ISO/CCITT and Internet Management Coexistence (IIMC):

ISO/CCITT to Internet Management Security

(IIMCSEC)

February, 1993

Lee LaBarre (Editor)

The MITRE Corporation
Burlington Road
Bedford, MA 01730
cel@mbunix.mitre.org

Status of this Memo

This document provides information to the network and systems management community. This document is intended as a contribution to ongoing work in the area of multi-protocol management coexistence and interworking. This document is part of a package; see also [IIMCOMIBTRANS] [IIMCMIB-II] [IIMCPROXY] and [IIMCIMIBTRANS]. Distribution of this document is unlimited. Comments should be sent to the Network Management Forum IIMC working group (iimc@thumper.bellcore.com).

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the 1id-abstracts.txt listing contained in the internet-drafts Shadow Directories on ds.internic.net, nic.nordu.net, ftp.nisc.sri.com, munnari.oz.au to learn the

current status of any Internet Draft.

Abstract

This document is intended to facilitate the multi-protocol management coexistence and interworking for networks that are managed using the ISO/CCITT Common Management Information Protocol (CMIP) and networks that are managed using the Internet Simple Network Management Protocol (SNMP). This document defines the end-to-end security architecture, services, and mechanisms for use with ISO/CCITT-Internet proxies. This document also contains the ISO/CCITT GDMO definition and registration of the SNMP Parties MIB, derived from the Internet SNMP Parties MIB [27] according to the procedures defined in "Translation of Internet MIBs to ISO/CCITT GDMO MIBs" [32].

Table of Contents

1. INTRODUCTION 1

1.1 PROBLEM STATEMENT.....1

1.2 OVERVIEW OF IIMC.....2

1.3 MIB TRANSLATION PROCEDURES.....3

1.4 NATIVE MANAGEMENT MODEL.....3

1.5 PROXY MANAGEMENT MODEL.....5

1.6 SCOPE OF THIS DOCUMENT.....6

1.7 TERMS AND CONVENTIONS.....6

2. SECURITY AND MANAGEMENT CONSIDERATIONS 8

2.1 GENERAL CONSIDERATIONS.....8
2.1.1 Security of Management 8
2.1.2 Management of Security 8

2.1.3 Threat Characterization 9
2.1.3.1 Communications Path Security.....9
2.1.3.2 Managed System Security.....10

2.2 ISO/CCITT TO INTERNET SECURITY ENVIRONMENT.....11
2.2.1 Security Model 11
2.2.2 Security Capabilities 12
2.2.3 Internet Management Security 13
2.2.3.1 SNMPv1 Security.....13
2.2.3.2 SNMPv2 Security.....13
2.2.4 Constraints on Mapping Security Services 14
2.2.5 SNMP Security and the ISO Access Control Framework.....16

3. SECURITY SPECIFICATIONS 17

3.1 ISO MANAGER TO ISO/CCITT-INTERNET PROXY SECURITY...17
3.1.1 Peer Authentication Services 17
3.1.2 Transfer of SNMP Security Parameters 18
3.1.3 ISO/CCITT-Internet Proxy Party MIB 20

3.2 ISO/CCITT-INTERNET PROXY TO INTERNET AGENT SECURITY 20

3.3 ISO/CCITT-INTERNET PROXY ACCESS CONTROL ENFORCEMENT 20

4. IIMC PARTY MIB 21

-- 4.1 PARTY MIB GDMO TEMPLATES.....21
-- 4.1.1 Party MIB Managed Object Classes 22
-- 4.1.2 Party MIB Attribute Types 26
-- 4.1.3 Party MIB Attributes 28
-- 4.1.4 Party MIB Name Bindings 48

-- 4.2 PARTY MIB ASN.1 MODULES.....50

5. IIMC ACL MIB 53

-- 5.1 IIMC ACL MIB GDMO TEMPLATES.....55
-- 5.1.1 IIMC ACL MIB Managed Object Classes 55
-- 5.1.2 IIMC ACL MIB Attributes 56
-- 5.1.3 IIMC ACL MIB Name Bindings 57

-- 5.2 IIMC ACL MIB ASN.1 MODULES.....58

6. CONFORMANCE 59

ANNEX A (NORMATIVE): MANAGED OBJECT CONFORMANCE STATEMENTS (MOCS).....A-1

ANNEX B: GLOSSARY B-1

ANNEX C: REFERENCES C-1

List of Figures

FIGURE 1. MIB TRANSLATION 3

FIGURE 2. NATIVE MANAGEMENT 4

FIGURE 3. PROXY MANAGEMENT 5

FIGURE 4. IIMC END-TO-END SECURITY MODEL 12

List of Tables

TABLE 1. SNMP SECURITY SERVICES 15

TABLE 2. SNMP SECURITY PARAMETERS 19

REVISION HISTORY

Issue 1.0, October 1993

This is the first issue of this document. The internet draft <draft-labarre-iimc-party-04>, dated February, 1994, is identical in content to Issue 1.0, October 1993. It has been reformatted for posting as an internet draft.

1. INTRODUCTION

This section provides an overview of ISO/CCITT and Internet Management Coexistence (IIMC) activities, insight into the problem being addressed by IIMC, and a brief introduction to the strategy adopted by IIMC: use of translated MIBs in either a proxy or native implementation. The section concludes by describing the scope of this document, and terms and conventions used by this document.

1.1 PROBLEM STATEMENT

The need for enterprise network management has been addressed by development of network management standards within various communities, most notably the ISO/CCITT and Internet communities.

* The ISO/CCITT community developed the Common Management Information Protocol (CMIP) [6], and related SMI documents [8,9,10].

* The Internet community developed the Simple Network Management Protocol (SNMP) [19], and its successor, SNMPv2 [28]. The Internet SMI is defined in [18] and [24].

These standards share a nearly common management model, but diverge due to differing management philosophies. Although functionally similar, the Internet and ISO/CCITT protocols and SMIs differ in terms of their complexity and specific operations. Business requirements for end-to-end enterprise management include the need to integrate the management of many different devices, potentially owned or administered by many independent organizations. This requires components to be accessed by ISO/CCITT management, Internet management, and proprietary management mechanisms in a manner which presents a unified view of the network, despite protocol and SMI differences.

For example, many telecommunications and computer vendors, represented by organizations such as the Network Management Forum (NMF), and the U.S. government, as specified in the Government Network Management Profile (GNMP) Version 1.0 [37], have based their enterprise management model on the ISO/CCITT management model. These organizations are particularly interested in integrated management of devices that use the Internet management. This interest is primarily due to the widespread commercial implementation and use of

such devices, especially devices that use the Internet TCP/IP protocol suite.

1.2 OVERVIEW OF IIMC

The ISO/CCITT and Internet Management Coexistence (IIMC) package includes the following documents.

IIMCIMIBTRANS	Translation of Internet MIBs to ISO/CCITT GDMO MIBs [33]
IIMCOMIBTRANS	Translation of ISO/CCITT GDMO MIBs to Internet MIBs [32]
IIMCMIB-II	Translation of Internet MIB-II (RFC 1213) to ISO/CCITT GDMO MIB [30]
IIMCPROXY	ISO/CCITT to Internet Management Proxy [31]
IIMCSEC	ISO/CCITT to Internet Management Security

These documents together comprise a package aimed at integrating ISO/CCITT-based and Internet-based management systems.

IIMC specifications address the problem that end-to-end management requires an integrated, unified view of the managed network, despite differences in management protocol and information structure. Integrated management can be facilitated by the development of "proxy" mechanisms which translate between functionally equivalent service, protocol, and SMI differences to create this unified view. MIB translation procedures can be used to support proxy management, as well as to take advantage of existing MIB definition and avoid duplication of effort. In this way, commercial investment in both ISO/CCITT and Internet-based management technologies can be preserved through deployment of common methods and tools which support integration.

This overall strategy was outlined in a joint publication developed by the NM Forum and X/Open entitled "ISO/CCITT and Internet Management: Coexistence and Interworking Strategy" [35]. The documents included in the IIMC package are the next level of detailed specifications which implement several of the methodologies identified in the strategy. Additional specifications may be defined in the future.

1.3 MIB TRANSLATION PROCEDURES

The foundation of IIMC is provided by a pair of Management Information Base (MIB) translation procedures.

* IIMCIMIBTRANS [24] specifies translation procedures for converting MIBs from Internet MIB macro format into ISO/CCITT GDMO template format.

* IIMCOMIBTRANS [33] specifies translation procedures for converting MIBs from ISO/CCITT GDMO template format into Internet MIB macro format.

The IIMC approach is to specify direct translation procedures which yield a pair of functionally-equivalent MIBs, as shown in Figure 1.

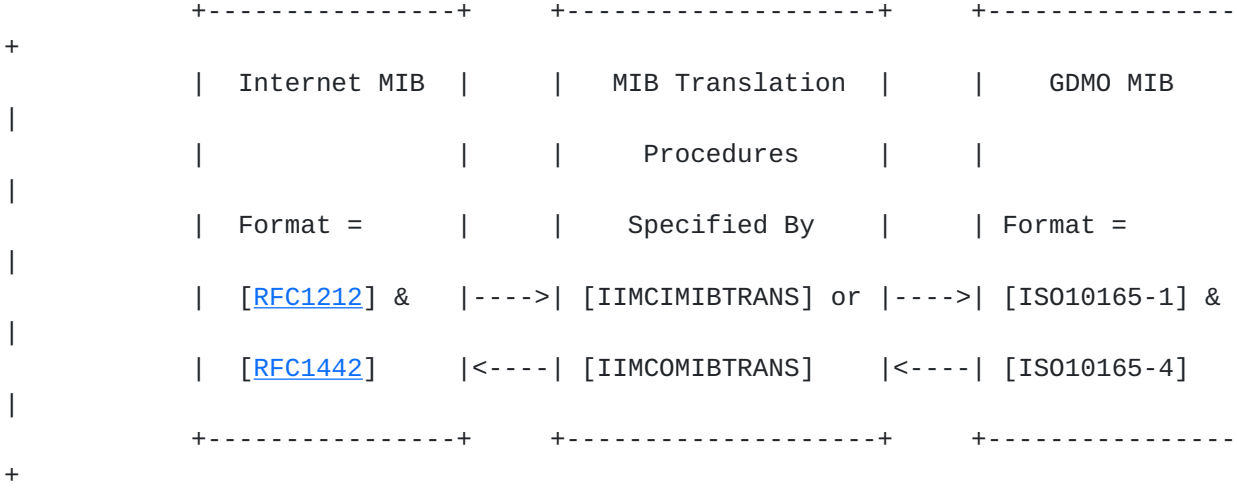


Figure 1. MIB translation.

MIBs translated by these procedures may be used to take advantage of existing MIB definitions when business needs require deployment in a different management environment. Translated MIBs may also be used to provide uniformity when multiple management environments are supported by a single system (e.g., dual stack managers). Finally, IIMC MIB translation procedures may be used to support service emulation by a proxy.

1.4 NATIVE MANAGEMENT MODEL

The basic model for ISO/CCITT and Internet management is illustrated in the following diagram.

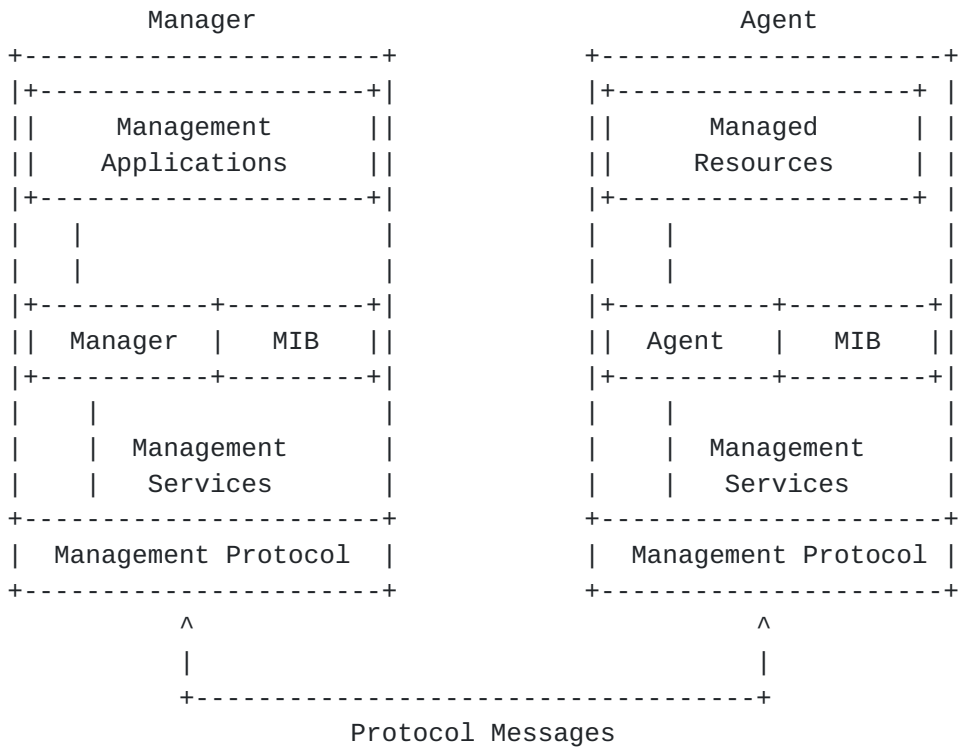


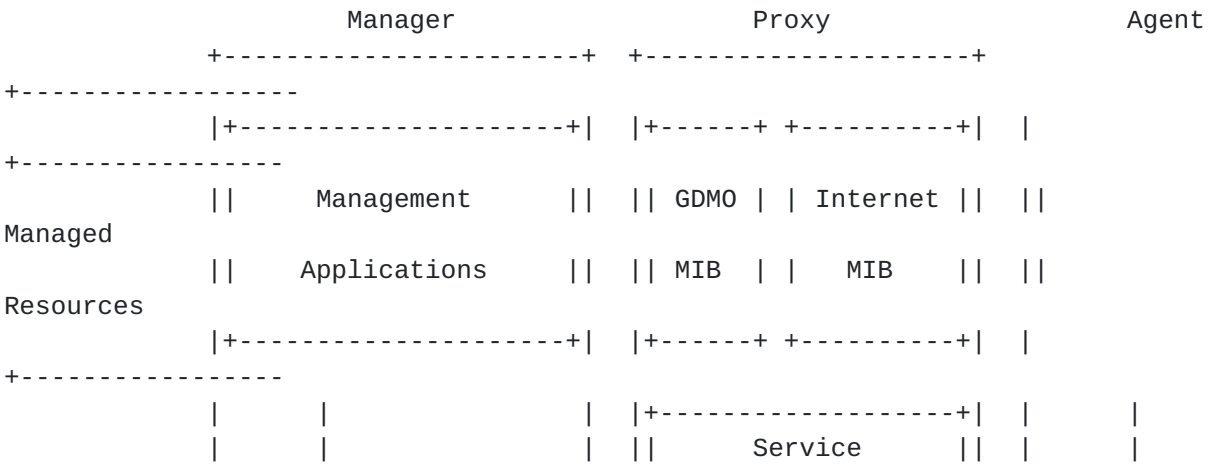
Figure 2. Native management.

Within IIMC documents, this model is referred to as the "native" management model. MIBs translated using IIMC procedures can be used by "native" agent implementations. For example, an ISO/CCITT agent can make visible TCP/IP managed resources using the translated GDMO version of the

Internet MIB-II [21] specified by [30]. Dual-stack managers or agents may also be implemented which support both the original MIB and the translated MIB generated using IIMC-specified procedures.

1.5 PROXY MANAGEMENT MODEL

The basic model for ISO/CCITT to Internet proxy management is illustrated in the following diagram. This proxy is specified by [31]. A similar approach could also be taken to specify an Internet to ISO/CCITT proxy, although no such IIMC document is currently specified.



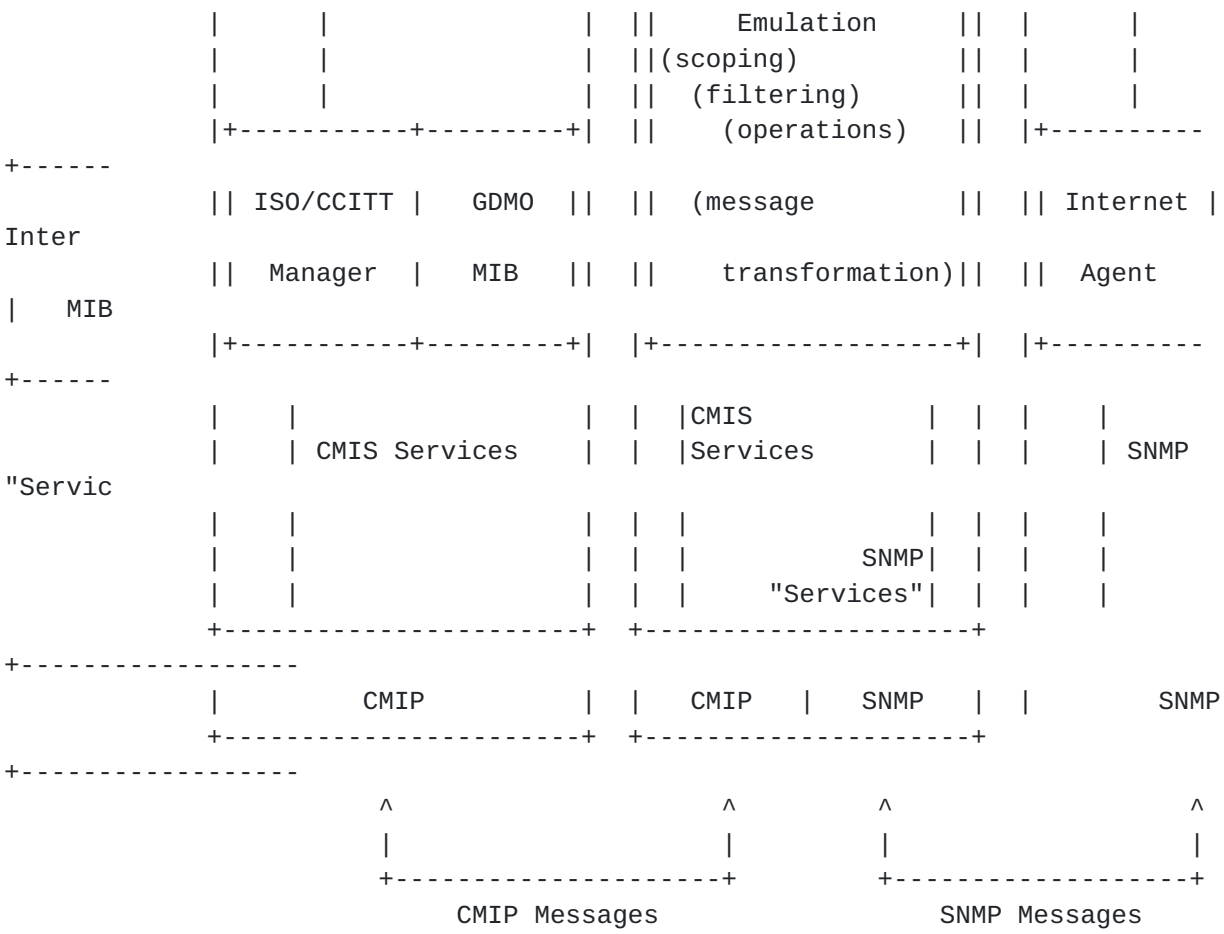


Figure 3. Proxy management.

This ISO/CCITT to Internet proxy provides emulation of CMIS services by mapping to the corresponding SNMP message(s) necessary to carry out the service request. The service emulation allows management of Internet objects by an ISO/CCITT manager. The left hand side of the proxy behaves like an ISO/CCITT agent, communicating with the ISO/CCITT manager using CMIP protocols. The right hand side of the proxy behaves like an Internet manager, communicating with the Internet agent using SNMP protocols.

The proxy relies on the existence of a pair of directly-related MIB definitions, where the Internet MIB has been translated into ISO/CCITT GDMO using the procedures specified in IIMCIMIBTRANS. The proxy uses these MIB definitions and rules to provide run-time translation of

management information carried in service requests and responses.

The proxy is designed with a specified interface between the proxy and the underlying protocol stacks, and so deals primarily in terms of CMIS services and SNMP "services". The proxy emulates services such as CMIS scoping and filtering, processing of CMIP operations, and forwarding/logging of CMIS notifications by performing a mapping process which must be tailored for each protocol (for example, SNMPv1 and SNMPv2 are variants of the same protocol mapping process).

1.6 SCOPE OF THIS DOCUMENT

One of the IIMC objectives is to provide for the secure end-to-end management of resources managed using ISO/CCITT and Internet management services, protocols and SMI. Security and management by their very nature are entwined such that each needs the services of the other. Security services are required to protect management services. Management services are required to monitor and control security services.

This document (IIMCSEC) defines the security architecture for end-to-end security between an ISO/CCITT manager and an Internet agent via proxies such as that defined in [31]. The architecture requires that information required to support Internet security mechanisms from an end-to-end perspective, and to manage it, be translated into the ISO/CCITT SMI. This document applies the procedures described in [33] to the translation and registration of the Internet SNMP Parties MIB defined in [27].

This document primarily addresses issues concerning security architecture and interoperability of security mechanisms. Issues concerning trusted implementations, although important, are beyond the scope of this document.

1.7 TERMS AND CONVENTIONS

This document assumes that the reader is familiar with the ISO/CCITT SMI and Internet SMI, and the terminology of each. The term SNMP will be used throughout the document to indicate either SNMPv1 or SNMPv2, unless a distinction needs to be made.

This document assumes that the reader is familiar with the

ISO/CCITT and Internet management security services, protocols and mechanisms.

This document assumes that the reader is familiar with the Internet to SMI translation procedures defined in [33].

Other terms and conventions used throughout this document are defined in Section 2.

2. SECURITY AND MANAGEMENT CONSIDERATIONS

Security and management are entwined by their very nature such that each needs the services of the other. Security services are needed to protect management services. Management services are needed to monitor and control security services. These considerations are briefly presented in this section. Additional background information can be found in ISO/IEC 7498-2, OSI Reference Model - Part 2: Security Architecture [38].

2.1 GENERAL CONSIDERATIONS

2.1.1 Security of Management

Management is most vulnerable to security attacks at the manager user interface, the communications path over which management messages are transmitted, and at the managed system that contains the resources being managed. Accordingly, management's security considerations are to overcome these threats by:

* Preventing unauthorized operator access to manager applications and associated management information contained in a manager workstation,

* Protecting management information in transit between managers and agents, and

* Enforcing management policy regarding access to information within the managed system.

Preventing unauthorized access to manager applications is beyond the scope of this document, and therefore will not be discussed. The characterization of the security threats in relation to the other two vulnerable areas are discussed more fully in the following sections.

2.1.2 Management of Security

Security requires management support for three basic activities:

* monitoring and control of security mechanisms,
* detection of security related events through security alarm generation, reporting and audit trail analysis, and
* damage assessment and recovery from a security attack.

Security mechanisms and algorithm resources are modeled as managed objects and the management information is stored in the management information base. The same management and security mechanisms used to manage non-security managed objects may be applied to the management of security objects, and the generation of security notifications associated with their operation.

2.1.3 Threat Characterization

Security threats for management are the same as for any distributed application. Security threats can be characterized as being active or passive. Active threats to a management system may effect changes to the state or operation of the managed resource. Examples of active threats are malicious changes to the routing tables of a system, or to the objects used to control decisions related to policies, such as security policies relating to resource access.

Active threats include:

* masquerade,
* modification and fabrication of messages and stored data,
* replay and reordering of messages, and
* denial of management services.

Passive threats are those which, if realized, would not result in any modifications to information contained in the system, e.g., management information, and where neither the operation nor the state of the system is changed.

Passive threats include:

* disclosure of message contents and stored data,
* traffic analysis, and
* repudiation.

2.1.3.1 Communications Path Security

The threats to the communications path used for manager to agent communications, and applicable security services include:

* modification and fabrication of management messages
    * integrity

* disclosure of management message data
    * confidentiality, selective field confidentiality

* replay and reordering of messages
    * integrity

* denial of management services
    * continuity of operations

* traffic analysis
    * confidentiality

Note that the communications path from the manager to an agent may be direct, or indirect via the management applications of an intermediate manager or proxy. In the indirect case, the portion of the message that must be exposed in the intermediate manager for the purpose of application layer relaying is subject to unauthorized disclosure and modification. Such entities must be trusted

not to perform such modifications and not to disclose the contents of the management messages. Selective field confidentially services may be needed if intermediate managers or proxies are acting as application layer relays in the path. Such selective field services allow only the information in management messages needed for application layer routing to be unprotected while preventing other fields in the message from disclosure or modification.

2.1.3.2 Managed System Security

The threats to the managed system include:

* masquerade of a manager application or operator
    * peer authentication, data origin authentication

* modification and fabrication of data residing in the management information base
    * access control, data integrity

* disclosure of management data in the managed system
    * access control, confidentiality

* repudiation of management requests at destination
    * non-repudiation at destination.

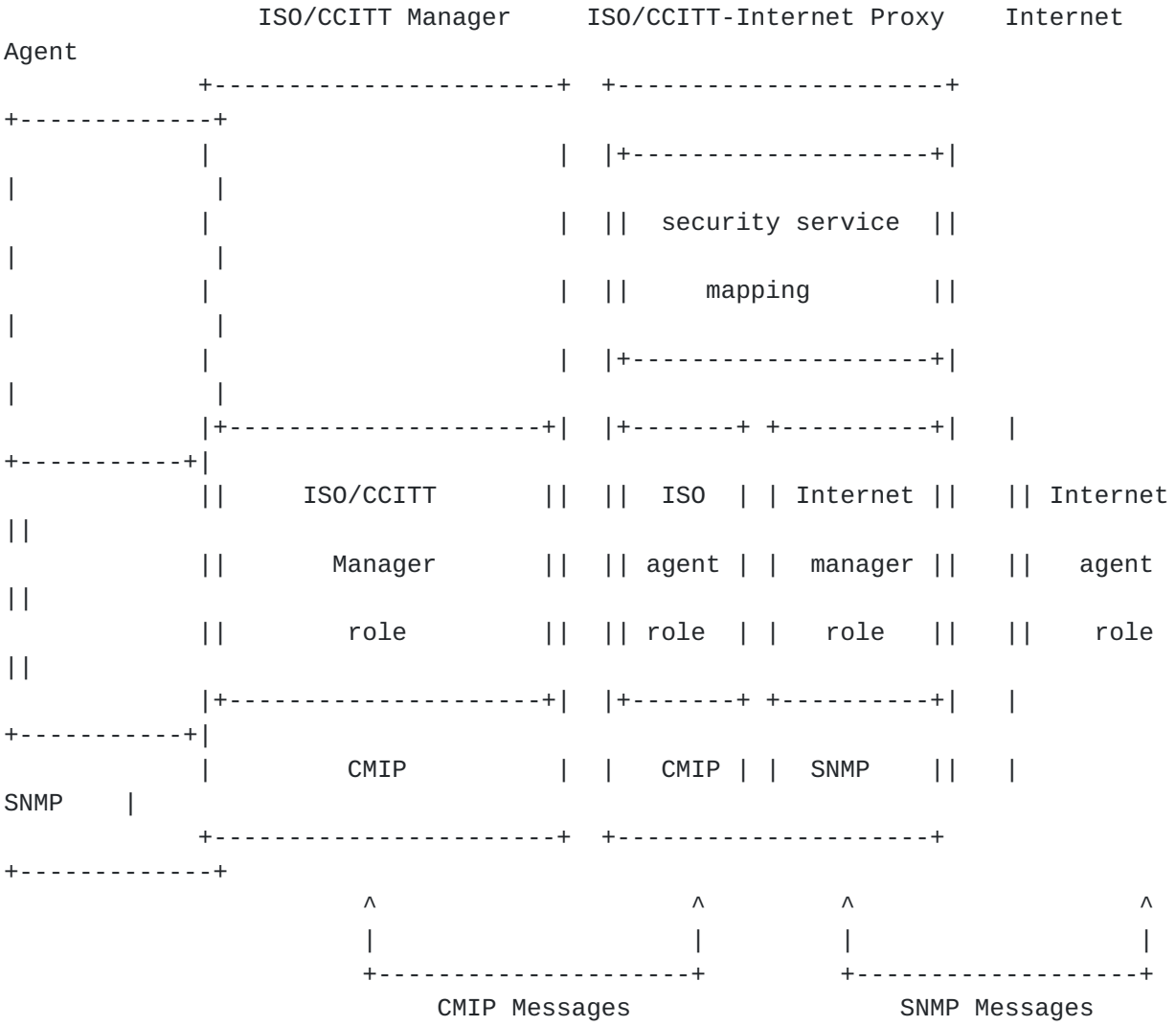
Non-repudiation services may be provided in circumstances where such accountability is needed. While the non-repudiation service does nothing to protect the network, it does provide the capability to trace the entities that are to be blamed for mis-management.

2.2 ISO/CCITT TO INTERNET SECURITY ENVIRONMENT

2.2.1 Security Model

The model for IIMC end-to-end security is illustrated in Figure 4. The objective is to provide continuity of security services from the ISO/CCITT Manager through to the Internet Agent. The end-to-end solution is constrained by the security services available at the Internet agent and those available at the ISO/CCITT Manager. The mapping of security services is provided by the ISO/CCITT-Internet proxy. The mapping of those services at the proxy depends upon the availability of the services and the compatibility of the mechanisms used to provide the services.

This figure illustrates the proxy in a separate device from the manager or the agent. If the proxy function is performed in the manager, then how the manager's internal security mechanisms map to Internet security services is beyond the scope of this document. If ISO management services and protocol are provided in the managed device (native CMIP agent), the ISO security services apply at the managed system. The mapping of any ISO security services that may still possibly apply at the internal proxy to Internet agent interface into equivalent Internet services, e.g., authentication and access control, is beyond the scope of this document.



* ISO peer authentication
* ISO data origin authentication* * Internet data origin authentication
* ISO integrity, confidentiality* * Internet integrity, confidentiality
* Internet access control * Internet access control#
* ISO access control+

* OSI application layer standards [13,14,15,16] are in progress. These services may also be provided by lower layers in some environments, e. transport and network

```
        # SNMPv1 mechanisms differ

        + ISO access control may be applied by the proxy to GDMO
objects, if
        enforcement is at the proxy.
```

Figure 4. IIMC End-to-end Security Model.

All security services do not have to be provided at the same layers in the protocol suites on the two external proxy interfaces. For example, integrity and confidentiality services may be applied at the transport or network layer at the interface to the ISO/CCITT manager, and at the application layer at the interface to the Internet agent. However, authentication and access control services should be provided at the application layer so that the same granularity of control may be achieved on both sides of the interface. For example, access should be controlled to the application user, and to the level of individual attributes within OSI objects.

Some security services may not be needed depending on the environment and the security policy. For example, data origin authentication and confidentiality services may not be needed between the proxy and ISO/CCITT manager if the two

devices are close together and physical security is adequate to satisfy the security policy.

2.2.2 Security Capabilities

The basic security capabilities that should be met by an architecture for providing end-to-end security services are:

* enforcement of SNMPv1 security services at the agent (community string, and possibly source node address),

* enforcement of SNMPv2 security services at the agent (party/context based),

* enforcement of access control at the proxy using OSI access control mechanisms [7] for the ISO/CCITT managed objects derived from Internet objects for all proxied agents, and for the MIB specific to the proxy [31],

* OSI security services between the ISO/CCITT manager and the proxy, e.g., those provided by [13,14,15,16], and

* mapping of OSI security services into Internet security services, where possible, and forwarding from the ISO/CCITT manager of information needed for Internet security mechanisms.

2.2.3 Internet Management Security

The security services for Internet management differs depending on the version of SNMP (SNMPv1 or SNMPv2) used.

2.2.3.1 SNMPv1 Security

The SNMPv1 security relies on the transfer of an unprotected community string that represents the capabilities that the initiator has with respect to operations on a set of objects.

2.2.3.2 SNMPv2 Security

The SNMPv2 security architecture relies on the identification of distinct, globally unique, entities, called "parties", for peers that exchange SNMP messages [25]. Multiple parties may exist at the manager and at the agent.

Each distinct SNMPv2 peer is identified by a "party identifier", an OID. Associated with the party identifier are it's agent address, and parameters for access control, authentication, integrity and confidentiality services which may be used when communicating with other parties. Since

parties form a peer relationship, these security service parameters for peer parties must be compatible.

The peer relationship between SNMPv2 parties is established via an associated "context", identified by an OID, which provides a means to identify constraints on valid management operations and access to associated resources (MIB objects). The context also specifies whether the constraints apply to local resources or to remote resources via a (yet another)

proxy relationship.

The minimal SNMPv2 security service allowed is access control as specified by the source (srcParty), destination party (dstParty), and context identifiers. SNMPv2 requests that do not contain all three identifiers are discarded.

As discussed in 2.2.5, the access control scheme used by SNMP security can be considered a form of capability scheme.

2.2.4 Constraints on Mapping Security Services

The mapping of security services end-to-end is constrained by the security services available at the Internet agent. The possible application level security services at Internet agents is well defined for SNMPv1, and for SNMPv2. However, it cannot be assumed that all Internet agents will implement the full range of their defined security services, or that they are all required for any given environment.

Given the known potential availability of Internet security services at Internet agents, and at the Internet proxy, three major problems arise in proxy situations:

a) Selection from the security services available at the Internet proxy to Internet agent interface of those services that are appropriate to the threats at that interface, according to the established security policy.

b) Providing security services at the ISO manager to Internet proxy interface that are appropriate to the threats at that interface, according to the established security policy.

c) Transfer to the Internet proxy from the ISO manager of security parameters required for Internet proxy to Internet agent security.

Note: An exact mapping of security services between both Internet proxy interfaces may not be required. The environments at the two interfaces may be completely different, e.g., the manager and proxy may be in the same room while the agents are geographically distributed.

Assume the following environment and constraints.

i)The ISO/CCITT-Internet proxy is geographically remote from both the ISO/CCITT manager and the Internet agents, and the threats at both interfaces are the same.

ii)Only application level security services are used.

iii)The Internet agents and Internet proxy support the full range of security services defined for them. They include, for the respective SNMP versions:

Service	SNMPv1	SNMPv2
Peer Authentication	-	X
Data Origin Authentication	-	X
Access Control	X	X
Connectionless Integrity	-	X
Connectionless Confidentiality	-	X
Replay, Reorder Protection	-	X

Table 1. SNMP security services.

The first problem (a) can be solved by configuring the security parameters of the Internet agents and Internet proxy, either through local or remote management mechanisms.

The second problem (b) can be solved by implementing the appropriate OSI management services in the ISO/CCITT manager and ISO/CCITT-Internet proxy, and configuring the mechanisms to provide the service. This problem is complicated by the current lack of Stable OSI security standards for data origin authentication, integrity, confidentiality, and access control. Future documents will describe how the stable versions of [7], Objects and Attributes for Access Control, will be applied to access control at the proxy, and [13,14,15,16], Generic Upper Layer Security, will be applied to provide data origin authentication, integrity, and confidentiality services between the ISO/CCITT manager and the proxy.

The third problem (c) can be solved by using an access control certificate to transfer the Internet security parameters. This problem is complicated by the current lack of a stable standard for access control certificates. Given the necessity for such transfers in proxy situations, an preliminary Access Control Certificate (ACC) will have to be used. However, an attempt should be made to align as closely as possible with proposed ISO standards.

2.2.5 SNMP Security and the ISO Access Control Framework

The SNMP access control schemes can be most nearly categorized as capability schemes using the definitions in the ISO Access Control Framework [12]. A capability defines a set of allowed operations that the initiator of the operation is allowed to perform on an identified set of targets.

The SNMPv1 scheme is a very weak capability scheme which uses the community string to identify which operations are permitted or not permitted on a set of objects. However the community string is not bound to the initiator, i.e., it may possibly be changed in transit. Proof of its authenticity can be inferred from the fact that the SNMPv1 agent is configured to have knowledge of the capabilities represented by the community string. In that sense, the person that configured the community string, either via local mechanisms, or via remote management mechanisms can be considered to provide the third party level of authenticity which is acceptable for the environment. However, if security management parameters, exchanged between the manager and the agent, or proxy, are not protected, then authenticity is not guaranteed since the community string may have been compromised.

The SNMPv2 scheme is a stronger capability scheme, tied to a strong source authentication mechanism, which uses the combination of SNMPv2 srcParty, dstParty, and context identifiers to identify which operations are permitted or not-permitted on a set of objects. The srcParty binds it to the initiator of the request. Proof of its authenticity can be inferred from the fact that the SNMPv2 agent is configured to have knowledge of the capabilities represented by the interrelated parameters. In that sense, the person that configured the Internet Party MIB, either via local mechanisms, or via remote management mechanisms can be considered to provide the third party level of authenticity which is acceptable for the environment. However, if security management parameters, exchanged between the manager and the agent, or proxy, are not protected, then authenticity is not guaranteed since the security parameters

may have been compromised.

Capabilities may be carried in access control tokens or access control certificates. Tokens and certificates are similar. A token differs from a certificate in that it is not created by an entity which is a security authority, and it does not necessarily contain an indication of the time period for which it is valid. The sealing of the ACC by a security authority and the provision of a time period for which it is valid provides third party proof of its authenticity.

3. SECURITY SPECIFICATIONS

3.1 ISO MANAGER TO ISO/CCITT-INTERNET PROXY SECURITY

OSI data origin authentication services, integrity services, confidentiality services, and access control services are currently beyond the scope of this document due to the lack of stable relevant ISO security standards. Specifications for these services, based on [7] and [13,14,15,16], will be defined in a future Issue when the relevant base standards become International Standards.

3.1.1 Peer Authentication Services

OSI peer authentication services shall be supported in accordance with OMNIPoint 1 security specifications in [34] Annex B. The authentication class supported shall be Class 2, as defined in [34] Annex G. The minimum requirements for Class 2 authentication are support for the use of simple authentication with protected password as specified in [34] Annex B.2.

The authentication value for Class 2 authentication shall include the ASN.1 SimpleCredentials definition contained in [34] Annex B.2, including the name field (an ASN.1 DistinguishedName), the validity field with sub-field time1

included, and the password field with the choice of PROTECTED OCTET STRING that contains the OID for the hash algorithm used in the hashing function and a transformed password.

The procedure for producing the transformed password shall be as follows:

1) Insert the initiator Distinguished Name into the name field of the SimpleCredentials construct.

2) Insert the current time into the validity.time1 field of the SimpleCredentials construct.

3) Insert the password value into the password field of the SimpleCredentials construct.

4) Encode the SimpleCredentials construct using the Distinguished Encoding Rules as specified in [17], Part 12, clause 7.14.1.

5) Apply the hash algorithm to the encoded SimpleCredentials to produce the hash value, i.e., the transformed password.

The password field of the SimpleCredentials construct shall be an octet string formatted as:

<hash algorithm OID> || "@" || <transformed password>

where "||" means concatenate, and:

<hash algorithm OID> - the alphanumeric form of the OID that contains the OID sub-identifiers represented as decimal integers separated by decimal points, i.e., the familiar "dot" notation.

3.1.2 Transfer of SNMP Security Parameters

Support shall be provided for the transfer of Internet security service parameters from the ISO/CCITT manager to an ISO/CCITT-Internet proxy at the time of management association establishment, and with CMIP requests. Access

control security parameters shall be transferred as security attributes of an Access Control Certificate (ACC), also referred to as a Privilege Attribute Certificate (PAC). Implementations shall support the ACC defined in OMNIPoint 1 [34] Annex K, and the associated PICS in Annex E.8.8.

To allow for later inclusion of security attributes for OSI access control, the ACC transferred to an ISO/CCITT-Internet proxy shall be a compound ACC, with the first ACC in the sequence of contained ACCs being the ACC with the SNMP security attributes. The containing ACC of the compound ACC transferred to an ISO/CCITT-Internet proxy shall contain the privilegeAttributes sequence, in accordance with [34] Annex E.8.8. The privilegeAttributes sequence may be empty.

The Internet security parameters shall be transferred using the security attributes defined by [36]. The security attributes are defined in the Security-Services-Attributes module in [36] using the attribute macro defined for the ISO Directory Services [4].

Note: For ease of use, these definitions are repeated below; in the event of transcription error, the original source specification [36] is normative.

The security parameters to be transferred in the ACC shall include the Security-capability-type-1 attribute as defined in [36].

The Security-capability-type-1 attribute grants access of the type accessType to the object(s) defined by objectDefiner.


```
Security-capability-type-1 ::= ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX CapabilityType1
     SINGLE VALUE

CapabilityType1 ::= SEQUENCE {
     objectDefiner  ObjectDefiner,
     accessType     AccessDefiner}

ObjectDefiner ::= IntegerOrString
AccessDefiner ::= IntegerOrString

IntegerOrString ::= CHOICE {
```

```
          integerPart    INTEGER,
          stringPart     IA5String}
```

The IntegerOrString choice shall be IA5String.

The Security-capability-type-1 attribute is registered as:

```
      desd-att-capability-type-1 OBJECT IDENTIFIER ::=
{iso(1) identified-organization(3) icd-ecma(0012)
standard(0) desd(138) attributeIdentifiers(3) 4}
```

The Security-capability-type-1 attribute shall be used for proxy to SNMPv1 agents to carry the community string indicating the accessType. The objectDefiner shall be undefined and represented as an empty string (''H).

The Security-capability-type-1 attribute shall be used for proxy to SNMPv2 agents to carry the srcParty and dstParty as the objectDefiners, and the context identifier as the accessType.

The contents of the Security-capability-type-1 attribute shall be as described below.

ECMA [36] Security Attribute	SNMPv2 Security Parameter	SNMPv1 Security Parameter
Security-capability-type-1		
objectDefiner	src/dst Parties	""H
accessType	context	community string

Table 2. SNMP security parameters.

The src/dst Parties are the srcParty and dstParty SNMPv2 parameters separated by a slash (/). The SNMPv2 parameters shall be ASN.1 object identifiers represented using the "dot" notation convention, where the OID sub-identifiers are represented in decimal character form and separated by decimal points. For example, {1 3 6 1 6 3 3 2 3 1 1 3} is represented as "1.3.6.1.6.3.3.2.3.1.1.3".

3.1.3 ISO/CCITT-Internet Proxy Party MIB

The ISO/CCITT-Internet proxy shall maintain the security

parameters used for communicating with the ISO/CCITT manager in a partyEntry managed object within the Party MIB instantiated in the proxy.

The OID for the hash algorithm used for authentication shall be contained in the partyAuthProtocol object.

The OID for the default MD5 hash algorithm shall be as defined in [17], clause 7.10.4.1.

The password used for authentication shall be contained in the partyAuthPrivate columnar object, and shall be stored and updated in accordance with the procedures defined for the object, i.e., the exclusive OR mechanism.

The partyAuthLifetime object shall contain the value, in seconds, to be added to the time1 value contained in the SimpleCredentials construct passed to the proxy by the manager. If the value of time1, augmented by the partyAuthLifetime value, is less than the proxy's local notion of time, then the authentication shall be considered invalid.

3.2 ISO/CCITT-INTERNET PROXY TO INTERNET AGENT SECURITY

An ISO/CCITT-Internet proxy that supports SNMPv1 shall support the community string security services as defined in [19].

An ISO/CCITT-Internet proxy that supports SNMPv2 shall support the security services as defined in [27], with minimal support for the "Party No Privacy" compliance level specified by clause 3.7.1 of [27].

3.3 ISO/CCITT-INTERNET PROXY ACCESS CONTROL ENFORCEMENT

Access control enforcement at the ISO/CCITT-Internet proxy is currently beyond the scope of this document. However, access control enforcement at the proxy will be based on OSI access control for management as defined in [7] when it becomes an International Standard.

4. IIMC PARTY MIB

Support for the SNMPv2 security services requires that some of the Internet Party MIB [27] be instantiated in the ISO/CCITT-Internet proxy, and SNMPv2 agents. The IIMC Party MIB is derived from the Internet Party MIB defined in [27]. Adjustments have been made to the behavior of some elements in the MIB to accommodate SNMPv1 community string based security.

A Naming Tree diagram for IIMC Party MIB managed object classes is illustrated below. The IIMC Party MIB is subordinate to the ISO/CCITT system managed object that represents the Internet agent or proxy.

```
"Rec. X.721 | ISO/IEC 10165-2 : 1992":system
     |
     |
     partyMIBObjects
        |
        |---partyEntry
        |
        |---contextEntry
        |
        |---aclEntry (instantiated only at agent)
        |
        |---viewEntry (instantiated only at agent)
```

The IIMC Party MIB elements that are specific to the proxy, i.e., local instantiations of partyMIBGroup, partyEntry and contextEntry, should be instantiated under the ISO system object that is specific to the proxy. The IIMC MIB elements that are specific to each SNMP agent are actually instantiated in the SNMP agent. Duplicates of MIB elements instantiated in the SNMP agent should not be instantiated in the proxy if a stateless approach to proxy is used as described in [31].

The GDMO templates and ASN.1 modules are included here in one section to facilitate automated processing. Comments and subsection headers are included in the form of ASN.1 comments, i.e., preceded by "--".

This document (IIMCSEC) is allocated the following registration identifier for purposes of referencing the

translated RFC 1147 Party MIB contained herein.

```
  iimcRFC1447 OBJECT IDENTIFIER ::={ iimcAutoDocument 1447 }


-- 4.1  PARTY MIB GDMO TEMPLATES
```



```
-- 4.1.1  Party MIB Managed Object Classes


aclEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
          aclEntryPkg PACKAGE
     BEHAVIOUR
          aclEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This managed object class maps to aclEntry
               object in [27].!!;
          DESCRIPTION
               !!The access privileges for a particular
               requesting SNMP party in accessing a particular
               target SNMP party.!!;
          INDEX     SNMPv2-Party-MIB.aclTarget,
               SNMPv2-Party-MIB.aclSubject,
               SNMPv2-Party-MIB.aclResources;
          ENDPARSE!;;
          ATTRIBUTES
               aclEntryId                    GET,
               aclTarget                     GET,
               aclSubject                    GET,
               aclResources                  GET,
               aclPrivileges
                    DEFAULT VALUE IIMCRFC1447ASN1.c-aclPrivileges
                    GET-REPLACE,
               aclStorageType
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTStorageType
                    GET-REPLACE,
               aclStatus                GET-REPLACE;;;
REGISTERED AS  {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1};
```

```
contextEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     contextEntryPkg PACKAGE
     BEHAVIOUR
          contextEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This managed object class maps to
               contextEntry object in [27].!!;
          DESCRIPTION
               !!Locally held information about a particular
               SNMPv2 context.!!;
          INDEX     IMPLIED SNMPv2-Party-MIB.contextIdentity;
          ENDPARSE!;;
          ATTRIBUTES
```



```
               contextEntryId               GET,
               contextIdentity              GET,
               contextIndex            GET-REPLACE,
               contextLocal
                    DEFAULT VALUE IIMCRFC1447ASN1.c-contextLocal
                    GET-REPLACE,
               contextViewIndex        GET-REPLACE,
               contextLocalEntity
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTNullString
                    GET-REPLACE,
               contextLocalTime
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-contextLocalTime
                    GET-REPLACE,
               contextProxyDstParty         GET-REPLACE,
               contextProxySrcParty         GET-REPLACE,
               contextProxyContext     GET-REPLACE,
               contextStorageType
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTStorageType
                    GET-REPLACE,
               contextStatus           GET-REPLACE;;;
REGISTERED AS  {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1};
```

```
partyEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
     CHARACTERIZED BY
     partyEntryPkg PACKAGE
     BEHAVIOUR
          partyEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !REFERENCE
               !!This managed object class maps to
               partyEntry object in [27].!!;
          DESCRIPTION
               !!Locally held information about a particular
               SNMPv2 party.!!;
          INDEX     IMPLIED SNMPv2-Party-MIB.partyIdentity;
          ENDPARSE!;;
          ATTRIBUTES
               partyEntryId             GET,
               partyIdentity            GET,
               partyIndex               GET,
               partyTDomain
                    DEFAULT VALUE IIMCRFC1447ASN1.c-partyTDomain
                    GET-REPLACE,
               partyTAddress
                    DEFAULT VALUE IIMCRFC1447ASN1.c-partyTAddress
                    GET-REPLACE,
               partyMaxMessageSize
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-partyMaxMessageSize
                    GET-REPLACE,
```



```
               partyLocal
                    DEFAULT VALUE IIMCRFC1447ASN1.c-partyLocal
                    GET-REPLACE,
               partyAuthProtocol
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-partyAuthProtocol
                    GET-REPLACE,
               partyAuthClock
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-partyAuthClock
                    GET-REPLACE,
               partyAuthPrivate
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTNullString
```

```
                    GET-REPLACE,
               partyAuthPublic
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTNullString
                    GET-REPLACE,
               partyAuthLifetime
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-partyAuthLifetime
                    GET-REPLACE,
               partyPrivProtocol
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-partyPrivProtocol
                    GET-REPLACE,
               partyPrivPrivate
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTNullString
                    GET-REPLACE,
               partyPrivPublic
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTNullString
                    GET-REPLACE,
               partyCloneFrom      GET-REPLACE,
               partyStorageType
                    DEFAULT VALUE
                    IIMCRFC1447ASN1.c-DEFAULTStorageType
                    GET-REPLACE,
               partyStatus              GET-REPLACE;;;
REGISTERED AS  {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1};

partyMIBObjects  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
          partyMIBObjectsPkg PACKAGE
          BEHAVIOUR
          partyMIBObjectsPkgBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This managed object class maps to
               partyMIBObjects group OID in [27].!!;
```



```
          DESCRIPTION
               !!This group contains the security related
               parameters needed for communicating with SNMP
```

```
                agents.  The security services to which these
                parameters apply are authentication, integrity,
                confidentiality, and access control.

                This object class contains only the naming
                attribute.!!;
           ENDPARSE!;;
           ATTRIBUTES
                           partyMIBObjectsId       GET;;;
REGISTERED AS  {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2};

viewEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
          viewEntryPkg PACKAGE
          BEHAVIOUR
          viewEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This managed object class maps to viewEntry
               object in [27].!!;
          INDEX     SNMPv2-Party-MIB.viewIndex,
               IMPLIED SNMPv2-Party-MIB.viewSubtree;
          DESCRIPTION
               !!Information on a particular family of view
               subtrees included in or excluded from a
               particular  SNMPv2 context's MIB view.

               Each SNMPv2 context which is locally
               accessible has a single MIB view which is
               defined by two collections of view subtrees:
               the included view subtrees, and the excluded
               view subtrees.  Every such subtree, both
               included and excluded, is defined in an
               entry.

               To determine if a particular object instance
               is in a particular MIB view, compare the
               object instance's OBJECT IDENTIFIER with each
               of the MIB view's entries.  If none match,
               then the object instance is not in the MIB
               view.  If one or more match, then the object
               instance is included in, or excluded from,
               the MIB view according to the value of
               viewType in the entry whose value of
               viewSubtree has the most sub-identifiers.  If
               multiple entries match and have the same
               number of sub-identifiers, then the
               lexicographically greatest instance of
```

```
          viewType determines the inclusion or
          exclusion.

          An object instance's OBJECT IDENTIFIER X
          matches an entry when the number of sub-
          identifiers in X is at least as many as in
          the value of viewSubtree for the entry, and
          each sub- identifier in the value of
          viewSubtree matches its corresponding sub-
          identifier in X.  Two sub identifiers match
          either if the corresponding bit of viewMask
          is zero (the 'wild card' value), or if they
          are equal.

           Due to this 'wild card' capability, we
          introduce the term, a 'family' of view
          subtrees, to refer to  the set of subtrees
          defined by a particular combination of values
          of viewSubtree and viewMask. In the case
          where no 'wild card' is defined in  viewMask,
          the family of view subtrees reduces to a
          single view subtree.

          Implementations must not restrict the number of
          families of view subtrees for a given MIB view,
          except as dictated by resource constraints on
          the overall number of entries in the
          viewTable.!!;
     ENDPARSE!;;
     ATTRIBUTES
          viewEntryId              GET,
          viewIndex                GET,
          viewSubtree              GET,
          viewMask
               DEFAULT VALUE
               IIMCRFC1447ASN1.c-DEFAULTNullString
               GET-REPLACE,
          viewType
               DEFAULT VALUE IIMCRFC1447ASN1.c-viewType
               GET-REPLACE,
          viewStorageType
               DEFAULT VALUE
               IIMCRFC1447ASN1.c-DEFAULTStorageType
```

```
                    GET-REPLACE,
               viewStatus               GET-REPLACE;;;
REGISTERED AS  {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1};


-- 4.1.2  Party MIB Attribute Types

party ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
```



```
          partyBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the type defined in
               [27] by the same name.!!;
          DESCRIPTION
               !!Denotes a SNMPv2 party identifier.  Note
               that agents may impose implementation
               limitations on the length of OIDs used to
               identify Parties. As such, management
               stations creating new parties should be aware
               that using an excessively long OID may result
               in the agent refusing to perform the set
               operation and instead returning the
               appropriate error response, e.g.,
               noCreation.!!;
          ENDPARSE!;;;

tAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          tAddressBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the type defined in
               [27] by the same name.!!;
          DESCRIPTION
               !!Denotes a transport service address.  For
               snmpUDPDomain, a TAddress is 6 octets long,
               the initial 4 octets containing the IP-
```

```
                address in network-byte order and the last 2
                containing the UDP port in network-byte
                order.  Consult [28] for further information
                on snmpUDPDomain.!!;
          ENDPARSE!;;;

clock ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:uInteger32;
     BEHAVIOUR
          clockBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
                !!This corresponds to the type defined in
                [27] by the same name.!!;
          DESCRIPTION
                !!A party's authentication clock - a non-
                negative integer which is incremented as
                specified/allowed by the party's
                Authentication Protocol.  For noAuth, a
```



```
                party's authentication clock is unused and
                its value is undefined.

                For v2md5AuthProtocol, a party's
                authentication clock is a relative clock with
                1-second granularity.!!;
          ENDPARSE!;;;

context ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
                !!This corresponds to the type defined in
                [27] by the same name.!!;
          DESCRIPTION
                !!Denotes a SNMPv2 context identifier. Note
                that agents may impose implementation
                limitations on the length of OIDs used to
                identify Parties. As such, management
```

```
                stations creating new parties should be aware
                that using an excessively long OID may result
                in the agent refusing to perform the set
                operation and instead returning the
                appropriate error response, e.g.,
                noCreation.!!;
           ENDPARSE!;;;


storageType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.StorageType;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          storageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the type defined in
               [27] by the same name.!!;
          DESCRIPTION
               !!Describes the memory realization of a
               conceptual row.  A row which is volatile(2)
               is lost upon reboot.  A row which is
               nonVolatile(3) is backed up by stable
               storage.  A row which is permanent(4) cannot
               be changed nor deleted.!!;
          ENDPARSE!;;;


-- 4.1.3  Party MIB Attributes
```



```
aclEntryId ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.AclEntryIdValue;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          aclEntryIdBehaviour BEHAVIOUR
          DEFINED AS
          !The naming attribute for object class aclEntry!;;
REGISTERED AS {iimcAutoName 1 3 6 1 6 3 3 2 3 1 1};

aclPrivileges ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.AclPrivileges;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
```

```
          aclPrivilegesBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The access privileges which govern what
               management operations a particular target party
               may perform with respect to a particular SNMPv2
               context when requested by a particular subject
               party.  These privileges are specified as a sum of
               values, where each value specifies a SNMPv2 PDU
               type by which the subject party may request a
               permitted operation.  The value for a particular
               PDU type is computed as 2 raised to the value of
               the ASN.1 context-specific tag for the appropriate
               SNMPv2 PDU type.  The values (for the tags defined
               in [28]) are defined in [25] as:

                              Get           :   1
                              GetNext       :   2
                              Response      :   4
                              Set           :   8
                              unused        :  16
                              GetBulk       :  32
                              Inform        :  64
                              SNMPv2-Trap   : 128
               The null set is represented by the value
               zero.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1 4};

aclResources ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclResourcesBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
```



```
               !!This corresponds to the object type efined
               in [27] by the same name.!!;
```

```
          DESCRIPTION
               !!The value of an instance of this object
               identifies a SNMPv2 context in an access
               control policy, and has the same value as the
               instance of the contextIndex object for that
               SNMPv2 context.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1 3};

aclStatus ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:rowStatus;
     BEHAVIOUR
          aclStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The status of this conceptual row in the
               aclTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1 6};

aclStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          aclStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The storage type for this conceptual row in
               the aclTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1 5};

aclSubject ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclSubjectBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The value of an instance of this object
```

```
                of an access control policy, and has the same
                value as the instance of the partyIndex object
                for that SNMPv2 party.!!;
           ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1 2};

aclTarget ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclTargetBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The value of an instance of this object
               identifies a SNMPv2 party which is the target
               of an access control policy, and has the same
               value as the instance of the partyIndex object
               for that party.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 3 1 1 1};

contextEntryId ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
IIMCRFC1447ASN1.ContextEntryIdValue;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          contextEntryIdBehaviour BEHAVIOUR
          DEFINED AS
          !The naming attribute for object class
          contextEntry!;;
REGISTERED AS {iimcAutoName 1 3 6 1 6 3 3 2 2 1 1};

contextIdentity ATTRIBUTE
     DERIVED FROM context;
     BEHAVIOUR
          contextIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
```

```
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!A context identifier uniquely identifying a
               particular SNMPv2 context.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 1};

contextIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR EQUALITY, ORDERING;
```



```
     BEHAVIOUR
          contextIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!A unique value for each SNMPv2 context. The
               value for each SNMPv2 context must remain
               constant at least from one re-initialization of
               the entity's network management system to the
               next re-initialization.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 2};

contextLocal ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:truthValue;
     BEHAVIOUR
          contextLocalBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!An indication of whether this context is
               realized by this SNMPv2 entity.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 3};
```

```
contextViewIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextViewIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!If the value of an instance of this object is
               zero, then this corresponding conceptual row in
               the contextTable refers to a SNMPv2 context which
               identifies a proxy relationship; the values of the
               corresponding instances of the
               contextProxyDstParty, contextProxySrcParty, and
               contextProxyContext objects provide further
               information on the proxy relationship.

               Otherwise, if the value of an instance of this
               object is greater than zero, then this
               corresponding conceptual row in the
```



```
               contextTable refers to a SNMPv2 context which
               identifies a MIB view of a locally accessible
               entity; the value of the instance identifies
               the particular MIB view which has the same
               value of viewIndex; and the value of the
               corresponding instances of the
               contextLocalEntity and contextLocalTime objects
               provide further information on the local entity
               and its temporal domain.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 4};

contextLocalEntity ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextLocalEntityBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
```

```
                !!This corresponds to the object type defined
                in [27] by the same name.!!;
            DESCRIPTION
                !!If the value of the corresponding instance of
                the contextViewIndex is greater than zero, then
                the value of an instance of this object
                identifies the local entity whose management
                information is in the SNMPv2 context's MIB
                view.  The empty string indicates that the MIB
                view contains the SNMPv2 entity's own local
                management information; otherwise, a non-empty
                string indicates that the MIB view contains
                management information of some other local
                entity, e.g.,'Repeater1'.!!;
            ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 5};

contextLocalTime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextLocalTimeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!If the value of the corresponding instance of
               the contextViewIndex is greater than zero, then
               the value of an instance of this object
               identifies the temporal context of the
               management information in the MIB view.!!;
          ENDPARSE!;;
```



```
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 6};

contextProxyDstParty ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          contextProxyDstPartyBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
```

```
                !!This corresponds to the object type defined
                in [27] by the same name.!!;
            DESCRIPTION
                !!If the value of the corresponding instance of
                the contextViewIndex is equal to zero, then the
                value of an instance of this object identifies a
                SNMPv2 party which is the proxy destination of a
                proxy relationship.

                If the value of the corresponding instance of
                the contextViewIndex is greater than zero, then
                the value of an instance of this object is
                zero.!!;
            ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 7};

contextProxySrcParty ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          contextProxySrcPartyBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
                !!This corresponds to the object type defined
                in [27] by the same name.!!;
          DESCRIPTION
                !!If the value of the corresponding instance of
                the contextViewIndex is equal to zero, then the
                value of an instance of this object identifies a
                SNMPv2 party which is the proxy source of a proxy
                relationship.

                Interpretation of an instance of this object
                depends upon the value of the transport domain
                associated with the SNMPv2 party used as the proxy
                destination in this proxy relationship.

                If the value of the corresponding instance of
                the contextViewIndex is greater than zero, then
                the value of an instance of this object is
                zero.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 8};

contextProxyContext ATTRIBUTE
```

```
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextProxyContextBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!If the value of the corresponding instance of
               the contextViewIndex is equal to zero, then the
               value of an instance of this object identifies the
               context of a proxy relationship.

               Interpretation of an instance of this object
               depends upon the value of the transport domain
               associated with the SNMPv2 party used as the proxy
               destination in this proxy relationship.

               If the value of the corresponding instance of
               the contextViewIndex is greater than zero, then
               the value of an instance of this object is { 0
               0 }.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 9};

contextStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          contextStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The storage type for this conceptual row in
               the contextTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 10};

contextStatus ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:rowStatus;
     BEHAVIOUR
          contextStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
```

```
                in [27] by the same name.!!;
          DESCRIPTION
                !!The status of this conceptual row in the
                contextTable.
```



```
                A context is not qualified for activation until
                instances of all corresponding columns have the
                appropriate value.  In  particular, if the
                context's contextViewIndex is greater than
                zero, then the viewStatus column of the
                associated conceptual row(s) in the viewTable
                must have the value `active'.  Until instances
                of all corresponding columns are appropriately
                configured, the value of the corresponding
                instance of the contextStatus column is
                `notReady'.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 2 1 1 11};

partyAuthClock ATTRIBUTE
     DERIVED FROM clock;
     BEHAVIOUR
          partyAuthClockBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
                !!This corresponds to the object type defined
                in [27] by the same name.!!;
          DESCRIPTION
                !!The authentication clock which represents the
                local notion of the current time specific to
                the party.  This value must not be decremented
                unless the party's secret information is
                changed simultaneously, at which time the
                party's nonce and last-timestamp values must
                also be reset to zero, and the new value of the
                clock, respectively.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 8};

partyAuthLifetime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.PartyAuthLifetime;
     MATCHES FOR      EQUALITY, ORDERING;
```

```
     BEHAVIOUR
          partyAuthLifetimeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The lifetime (in units of seconds) which
               represents an administrative upper bound on
               acceptable delivery delay for protocol messages
               generated by the party.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 11};
```



```
partyAuthPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCRFC1447ASN1.OctetString;
     MATCHES FOR      EQUALITY, SUBSTRINGS;
     BEHAVIOUR
          partypartyAuthPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type efined
               in [27] by the same name. It is modified to
               accommodate SNMPv1 community strings.!!;
          DESCRIPTION
               !!If the value of partyAuthProtocol is
               {snmpv1CommString} then this attribute
               contains the community string to be used with
               SNMPv1 security.

               If the value of partyAuthProtocol is not
               {snmpv1CommString} then this attribute
               contains an encoding of the party's private
               authentication  key which may be needed to
               support the authentication protocol.
               Although the value of  this variable may be
               altered by a management  operation (e.g., a
               SNMPv2 Set-Request), its value  can never be
               retrieved by a management operation:  when
               read, the value of this variable is the zero
```

```
                length OCTET STRING.

                The private authentication key is NOT directly
                represented by the value of this variable, but
                rather it is represented according to an
                encoding.  This encoding is the bitwise
                exclusive-OR of the old key with the new key,
                i.e., of the old private  authentication key
                (prior to the alteration) with  the new private
                authentication key (after the  alteration).
                Thus, when processing a received  protocol Set
                operation, the new private  authentication key
                is obtained from the value of  this variable as
                the result of a bitwise exclusive-OR of the
                variable's value and the old private
                authentication key.  In calculating the
                exclusive-OR, if the old key is shorter than
                the new key, zero-valued padding is appended to
                the old key.  If no value for the old key
                exists, a zero-length OCTET STRING is used in
                the calculation.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 9};

partyAuthProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
```



```
          IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partypartyAuthProtocolBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The authentication protocol by which all
               messages generated by the party are
               authenticated as to origin and integrity.  In
               this context, the value {noAuth } signifies
               that messages generated by the party are not
               authenticated.
```

```
                The value {snmpv1CommString} indicates that
                SNMPv1 community string is to be used.  The
                community string shall be present in
                partyAuthPrivate!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 7};

partyAuthPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.OctetString;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partyAuthPublicBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
                !!This corresponds to the object type defined
                in [27] by the same name.!!;
          DESCRIPTION
                !!A publicly-readable value for the party.
                Depending on the party's authentication
                protocol,this value may be needed to support
                the party's authentication protocol.
                Alternatively, it may be used by a manager
                during the procedure foraltering secret
                information about a party.  (For example, by
                altering the value of an instance of this
                object in the same SNMP Set-Request used to
                update an instance of partyAuthPrivate, a
                subsequent Get-Request can determine if the
                Set- Request was successful in the event that
                no response to the Set-Request is received,
                see RFC1446.)

                The length of the value is dependent on the
                party's authentication protocol.  If not used
                by the authentication protocol, it is
                recommended that agents support values of any
```



```
                length up to and including the length of the
                corresponding partyAuthPrivate object.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 10};

partyCloneFrom ATTRIBUTE
```

```
     DERIVED FROM party;
     BEHAVIOUR
          partyCloneFromBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The identity of a party to clone authentication
               and privacy parameters from.  When read, the value
               { 0 0 } is returned.

               This value can only be written when the
               associated instance of partyStatus either
               does not exist or has the value `notReady'.
               When written, the value identifies a party,
               the cloning party, whose status column has
               the value `active'.  The cloning party is
               used in two ways.

               One, if instances of the following objects do
               not exist for the party being created, then
               they are created with values identical to
               those of the corresponding objects for the
               cloning party:

                    partyAuthProtocol
                    partyAuthPublic
                    partyAuthLifetime
                    partyPrivProtocol
                    partyPrivPublic

               Two, instances of the following objects are
               updated using the corresponding values of the
               cloning party:

                    partyAuthPrivate
                    partyPrivPrivate

               (e.g., the value of the cloning party's
               instance of the partyAuthPrivate object is
               XOR'd with the value of the partyAuthPrivate
               instances of the party being created.)!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 15};

partyEntryId ATTRIBUTE
```

```
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.PartyEntryIdValue;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partyEntryIdBehaviour BEHAVIOUR
          DEFINED AS
          !The naming attribute for object class partyEntry!;;
REGISTERED AS {iimcAutoName 1 3 6 1 6 3 3 2 1 1 1};

partyIdentity ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          partyIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!A party identifier uniquely identifying a
               particular SNMP party.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 1};

partyIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!A unique value for each SNMPv2 party.  The
               value for each SNMPv2 party must remain
               constant at least from one re-initialization of
               the entity's network management system to the
               next reinitialization.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 2};

partyLocal ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:truthValue;
     BEHAVIOUR
          partyLocalBehaviour BEHAVIOUR
          DEFINED AS
```

```
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!An indication of whether this party
               executes at this SNMPv2 entity.  If this
```



```
               object has a value of true(1), then the
               SNMPv2 entity will listen for SNMPv2 messages
               on the partyTAddress associated with this
               party.  If this object has the value
               false(2), then the SNMPv2 entity will not
               listen for SNMPv2 messages on the
               partyTAddress associated with this party.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 6};

partyMaxMessageSize ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1447ASN1.PartyMaxMessageSize;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyMaxMessageSizeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The maximum length in octets of a SNMP
               message which this party will accept.  For
               parties which execute at an agent, the agent
               initializes this object to the maximum length
               supported by the agent, and does not let the
               object be set to any larger value.  For
               parties which do not execute at the agent,
               the agent must allow the manager to set this
               object to any legal value, even if it is
               larger than the agent can generate.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 5};

partyMIBObjectsId ATTRIBUTE
```

```
     WITH ATTRIBUTE SYNTAX
IIMCRFC1447ASN1.PartyMIBObjectsIdValue;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partyMIBObjectsIdBehaviour BEHAVIOUR
          DEFINED AS
          !The naming attribute for object class
partyMIBObjects!;;
REGISTERED AS {iimcAutoName 1 3 6 1 6 3 3 2};

partyPrivProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivProtocolBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
```



```
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The privacy protocol by which all protocol
               messages received by the party are protected
               from disclosure.  In this context, the value {
               noPriv } signifies that messages received by
               the party are not protected.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 12};

partyPrivPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!An encoding of the party's private
               encryption key which may be needed to support
               the privacy protocol.  Although the value of
```

```
                this variable may be altered by a management
                operation (e.g., a SNMPv2 Set-Request), its
                value can never be retrieved by a management
                operation: when read, the value of this
                variable is the zero length OCTET STRING.

                The private encryption key is NOT directly
                represented by the value of this variable,
                but rather it is represented according to an
                encoding. This encoding is the bitwise
                exclusive-OR of the old key with the new key,
                i.e., of the old private encryption key
                (prior to the alteration) with the new
                private encryption key (after the
                alteration). Thus, when processing a received
                protocol Set operation, the new private
                encryption key is obtained from the value of
                this variable as the result of a bitwise
                exclusive-OR of the variable's value and the
                old private encryption key.  In calculating
                the exclusive-OR, if the old key is shorter
                than the new key, zero-valued padding is
                appended to the old key.  If no value for the
                old key exists, a zero-length OCTET STRING is
                used in the calculation.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 13};

partyPrivPublic ATTRIBUTE
```



```
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivPublicBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!A publicly-readable value for the party.
               Depending on the party's privacy protocol,
               this value may be needed to support the
               party's privacy protocol.  Alternatively, it
```

```
                may be used by a manager as a part of its
                procedure for altering secret information
                about a party.  (For example, by altering the
                value of an instance of this object in the
                same SNMP Set-Request used to update an
                instance of partyPrivPrivate, a subsequent
                Get-Request can determine if the Set-Request
                was successful in the event that no response
                to the Set-Request is received, see RFC
                1446.)

                The length of the value is dependent on the
                party's privacy protocol.  If not used by the
                privacy protocol, it is recommended that
                agents support values of any length up to and
                including the length of the corresponding
                partyPrivPrivate object.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 14};

partyStatus ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:rowStatus;
     BEHAVIOUR
          partyStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The status of this conceptual row in the
               partyTable.

               A party is not qualified for activation until
               instances of all columns of its partyEntry
               row have an appropriate value.  In
               particular:

               A value must be written to the Party's
               partyCloneFrom object.
```



```
               If the Party's partyAuthProtocol object has
               the value md5AuthProtocol, then the
               corresponding instance of partyAuthPrivate
```

```
                must contain a secret of the appropriate
                length.  Further, at least one management
                protocol set operation updating the value of
                the party's partyAuthPrivate object must be
                successfully processed, before the
                partyAuthPrivate column is considered
                appropriately configured.

                If the Party's partyPrivProtocol object has
                the value desPrivProtocol, then the
                corresponding instance of partyPrivPrivate
                must contain a secret of the appropriate
                length.  Further, at least one management
                protocol set operation updating the value of
                the party's partyPrivPrivate object must be
                successfully processed, before the
                partyPrivPrivate column is considered
                appropriately configured.

                Until instances of all corresponding columns
                are appropriately configured, the value of
                the corresponding instance of the partyStatus
                column is `notReady'.!!;
           ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 17};

partyStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          partyStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The storage type for this conceptual row in
               the partyTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 16};

partyTAddress ATTRIBUTE
     DERIVED FROM tAddress;
     BEHAVIOUR
          partyTAddressBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
```

```
          DESCRIPTION
               !!The transport service address by which the
               party receives network management traffic,
               formatted according to the corresponding
               value of partyTDomain.  For rfc1351Domain,
               partyTAddress is formatted as a 4-octet IP
               Address concatenated with a 2-octet UDP port
               number.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 4};

partyTDomain ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partyTDomainBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!Indicates the kind of transport service by
               which the party receives network management
               traffic. An example of a transport domain is
               'rfc1351Domain' (SNMP over UDP).!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 1 1 1 3};

viewEntryId ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ViewEntryIdValue;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          viewEntryIdBehaviour BEHAVIOUR
          DEFINED AS
          !The naming attribute for object class viewEntry!;;
REGISTERED AS {iimcAutoName 1 3 6 1 6 3 3 2 4 1 1};

viewIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewIndexBehaviour BEHAVIOUR
```

```
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!A unique value for each MIB view.  The
               value for each MIB view must remain constant
               at least from one re-initialization of the
```



```
               entity's network management system to the
               next re-initialization.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1 1};

viewMask ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1447ASN1.ViewMask;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewMaskBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The bit mask which, in combination with the
               corresponding instance of viewSubtree, defines a
               family of view subtrees.

               Each bit of this bit mask corresponds to a
               sub-identifier of viewSubtree, with the most
               significant bit of the i-th octet of this
               octet string value (extended if necessary,
               see below) corresponding to the (8*i - 7)-th
               sub-identifier, and the least significant bit
               of the i-th octet of this octet string
               corresponding to the (8*i)-th sub-identifier,
               where i is in the range 1 through 16.

               Each bit of this bit mask specifies whether
               or not the corresponding sub-identifiers must
               match when determining if an OBJECT
```

```
                IDENTIFIER is in this family of view
                subtrees; a '1' indicates that an exact match
                must occur; a '0' indicates 'wild card',
                i.e., any sub-identifier value matches. Thus,
                the OBJECT IDENTIFIER X of an object instance
                is contained in a family of view subtrees if
                the following criteria are met:

                for each sub-identifier of the value of
                viewSubtree, either:

                     the i-th bit of viewMask is 0, or

                     the i-th sub-identifier of X is equal to  the
                i-th sub-identifier of
                     the value of  viewSubtree.

                If the value of this bit mask is M bits long
                and there are more than M sub-identifiers in
                the corresponding instance of viewSubtree,
```



```
                then the bit mask is extended with 1's to be
                the required length.

                Note that when the value of this object is
                the zero-length string, this extension rule
                results in a mask of all-1's being used
                (i.e., no 'wild card'), and the family of
                view subtrees is the one view subtree
                uniquely identified by the corresponding
                instance of viewSubtree.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1 3};

viewStatus ATTRIBUTE
     DERIVED FROM {iimcIIMCIMIBTRANS}:rowStatus;
     BEHAVIOUR
          viewStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The status of this conceptual row in the
```

```
               viewTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1 6};

viewStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          viewStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
               !!The storage type for this conceptual row in
               the viewTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1 5};

viewSubtree ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewSubtreeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
               in [27] by the same name.!!;
          DESCRIPTION
```



```
               !!A MIB subtree.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1 2};

viewType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1447ASN1.ViewType;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE
               !!This corresponds to the object type defined
```

```
               in [27] by the same name.!!;
          DESCRIPTION
               !!The status of a particular family of view
               subtrees within the particular SNMPv2
               context's MIB view.  The value 'included(1)'
               indicates that the corresponding instances of
               viewSubtree and viewMask define a family of
               view subtrees included in the MIB view.  The
               value 'excluded(2)' indicates that the
               corresponding instances of viewSubtree and
               viewMask define a family of view subtrees
               excluded from the MIB view.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoObjAndAttr 1 3 6 1 6 3 3 2 4 1 1 4};



-- 4.1.4  Party MIB Name Bindings

aclEntry-partyMIBObjectsNB  NAME BINDING
          SUBORDINATE OBJECT CLASS  aclEntry  AND SUBCLASSES;
          NAMED BY SUPERIOR OBJECT CLASS  partyMIBObjects  AND
SUBCLASSES;
     WITH ATTRIBUTE  aclEntryId;
          BEHAVIOUR
          aclEntry-partyMIBObjectsNBBehaviour     BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX     SNMPv2-Party-MIB.aclTarget,
                    SNMPv2-Party-MIB.aclSubject,
                    SNMPv2-Party-MIB.aclResources;
               DELETEATT aclStatus;
               DELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoNameBinding 1 3 6 1 6 3 3 2 3 1 1};

contextEntry-partyMIBObjectsNB  NAME BINDING
   SUBORDINATE OBJECT CLASS  contextEntry  AND SUBCLASSES;
```



```
   NAMED BY SUPERIOR OBJECT CLASS  partyMIBObjects  AND
SUBCLASSES;
   WITH ATTRIBUTE  contextEntryId;
```

```
          BEHAVIOUR
          contextEntry-partyMIBObjectsNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX IMPLIED SNMPv2-Party-MIB.contextIdentity;
               DELETEATT contextStatus;
               DELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoNameBinding 1 3 6 1 6 3 3 2 2 1 1};

partyEntry-partyMIBObjectsNB  NAME BINDING
          SUBORDINATE OBJECT CLASS  partyEntry AND SUBCLASSES;
          NAMED BY SUPERIOR OBJECT CLASS  partyMIBObjects  AND
SUBCLASSES;
     WITH ATTRIBUTE  partyEntryId;
          BEHAVIOUR
          partyEntry-partyMIBObjectsNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX IMPLIED SNMPv2-Party-MIB.partyIdentity;
               DELETEATT partyStatus;
               DELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoNameBinding 1 3 6 1 6 3 3 2 1 1 1};

viewEntry-partyMIBObjectsNB  NAME BINDING
          SUBORDINATE OBJECT CLASS  viewEntry AND SUBCLASSES;
          NAMED BY SUPERIOR OBJECT CLASS  partyMIBObjects  AND
SUBCLASSES;
          WITH ATTRIBUTE viewEntryId;
          BEHAVIOUR
          viewEntry-partyMIBObjectsNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX     SNMPv2-Party-MIB.viewIndex,
                    IMPLIED SNMPv2-Party-MIB.viewSubtree;
          DELETEATT viewStatus;
               DELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoNameBinding 1 3 6 1 6 3 3 2 4 1 1};

partyMIBObjects-systemNB  NAME BINDING
```

```
          SUBORDINATE OBJECT CLASS partyMIBObjects AND
SUBCLASSES;
          NAMED BY SUPERIOR OBJECT CLASS
          "Rec. X.721 | ISO/IEC 10165-2 : 1992":system  AND
SUBCLASSES;
          WITH ATTRIBUTE  partyMIBObjectsId;
          BEHAVIOUR
          partyMIBObjects-systemNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX     NULL;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoNameBinding 1 3 6 1 6 3 3 2};



-- 4.2  PARTY MIB ASN.1 MODULES

IIMCRFC1447ASN1
{iso(1) member-body(2) 124 forum(360501) iimcAutoTrans(14)
iimcAutoModule(0) 1447}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
          UInteger32, snmpv2
               FROM SNMPv2-SMI
          partyAdmin, partyProtocols, noAuth, noPriv,
          desPrivProtocol, v2md5AuthProtocol,
          temporalDomains, currentTime, restartTime,
          cacheTime, initialPartyId,
          initialContextId
               FROM SNMPv2-Party-MIB
          snmpUDPDomain
               FROM SNMPv2-TM
          iimcAutoModule, iimcAutoObjAndAttr,
          iimcAutoNameBinding,
          iimcAutoDocument, iimcAutoName, iimcIIMCIMIBTRANS,
               FROM IimcAssignedOIDs
               {iso(1) member-body(2) 124 forum(360501)
               iimcManual(15) iimcModule(0) 1}
          Integer, OctetString, ObjectIdentifier
               FROM IimcCommonDef
```

```
               {iso(1) member-body(2) 124 forum(360501)
               iimcManual(15) iimcModule(0) 2};

iimcRFC1447 OBJECT IDENTIFIER ::= {iimcAutoDocument 1447}

AclPrivileges ::= INTEGER (0..255)

AclEntryIdValue     ::= SEQUENCE   {
                         aclTarget [1] Index,
                         aclSubject     [2] Index,
```



```
                         aclResources   [3] Index
                                   }

Clock     ::= UInteger32

ContextEntryIdValue ::= SEQUENCE   {
                    contextIdentity     [1] OBJECT IDENTIFIER
                              }

Index     ::= INTEGER (1..65535)

PartyAuthLifetime   ::= INTEGER (0..2147483647)

PartyEntryIdValue   ::= SEQUENCE   {
                         partyIdentity  [1] OBJECT IDENTIFIER
                              }

PartyMIBObjectsIdValue   ::= NULL

PartyMaxMessageSize ::= INTEGER (484..65507)

StorageType ::= INTEGER {
   other(1),       -- eh?
   volatile(2),    -- e.g., in RAM
   nonVolatile(3), -- e.g., in NVRAM
   permanent(4)    -- e.g., in ROM
   }

ViewEntryIdValue    ::= SEQUENCE   {
                    viewIdentity   [1] OBJECT IDENTIFIER
                         }

ViewMask  ::= OCTET STRING (SIZE (0..16))
```

```
ViewType ::= INTEGER  {
    included(1),
    excluded(2)
                      }


-- Default value constants

c-aclPrivileges          INTEGER ::= 35
c-contextLocal           BOOLEAN ::= TRUE
c-DEFAULTNullString      OCTET STRING ::= ''H
c-contextLocalTime       Clock ::= currentTime
c-DEFAULTStorageType     INTEGER ::= 3
c-partyTDomain           OBJECT IDENTIFIER ::= snmpUDPDomain
c-partyTAddress          OCTET STRING  ::= '000000000000'H
c-partyMaxMessageSize    INTEGER ::= 484
c-partyLocal             BOOLEAN ::= FALSE
c-partyAuthProtocol      OBJECT IDENTIFIER ::= v2md5AuthProtocol
c-partyAuthClock         INTEGER ::= 0
c-partyAuthLifetime      INTEGER ::= 300
```



```
c-partyPrivProtocol      OBJECT IDENTIFIER ::= noPriv
c-viewType               INTEGER ::= 1

END
```

5. IIMC ACL MIB

The use of parties and contexts and community strings can be very confusing for application programmers. Also the actual privileges associated with an individual user of an application are not generally at the discretion of the user or programmer, but are at the discretion of the person responsible for enforcing the security policy, i.e., configuring the security MIB elements. The actual party identities and associated contexts, or community strings that the user needs for access could remain hidden from the user - and perhaps should.

A mechanism for hiding most of the assignment and

configuration of security parameters associated with user security privileges for proxy/agent communications is to implement an access control list (ACL) scheme at the proxy. The ACL scheme allows an identity to be specified with the CMIP request, or ACSE association, have it authenticated, and on the basis of that authenticated identity be assigned the context and source/destination party pairs, or community string, that grants or denies them access to specific operations on specific objects associated with specific managed systems. The actual association between the identity and the party/context or community string shall be accomplished by configuring the security management parameters within the aclSecurityInfoEntry objects of the proxy system.

The information for the access control list for ISO/CCITT-Internet shall be maintained in a table of entries (aclSecurityInfoEntry) that contain:

* an <identity> associated with a user, application, or role,

* the SNMP agent identification, and

* the associated party/context or community string information needed to determine, from other elements in the MIB, the security and communication parameters required to communicate with the remote SNMP agent.

The <identity> shall be provided in the access control certificate (ACC) used for security services between the ISO/CCITT manager and the proxy. Therefore, no additional information needs to be delivered in a contained ACC that holds security parameters for proxy to SNMP agent communications. It is a local security policy matter whether the SNMP security parameters delivered in an ACC over an association, or in a CMIP PDU, will have precedence over those in the aclSecurityInfoEntry's parameters.

The naming attribute of the aclSecurityInfoEntry shall contain a sequence of the <identity> (an octet string) and the <agent id>. The <agent id> shall be the same format as that specified for the cmipsnmpProxyAgentId attribute of the cmipsnmpProxyAgent object class [31].

The proxy shall determine the transport address of the SNMP agent from data in the Proxy MIB [31], i.e., in the cmipsnmpProxyAgent entry that contains the indicated <agent id>. It shall then formulate the combination of identity and transport address necessary to find the party/context/community string in a aclSecurityInfoEntry.

The combination of SNMP agent transport address and community string provides sufficient information to communicate with an SNMPv1 agent.

The partyEntry and contextEntry associated with the parties and context found in the aclSecurityInfoEntry provide the communication and security parameters necessary to communicate with the SNMPv2 agent.

For efficiency, this mechanism has been adapted for use with defined parties and contexts in the UDP domain that are derived from the IP address of the agent, as described in [27].

To accommodate defined UDP contexts and parties, the iimcAclIdentity naming attribute shall be allowed to have values that do not contain the "<agent Id>" component. The naming attribute shall also be allowed to have the associated party and context OIDs of the form for the default party {initialPartyId 0 0 0 0 x} and default context {initialContextId 0 0 0 0 x} as defined in [27]. The string of "0"s are to be interpreted as the OID fragment representation of the IP address. The x assumes the value as defined in [27].

For example the aclSecurityInfoEntry below, associated with "John Doe" provides for access to any SNMPv2 device in the snmpUDPDomain conforming to [27], that uses the defaults for mD5Auth/noPriv and allows Get, Get-Next, Set, and Get-Bulk operations.

```
iimcAclIdentity              =    "John Doe"

aclTarget           =    {initialPartyId 0 0 0 0 3 }
aclSubject          =    {initialPartyId 0 0 0 0 4 }
aclResources        =    {initialContextId 0 0 0 0 2 }
snmpv1CommunityString    =    ''H
```

The proxy can determine the actual party and context OIDs by filling in the "0 0 0 0" portion of the OIDs with the agents IP address.

It is possible for "John Doe" to have different privileges for different SNMP agents. Other entries could exist for "John Doe" that are specific to the agent. Therefore, a procedure must be established for processing entries in the table. The procedure for processing the table of aclSecurityInfoEntry shall be:

1) check for an entry that may be specific to an agent;

2) if an agent specific entry is not found, then check for an entry for a generic agent.

The above technique avoids the necessity for creating a table entry for every party pair and context for each agent. It is also possible to extend the generic policies by using the same technique defined in [27].

The objects, attributes, and name bindings for an access control list scheme within a proxy are defined below. Support for this access control list scheme shall require the proxy to instantiate the partyEntry and the contextEntry managed objects for communicating with SNMP agents.

A Naming Tree diagram for the ACL related managed object classes is illustrated below. The aclSecurityInfoEntry is subordinate to the ISO/CCITT system managed object that represents the proxy.

```
"Rec. X.721 | ISO/IEC 10165-2 : 1992" : system
     |
     |--- aclSecurityInfoEntry
```

This document (IIMCSEC) is allocated the following registration identifier for purposes of referencing the ACL MIB contained herein.

```
    iimcIIMCACLMIB OBJECT IDENTIFIER ::= { iimcDocument 4 }
```

```
-- 5.1 IIMC ACL MIB GDMO TEMPLATES
```

```
-- 5.1.1 IIMC ACL MIB Managed Object Classes

aclSecurityInfoEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
```



```
     aclSecurityInfoEntryPkg PACKAGE
     BEHAVIOUR
          aclSecurityInfoEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          DESCRIPTION
               !!The security information for a particular
               requesting identity when communicating with a
               particular SNMP agent.  The identity is
               associated with either the specific SNMP
               source party, destination party, and context
               triplet, or the community string to be used
               for the communication between the proxy and
               the SNMP agent.

               Attributes for this object are read only.

               When created, only the security information
               specific to the SNMP protocol version needs
               to be provided.  The security parameters for
               the version not chosen are set to default
               values indicating that they are not
               applicable for communicating with the SNMP
               agent.!!;
          ENDPARSE!;;
               ATTRIBUTES
               iimcAclIdentity                         GET,
               {iimcRFC1447}:aclTarget
               DEFAULT VALUE IIMCACLMIBASN1.c-DEFAULToidNA
                                              GET,

               {iimcRFC1447}:aclSubject
               DEFAULT VALUE IIMCACLMIBASN1.c-DEFAULToidNA
                                              GET,

               {iimcRFC1447}:aclResources
               DEFAULT VALUE IIMCACLMIBASN1.c-DEFAULToidNA
                                              GET,
```

```
               snmpv1CommunityString
               DEFAULT VALUE IIMCACLMIBASN1.c-DEFAULTNullString
                                              GET;;;
REGISTERED AS  {iimcObjectClass 5};


-- 5.1.2 IIMC ACL MIB Attributes

iimcAclIdentity ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCACLMIBASN1.IimcAclIdentity;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          iimcAclIdentityBehaviour BEHAVIOUR
          DEFINED AS
```



```
          !The value of an instance of this object is an
          identity that is associated either with a specific
          source party, destination party, and context
          triplet, or a community string.  It may optionally
          include the specific SNMP agent identity, where
          the agent identity shall be the same format as is
          specified for the
          {iimcIIMCProxy}:cmipsnmpProxyAgentId attribute
          defined in [31].!;;
REGISTERED AS {iimcAttribute 18};

snmpv1CommunityString ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCACLMIBASN1.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          snmpv1CommunityStringBehaviour BEHAVIOUR
          DEFINED AS
          !The value of an instance of this object is an
          SNMPv1 community string that is associated with
          the remote SNMP agent.!;;
REGISTERED AS {iimcAttribute 19};



-- 5.1.3 IIMC ACL MIB Name Bindings

aclSecurityInfoEntry-systemNB  NAME BINDING
       SUBORDINATE OBJECT CLASS aclSecurityInfoEntry AND
          SUBCLASSES ;
```

```
     NAMED BY SUPERIOR OBJECT CLASS
          "Rec. X.721 | ISO/IEC 10165-2 : 1992":system AND
          SUBCLASSES;
     WITH ATTRIBUTE iimcAclIdentity;
     CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                    WITH-REFERENCE-OBJECT;
     DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcNameBinding 4};
```



```
-- 5.2 IIMC ACL MIB ASN.1 MODULES

IIMCACLMIBASN1
{iso(1) member-body(2) 124 forum(360501) iimcManual(15)
iimcModule(0) 4}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
          iimcModule, iimcDocument, iimcObjectClass,
          iimcAttribute,
          iimcNameBinding, iimcIIMCACLMIB
               FROM IimcAssignedOIDs
               {iso(1) member-body(2) 124 forum(360501)
               iimcManual(15) iimcModule(0) 1}
          OctetString
               FROM IimcCommonDef
               {iso(1) member-body(2) 124 forum(360501)
               iimcManual(15) iimcModule(0) 2}
          CmipsnmpProxyAgentId
               FROM IimcProxyASN1
               {iso(1) member-body(2) 124 forum(360501)
```

```
               iimcManual(15) iimcModule(0) 3};

IimcAclIdentity     ::= SEQUENCE   {
                    identity  OCTET STRING,
                    agentId   CmipsnmpProxyAgentId
                    OPTIONAL}

c-DEFAULToidNA           OBJECT IDENTIFIER ::= {0 0}
c-DEFAULTNullString      OCTET STRING ::= ''H

END
```


6. CONFORMANCE

An implementation claiming conformance to this document:

(a) shall conform the to translated ISO/CCITT GDMO Party MIB {iimcRFC1447} requirements stated in the corresponding MOCS proforma specified by Annex A;

(b) shall optionally conform to all of the ISO Manager to

ISO/CCITT Internet Proxy security requirements stated in section 3.1, in the agent role;

(c) shall support all of the ISO/CCITT-Internet Proxy to Internet Agent security requirements stated in section 3.2, in the manager role; and

(d) shall optionally conform to the aclSecurityInfoEntry class requirements stated in the corresponding MOCS proforma specified by Annex A.

ANNEX A (NORMATIVE): MANAGED OBJECT CONFORMANCE STATEMENTS (MOCS)

Class	Status	Support
partyEntry	m	
contextEntry	m	
viewEntry	c1	
aclEntry	c1	
aclInfoEntry	o	

c1: - if ISO/CCITT-Internet Proxy implementation, else m

This section available only in Postscript Format.

ANNEX B: GLOSSARY

ACC	Access Control Certificate
ACL	Access Control List
ACSE	Association Control Service Element
ASN.1	Abstract Syntax Notation One
CCITT	Consultative Committee on Telephony and Telegraphy
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
DN	Distinguished Name
GDMO	Guidelines for the Definition of Managed Objects
GNMP	Government Network Management Profile
IIMC	ISO/CCITT and Internet Management Coexistence
ISO	International Standards Organization
MD5	Message Digest 5
MIB	Management Information Base
MOCS	Managed Object Conformance Statement
NMF	Network Management Forum
OID	Object Identifier
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RDN	Relative Distinguished Name
RFC	Request For Comments
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SNMPv1	Simple Network Management Protocol Version 1
SNMPv2	Simple Network Management Protocol Version 2
TCP/IP	Transmission Control Protocol/Internetwork Protocol

ANNEX C: REFERENCES

1) CCITT Recommendation X.700, Management Framework Definition for Open Systems Interconnection (OSI).

ISO/IEC 7498-4: 1989, Information Processing Systems -- Open Systems Interconnection -Basic Reference Model Part 4 -- Management Framework.

2) ISO/IEC 8824: Information Technology -- Open System Interconnection -- Specification of Abstract Syntax Notation One (ASN.1),1990.

3) CCITT Recommendation X.209 (1988), Specification of basic encoding rules for abstract syntax notation one (ASN.1).

ISO/IEC 8825: 1990, Information Technology -- Open System Interconnection -- Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

4) CCITT Recommendation X.500 | ISO/IEC 9594, Information Technology - Open System Interconnection - The Directory - Parts 1-8.

5) CCITT Recommendation X.710, (1991), Common Management Information Service Definition for CCITT Applications.

ISO/IEC 9595: 1991, Information Technology -- Open System Interconnection -- Common Management Information Service Definition.

6) CCITT Recommendation X.711 | ISO/IEC 9596-1: 1991, Information Technology -- Open Systems Interconnection -- Common Management Information Protocol -- Part 1: Specification.

7) CCITT Recommendation X.733 (1992) | ISO/IEC DIS 10164-9, Information Technology -- Open Systems Interconnection -- Systems Management -- Part 9: Objects and Attributes for

Access Control, ISO/IEC JTC1/SC21/N7661, March, 1993.

8) CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1: 1992, Information Technology -- Open Systems Interconnection -- Structure of Management Information -- Part 1: Management Information Model.

9) CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2: 1992, Information Technology -- Open Systems Interconnection -- Structure of Management Information -- Part 2: Definition of Management Information.

10) CCITT Recommendation X.721 (1992) | ISO/IEC 10165-4: 1992, Information Technology -- Open Systems Interconnection -- Structure of Management Information -- Part 4: Guidelines for the Definition of Managed Objects.

11) CCITT Recommendation X.723 (1993) | ISO/IEC 10165-6: 1993, Information Technology -- Open Systems Interconnection -- Structure of Management Information -- Part 6: Requirements and Guidelines for Implementation Conformance Statement Proformas associated with OSI Management.

12) ISO/IEC DIS 10181-3, Information Technology , OSI Security Model, Part 3: Access Control Framework, 1993.

13) ISO/IEC CD 11586-1, Information Technology - Generic Upper Layers Security - Part 1: Overview, Models and Notation, November 1992.

14) ISO/IEC CD 11586-2, Information Technology - Generic Upper Layers Security - Part 2: Security Exchange Service Element(SESE) Service Definition, November 1992.

15) ISO/IEC CD 11586-3, Information Technology -Generic Upper Layers Security - Part 3: Security Exchange Service Element(SESE) Protocol Specification, November 1992.

16) ISO/IEC CD 11586-4, Information Technology - Generic Upper Layers Security - Part 4: Protecting Transfer Syntax Specification, November 1992.

17) NIST Special Publication 500-206, Stable Implementation

Agreements for Open Systems Interconnection Protocols, Version 6, Edition 1, December 1992

18) RFC1155, M. Rose and K. McCloghrie, Structure and Identification of Management Information for TCP/IP based internets, May 1990.

19) RFC1157, J.D. Case, M.S. Fedor, M.L. Schoffstall,C. Davin, Simple Network Management Protocol (SNMP), May 1990.

20) RFC1212, M. Rose, K. McCloghrie -- Editors, Concise MIB Definitions, March 1991.

21) RFC1213, K. McCloghrie and M. Rose -- Editors, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991.

22) RFC1215, M. Rose -- Editor, A convention for Defining Traps for use with the SNMP, March 1991.

23) RFC1441, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Introduction to version 2 of the Internet-standard Network Management Framework, April 1993.

24) RFC1442, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

25) RFC1445, J.R. Davin, J.M. Galvin, K.McCloghrie, Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

26) RFC1446, J.M. Galvin, K. McCloghrie, J.R. Davin, Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

27) RFC1447, J.D. Case, K. McCloghrie, M.T. Rose, S.L. Waldbusser, Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

28) RFC1448, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

29) RFC1452, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Coexistence between version 1 and version 2 of the Internet Network Management Framework, April 1993.

30) Network Management Forum: Forum 029, Translation of Internet MIB-II (RFC 1213) to ISO/CCITT GDMO MIB, Issue 1.0, October 1993.

31) Network Management Forum: Forum 028, ISO/CCITT to Internet Management Proxy, Issue 1.0, 1993.

32) Network Management Forum: Forum 026, Translation of Internet MIBs to ISO/CCITT GDMO MIBs, Issue 1.0, October 1993.

33) Network Management Forum: Forum 030, Translation of ISO/CCITT GDMO MIBs to Internet MIBs, Issue 1.0, October 1993.

34) Network Management Forum: Forum 016, Application Services: Security of Management, Issue 1.0, August, 1992.

35) NM Forum and X/Open, ISO/CCITT and Internet Management: Coexistence and Interworking Strategy, Issue 1.0, October, 1992.

36) ECMA-138, Security in Open Systems: Data Elements and Service Definitions, December 1989.

37) Federal Information Processing Standards Publication 179 -- Government Network Management Profile v1.0, December 1992.

38) ISO/IEC 7498-2, Information Processing Systems -- Open Systems Interconnection -Basic Reference Model Part 2 -- Security Architecture.