report by an ad hoc group of cryptographers and computer scientists. <http://www.crypto.com/papers/keylength.txt>.
- [3] John Clark and Jeremy Jacob. A survey of authentication protocol literature. Technical report, Department of Computer Science, University of York, 1997.
- [4] Joan Daemen and Vincent Rijmen. The block cipher Rijndael. In Quisquater and Schneier, editors, *Smart Card Research and Applications*, volume 1820 of *LNCS*, pages 288–296. Springer-Verlag, 2000.
- [5] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*. O'Reilly and Associates, 1998.
- [6] Geoffrey Keating. Performance analysis of AES candidates on the 6805 CPU core. In *The Second Advanced Encryption Standard AES Candidate Conference*, March 1999.
- [7] Michael Kintner Meyer and Martin Burns. Utility/energy management and controls system (EMCS) communication protocol requirement. Technical Report RP-1011, ASHRAE, 1999.
- [8] Susan Landau. Communications security for the twenty-first century: The Advanced Encryption Standard. *Notices of the AMS*, 47(4), 2000.
- [9] National Bureau of Standards. *Data Encryption Standard*. Federal Information Processing Standards (FIPS) Publication no. 46, 1977.
- [10] National Institute of Standards and Technology. *Advanced Encryption Standard*. Federal Information Processing Standards (FIPS) Publication no. 197, 2001.
- [11] Peter Ryan and Steve Schneider. *Modelling and Analysis of Security Protocols*. Addison-Wesley, 2001.
- [12] Bruce Schneier and Doug Whiting. A performance comparison of the five AES finalists. In *The Third Advanced Encryption Standard Candidate Conference (AES3)*, April 2000.