The ATM Forum

Technical Committee

ATM Connection Filtering MIB and Audit Log

AF-SEC-0188.000
July 2002

The ATM Forum1`
P.O. Box 29920
572 B Ruger ST
San Francisco, CA 94129-0920
Tel: +1 415 561 6275
Fax: +1 415 561 6120

1.0 Introduction

This document defines a Management Information Base (MIB) and audit log. The purpose of the MIB is to provide a standard mechanism to manage ATM network elements capable of filtering ATM SETUP messages based on security criteria. Each ATM network element processing a SETUP message will have the option to discard the SETUP message if it does not pass the security filters within the MIB.

This specification also defines an audit log that can be used by a node to record various events that may be of interest to a security administrator. The audit log is protected by a digital signature to prevent tampering.

This MIB and audit log are intended to be used with a secure network management strategy. Many objects have the access of read-write or read-create assigned to them. Making these objects writable from a remote management station requires authentication so only authorized managers access the node and access control so that only authorized network administrators can change security parameters. These services are out of scope for this document, but are defined in AF-SEC-0179.000. Without these security services, unauthorized users may change the security configuration of a node to bypass security filtering.

1.1 Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
BLLI	Broadband Lower Layer Information
CBR	Constant Bit Rate
DTL	Designated Transit List
DSA	Digital Signature Algorithm
MAC	Message Authentication Code
MIB	Management Information Base
OAM	Operations, Administration and Maintenance
OUI	Organizationally Unique Identifier
SHA	Secure Hash Algorithm
SME	Security Message Exchange
SSIE	Security Services Information Element
UBR	Unspecified Bit Rate
VBR	Variable Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Indicator
VP	Virtual Path
VPI	Virtual Path Indicator

1.2 References

[1] U.S. Department of Commerce, National Institute of Standards and Technology, “Digital Signature Standard (DSS) ”, FIPS PUB 186-1, December 15, 1998.

[2] U.S. Department of Commerce, National Institute of Standards and Technology, “Secure Hash Standard”, FIPS PUB 180-1, April 17, 1995.

[3] ATM Forum Technical Committee, “ATM Forum Security Specification”, Version 1.1, AF-SEC-0100.002, October 2002.

2.0 Description of Connection Filtering

The connection filtering MIB allows an ATM network element to filter incoming ATM SETUP messages. This feature allows an ATM network element to build communities of interest by only allowing authorized users to communicate. This capability also allows an ATM network element to enforce a policy that limits users to authorized services and privileges.

The connection filtering MIB allows ATM network elements to filter on:

- ?? **Source ATM Address** – Provides the capability to allow or disallow a connection originating from a particular source address or range of addresses.
- ?? **Outgoing Port** – Provides the capability to allow or disallow a connection to be routed through an outgoing port based upon the source address.
- ?? **Destination ATM Address** – Provides the capability to allow or disallow a connection to a particular destination address or range of addresses based upon the source address.
- ?? **ATM Adaptation Layer (AAL) Type** – Provides the capability to allow or disallow a connection using a particular AAL type based upon the source address.
- ?? **Addresses Present in the Designated Transit List (DTL)** – Provides the capability to allow or disallow a connection based upon where the connection will be routed based upon the source address.
- ?? **Traffic Contract** – Provides the capability to allow or disallow a connection requesting Unspecified Bit Rate (UBR), Constant Bite Rate (CBR), Variable Bit Rate (VBR), or Available Bit Rate (ABR) service based on the source address.
- ?? **Maximum Peak Cell Rate** – Provides the capability to disallow a connection based upon the maximum peak cell rate requested for the connection. It is permissible for a vendor to allow filtering on additional QoS parameters.
- ?? **Broadband Lower Layer Information (BLLI) Codepoints** – Provides the capability to disallow a connection based upon the BLLI codepoints present in the setup message based upon the source address.
- ?? **Encapsulation Type –** Provides the capability to allow or disallow a connection using a particular encapsulation type based on the source address.
- ?? **Virtual Path (VP) or Virtual Channel (VC) merge –** Provide the capability to allow or disallow VP or VC merge for a connection for a particular source address.
- ?? **Start and Stop Time** – Provides the capability to specify a time window that a source address can establish a connection through the ATM network element.

Additionally, the MIB controls:

1) Which signaling messages and Private Network to Network Interface (PNNI) messages are written to the audit log.
2) Whether Security Message Exchange (SME) is audited.

3.0 ATM Connection Filtering and Auditing MIB

```
ATM-FILTERING-MIB DEFINITIONS  : := BEGIN

     IMPORTS
               Integer32, Octet String
                    From SNMPv2-SMIv2
               Truthvalue
                    From SNMPv2-TC
               Integer
                    From SMIv1

ATM-FILTERING-MIB-IDENTITY
     LAST-UPDATED        0204250000Z
     ORGANIZATION        "The ATM Forum"
     CONTACT-INFO
          "The ATM Forum
          Worldwide Headquarters
          P.O. Box 29920
          572 B Ruger St
          San Francisco, CA 94129-0920
          Tel:  +1 415 561 6275
          Fax:  +1 415 561 6120
          Info@atmforum.com"
     DESCRIPTION
          "The MIB module for performing security related
          filtering on ATM connections"
     REVISION            0204250000Z    April 25, 2002
     DESCRIPTION
          "Objects for management of ATM security filtering."
          :: = {atmseconfilMIB}

--   The object identifier subtree for ATM Forum security
filtering MIBs.

atmForum  OBJECT IDENTIFIER :: = { enterprise 353 }
atmForumNetworkManagement  OBJECT IDENTIFIER :: = { atmForum 5 }
atmFsec  OBJECT IDENTIFIER :: = { atmForumNetworkManagement 12 }

atmseconfilMIBObjects  OBJECT IDENTIFIER :: = { atmseconfilMIB 1 }
```

```
AtmAddr  : := TEXTUAL-CONVERSION
     STATUS  current
     DESCRIPTION
          "The ATM address used by the network entity.  The
          address types are:  no address (0 octets), and NSAP
          (20 octets)."
     SYNTAX    OCTET STRING (SIZE(0|20))

AtmseconfilBaseGroup OBJECT IDENTIFIER :: = { atmseconfilMIBObjects 1 }

AuditsigKeyp        OBJECT-TYPE
     SYNTAX         Octet String (size 128)
     MAX-ACCESS          read-only
     STATUS         current
     DESCRIPTION
          "This object holds the p value of the DSA signature
          key in binary format using a character field.  The DSA
          signature values are defined in [1]."
     ::= { atmseconfilBaseGroup 1 }

AuditsigKeyq        OBJECT-TYPE
     SYNTAX         Octet String (size 20)
     MAX-ACCESS          read-only
     STATUS         current
     DESCRIPTION
          "This object holds the q value of the DSA signature
          key in binary format using a character field.  The DSA
          signature values are defined in [1]."
     ::= { atmseconfilBaseGroup 2 }

AuditsigKeyg        OBJECT-TYPE
     SYNTAX         Octet String (size 128)
     MAX-ACCESS          read-only
     STATUS         current
     DESCRIPTION
          "This object holds the g value of the DSA signature
          key in binary format using a character field.  The DSA
          signature values are defined in [1]."
     ::= { atmseconfilBaseGroup 3 }

AuditsigKeyx        OBJECT-TYPE
     SYNTAX         Octet String (size 20)
     MAX-ACCESS          not-accessible
     STATUS         current
     DESCRIPTION
```

```
            "This object holds the x value of the DSA signature
            key in binary format using a character field.  The DSA
            signature values are defined in [1]."
        ::= { atmseconfilBaseGroup 4 }
```

```
AuditsigKeyk        OBJECT-TYPE
     SYNTAX         Octet String (size 128)
     MAX-ACCESS          not-accessible
     STATUS         current
     DESCRIPTION
          "This object holds the k value of the DSA signature
          key in binary format using a character field.  The DSA
          signature values are defined in [1].  If this object is
          smaller in size than 128 octets, it shall be right
          aligned and padded with 0's."
     ::= { atmseconfilBaseGroup 5 }

--Security Filter Table

secfiltertable      OBJECT-TYPE
     SYNTAX         AtmconfilrulesEntry
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "The ATM Security Connection Filtering Rules table.
          This table specifies events that can be permitted or
          denied."
     INDEX               seconfilindex
     REFERENCE
     ::= { atmseconfilMIBObjects 2 }

AtmconfilrulesEntry
     SEQUENCE  {
               seconfilindex
               incomport
               outgoport
               incomaddrtable
               outgoaddrtable
               aaltype
               atmdtladdr
               trafficcontract
               qosmaxpcr
               connectiontype
               blli
               encapsulation
               vpvcmerge
               starthour
               startmin
               endhour
               endmin
               permitdeny
               }
```

```
seconfilindex       OBJECT-TYPE
     SYNTAX         Integer32
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "A value assigned to a entry in the ATM Security
          Connection Filtering Rules table that identifies it in
          the MIB."
     REFERENCE
     ::= { seconfilindex 1 }


incomport  OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the incoming ATM
          port in the row.  Ports may be wildcarded to specify
          ranges of ports"
     ::= { seconfilindex 2 }


outgoport  OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-create
     STATUS  current
          DESCRIPTION
          "The value of this object specifies the outgoing ATM
          port in the row.  Ports may be wildcarded to specify
          ranges of ports"
     ::= { seconfilindex 3 }

incomaddrtable  OBJECT-TYPE
     SYNTAX         incomaddrentry
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "The ATM incoming address filter table.  Rows can be
          added to this table to specify incoming address that
          are permitted or that are denied.  Addresses can be
          wildcarded to permit or deny ranges of addresses."

     INDEX               incomaddrindex
     REFERENCE
     ::= { seconfilindex 4 }
```

```
incomaddrentry ::=
     SEQUENCE  {
               incomaddrindex      the index for the incoming
                                   address table
               incomatmalias       the alias for the ATM
                                   address,
               incomatmaddress     the ATM address,
               incomaddrmask       the incoming address mask
     }

incomaddrindex  OBJECT-TYPE
     SYNTAX         Integer 32
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "A value assigned to a entry in the ATM filter
          incoming address table identifies it in the MIB."
     REFERENCE
     ::= { incomaddrindex 1 }

incomatmalias  OBJECT-TYPE

     SYNTAX  Octet String
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the alias for the
          ATM address in the row."
     ::= { incomaddrindex 2 }

incomatmaddr  OBJECT-TYPE

     SYNTAX  AtmAddr
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the ATM address
          in the row."
     ::= { incomaddrindex 3 }
```

```
incomaddrmask OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value for this object defines the incoming
          address mask that is used for this port."
     ::= { incomaddrindex 4 }

outgoaddrtable OBJECT-TYPE
     SYNTAX         outgoaddrentry
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "The ATM outgoing address filter table.  Rows can be
          added to this table to specify outgoing address that
          are permitted or that are denied.  Addresses can be
          wildcarded to permit or deny ranges of addresses."

     INDEX               atmconfiloutgoaddrindex
     REFERENCE
     ::= { seconfilindex 5 }

outgoaddrentry ::=
     SEQUENCE  {
               outgoaddrindex      the index for the outgoing
                                   address table
               outgoatmalias       the alias for the ATM
                                   address,
               outgoatmaddress     the ATM address,
               outgoaddrmask       the outgoing address mask
     }

outgoaddrindex OBJECT-TYPE
     SYNTAX         Integer32
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "A value assigned to a entry in the ATM filter
          outgoing address table identifies it in the MIB. "
     REFERENCE
     ::= { outgoaddrindex 1 }
```

```
outgoatmalias  OBJECT-TYPE

     SYNTAX  Octet String
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the alias for the
          ATM address in the row."
     ::= { outgoaddrindex 2 }

outgoatmaddress  OBJECT-TYPE

     SYNTAX  AtmAddr
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the ATM address
          in the row."
     ::= { outgoaddrindex 3 }

outgoaddrmask OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value for this object defines the outgoing
          address mask that is used for this port."
     ::= { outgoaddrindex 4 }

aaltype  OBJECT-TYPE
     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The AAL Types that are allowed for an incoming
          connection.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0001              AAL 1
          0x0002              AAL 2
          0x0004              AAL 3/4
          0x0008              AAL 5
     "
     ::= { seconfilindex 6 }
```

```
atmdtladdrtable     OBJECT-TYPE
     SYNTAX         dtladdrentry
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "This table holds addresses that are either permitted
          or denied to be within the dtl of a setup message.
          Rows can be added to this table to specify outgoing
          address that are permitted or that are denied.  By
          setting the row status addresses can be permitted or
          denied.  Addresses can be wildcarded to permit or
          deny ranges of addresses."
     INDEX               atmdtladdrindex
     REFERENCE
     ::= { seconfilindex 7 }

dtladdrentry  ::=
     SEQUENCE  {
               atmdtlalias         the alias for the ATM
                                   address,
               atmdtladdress       the ATM address,
               atmdtladdrmask      the dtl address mask
     }
atmdtladdrindex     OBJECT-TYPE
     SYNTAX         Integer32
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "A value assigned to a entry in the dtl address table
          identifies it in the MIB."
     REFERENCE
     ::= { atmdtladdrindex 1 }

atmdtlalias  OBJECT-TYPE

     SYNTAX  Octet String
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the alias for the
          ATM address in the row."
     ::= { atmdtladdrindex 2 }
```

```
atmdtlmaddress  OBJECT-TYPE

     SYNTAX  AtmAddr
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the ATM address
          in the row."
     ::= { atmdtladdrindex 3 }

atmdtladdrmask

     SYNTAX  AtmAddr
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          " The value for this object defines the dtl address
          mask that is used for this port."
     ::= { atmdtladdrindex 4 }

trafficcontract  OBJECT-TYPE
     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The type of traffic contract that the connection may
          use.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0001              CBR
          0x0002              VBR
          0x0004              UBR
          0x0008              ABR
          0x0010              undefined
     "
     ::= { seconfilindex 8 }

qosmaxpcr  OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object defines the maximum peak
          cell rate allowed for an incoming connection."
     ::= { seconfilindex 9 }
```

```
connectiontype  OBJECT-TYPE
     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object defines the type of
          connection allowed on this port.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0001              SVC
          0x0002              SPVC
          0x0004              SPVP
          0x0008              undefined"
     ::= { seconfilindex 10 }

blli  OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the BLLI
          codepoints that are permitted.
       Bits       BLLI value       Significance
     0x00000001     0000      Reserved
     0x00000002     0001      ATM Forum LAN Emulation Control
                              Direct VCC
     0x00000004     0002      ATM Forum LAN Emulation 802.3
                              Data Direct VCC
     0x00000008     0003      ATM Forum LAN Emulation 802.5
                              Data Direct VCC
     0x00000010     0004      ATM Forum LAN Emulation 802.3
                              Multicast VCC
     0x00000020     0005      ATM Forum LAN Emulation 802.5
                              Multicast VCC
     0x00000040     0006      Circuit Emulation Service (CES)
                              DS1/E1 Basic
     0x00000080     0007      CES E1 with Channel Associated
                              Signalling (CAS)
     0x00000100     0008      CES DS1 Super Frame with CAS
     0x00000200     0009      CES DS1 Extended Super Frame with
                              CAS
     0x00000400     000A      P-NNI Peer Group Leader to Peer
                              Group Leader
     0x00000800     000B      JT2 Nx64 Service with CAS
     0x00001000     000C      VTOA ATM trunking VCC
```

```
     0x00002000     000D      VTOA ATM trunking E1 w/CAS VCC
     0x00004000     000E      VTOA ATM trunking DS1 SF w/CAS
                              VCC
     0X00008000     000F      VTOA ATM trunking DS1 ESF w/CAS
                              VCC
     0X00010000     0010      VTOA ATM trunking CCS (N-ISDN)
                              VCC
     0X00020000     0011      VTOA ATM trunking DS1/E1 DBCES
                              Basic Service
     0X00040000     0012      VTOA ATM trunking E1 DBCES
                              Service w/CAS
     0X00080000     0013      VTOA ATM trunking DS1 SF DBCES
                              Service w/CAS
     0X00100000     0014      VTOA ATM trunking DS1 ESF DBCES
                              Service w/CAS
     0x00200000     0015      PNNI with security, preplaced
                              keys
     0x00400000     0016      PNNI with security, SME
     0x00800000     0017      PNNI with security, IKE"
     ::= { seconfilindex 11 }

encapsulation  OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object specifies the type of
          encapsulation allowed for the connection.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0001              LLC encapsulation
          0x0002              LLC for routed protocols
          0x0004              LLC for bridged protocols
          0x0008              MPOA Tagged encapsulation
          0x000A              null encapsulation"
     ::= { seconfilindex 12 }
```

```
vpvcmerge  OBJECT-TYPE

     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object specifies if VC merge or VP
          merge is allowed for the connection.  If the bit is
          set for the feature it is permitted.  If the bit is
          not set for the feature it is not permitted.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0001              VC merge
          0x0002              VP merge"
     ::= { seconfilindex 13 }

starthour  OBJECT-TYPE

     SYNTAX  Integer (0..23)
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object defines the hour portion of
          the start time that connections are allowed on this
          port."
     ::= { seconfilindex 14 }

startmin  OBJECT-TYPE

     SYNTAX  Integer (0..59)
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object defines the minutes portion
          of the start time that connections are allowed on
          this port."
     ::= { seconfilindex 15 }
```

```
endhour  OBJECT-TYPE

     SYNTAX  Integer (0..23)
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object defines the hour portion of
          the end time that connections are not allowed on this
          port."
     ::= { seconfilindex 16 }

endmin  OBJECT-TYPE

     SYNTAX  Integer (0..59)
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object defines the minute portion
          of the end time that connections are not allowed on
          this port."
     ::= { seconfilindex 17 }

permitdeny  OBJECT-TYPE

     SYNTAX  TruthValue
     ACCESS  read-create
     STATUS  current
     DESCRIPTION
          "The value of this object specifies if the filter
          rules specified in the row are permitted or denied.
          TRUE specifies that the rules in the row are
          permitted, FALSE specifies that the rules in the row
          are denied."
     ::= { seconfilindex 18 }
```

```
-- Audit Table

audittable          OBJECT-TYPE
     SYNTAX         AuditEntry
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "The audit table allows call level audit, routing
          audit and SSIE audit to be enabled on a per port
          basis."
     INDEX               auditentryindex
     REFERENCE
     ::= { atmseconfilMIBObjects 3 }

AuditEntry
     SEQUENCE  {
               Callaudit
               Routingaudit
               SSIEaudit

auditentryndex      OBJECT-TYPE
     SYNTAX         Integer32
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
          "A value in the Audit Entry table that identifies it
          in the MIB.  Any incoming connection on a port will
          have the parameters defined in this table audited."
     REFERENCE
     ::= { auditentryindex 1 }

Callaudit  OBJECT-TYPE
     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object determines which Call
          oriented events will be audited.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0000              none
          0x0001              Audit ALERTING message
          0x0002              Audit CALL PROCEEDING message
          0x0004              Audit CONNECT message
          0x0008              Audit CONNECT ACKNOWLEDGE message
          0x0010              Audit RELEASE message
```

```
          0x0020              Audit RELEASE COMPLETE message
          0x0040              Audit SETUP message
          0x0080              Audit STATUS message
          0x0100              Audit STATUS INQUIRY message
          0x0200              Audit RESTART message
          0x0400              Audit NOTIFY message
          0x0800              Count Dropped Cells
          0x1000              Count Tagged Cells
          0x2000              Count Total Passed Cells
          0x4000              Count Security OAM Cells
          0x8000              Vendor-Specific"
     ::= { auditentryindex 2 }

Routingaudit  OBJECT-TYPE
     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object determines which PNNI
          packet types will be audited.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0000              none
          0x0001              Audit Hello
          0x0002              Audit PTSP
          0x0004              Audit PTSE Acknowledgement
          0x0008              Audit Database Summary
          0x0010              Audit PTSE Request"
     ::= { auditentryindex 3 }

SSIEaudit  OBJECT-TYPE
     SYNTAX  Integer32
     ACCESS  read-write
     STATUS  current
     DESCRIPTION
          "The value of this object determines if SSIE will be
          audited.
          The bits in this integer have the following meaning:

          Bit                 Significance
          0x0000              none
          0x0001              Audit SSIE
     "
     ::= { auditentryindex 4 }
```

4.0 Audit File

The audit file is constructed of several sections each formatted in Type Length Value format. Each section is turn comprised of individual information elements formatted in the Type Length Value format. Each information element states whether inclusion of that element is mandatory or optional.

The first section is switch specific and provides a mechanism to identify the switch that originated this audit file and the beginning and ending times for the audit log. One instance of the switch specific section is generated by a single switch every X minutes.

Each audited call has its own call statistics section. This section contains the parameters specific to each call including cell counts, addressing, path through the network, etc.

A third section allows for the auditing of PNNI routing information.

A fourth section allows for the auditing of signaling messages. This section allows entire signaling messages to be written to the audit log.

The last section contains information about the call's security associations. An ATM node audits a SME if only (1) the SME is done in signaling, or (2) the SME is done in-band and the ATM node is a participant in the exchange. An ATM node is not capable of auditing an in-band SME in which it is not a participant.

4.1 Audit Log Header

Each audit record shall be written to the audit log with the following header:

Audit Record Identifier 0x5555BBBB	Type	Record Length	Signature ID	Time Stamp	Signature	Value

4.1.1 Audit Record Type

The Audit Record Type has length of 4 octets and is coded as follows:

Type	Audit Record
0x00000001	Switch Specific Section
0x00000002	Signaling Message
0x00000003	PNNI Message
0x00000004	In-band SSIE
0x00000005	Call Statistics

4.1.2 Audit Record Length

The audit record length indicates the number of octets of the audit record and has length 4 octets. The audit record length includes the length of the audit record value, the audit record signature ID, the audit record time stamp, and the audit record signature. It excludes the audit record identifier type and length.

4.1.3 Signature ID

The audit record signature ID shall have length of 4 octets. The first octet shall be coded as follows:

Bits 87654321	Definition
00000001	DSA/SHA -1
11110000 – 11111111	User defined

The remaining 3 octets contain the length of the signature.

The algorithm details for Digital Signature Algorithm (DSA) can be found in [1], the details for the Secure Hash Algorithm – 1 (SHA-1) can be found in [2].

Note: using a Message Authentication Code (MAC) instead of a digital signature may provide a more efficient solution with adequate message integrity albeit with a loss of non-repudiation. Other schemes that use batch signatures may retain the non-repudiation property at some loss of granularity.

4.1.4 Audit Record Time Stamp

The audit record time stamp has length 8 octets and contains the local switch time that the audit record was written to the audit log. The time stamp has the following structure: 342 bits of Unix time followed by 32 bits coded as 0 or microseconds.

4.1.5 Audit Record Signature

The audit record signature shall have a length as specified in the last 3 octets of the Signature ID field. The signature type is specified in the first octet of the signature ID field. The signature is computed across the audit record type, audit record length, signature ID, audit record time stamp, and audit record value (including the audit record padding, if present). When the signature type is DSA, the signature is composed of 2 values, r and s. Both r and s are 160 bits in length, with the value of r residing in the high order 160 bits and the value of s residing in the low order 160 bits. The DSA signature shall be computed in accordance with [1].

4.1.6 Audit Record Value

The audit record value shall be variable in length and shall contain the audit record. The audit record value shall be padded out to a multiple of 32 bits with "0"s.

4.2 Switch-Specific Section

The following fields are mandatory and recorded in the audit log. The switch specific section shall be written to the audit log every X minutes, where X is an implementation specific value.

Bits 8	7	6	5	4	3	2	1	octet
Audit Record Header								
Address Identification			0	0	0	0	0	1
Start Time								2
Start Time (cont)								3
Start Time (cont)								4
Start Time (cont)								5
End Time								6
End Time (cont)								7
End Time (cont)								8
End Time (cont)								9
Switch ID								10
Switch ID (cont)								11
Switch ID (cont)								12
Switch ID (cont)								42
IP Address								43
IP Address (cont)								44
IP Address (cont)								45
IP Address (cont)								46
IP Address (cont)								47
IP Address (cont)								48
IP Address (cont)								49
IP Address (cont)								50
ATM Address								51
ATM Address (cont)								52
ATM Address (cont)								53
ATM Address (cont)								54
ATM Address (cont)								71
p								72 – 200
q								201 – 221
g								222 – 350
k								351 - 469
Vendor-Specific Information OUI								470
Vendor-Specific Information OUI								471
Vendor-Specific Information OUI								472
Vendor-Specific Information								473 etc.

4.2.1 Addressing Identification

The Addressing Identification field shall have length of 3 bits and coded as follows:

Bits 876	Definition
001	E.164
010	ATM End System Address

4.2.2 Start Time

The mandatory 4 octet start time field indicates the beginning of the period that is audited. Each time auditing is enabled the start time is set. This value is the binary encoding of the number of seconds since 00:00:00 GMT on January 1, 1970 (same as UNIX time).

4.2.3 End Time

The mandatory 4 octet end time field indicates the end of the period that is audited. Each time auditing is disabled, the end hour is set. This value is the binary encoding of the number of seconds since 00:00:00 GMT on January 1, 1970 (same as UNIX time).

4.2.4 Switch Identifier

This mandatory 32 octet field contains an identifier for the switch.

4.2.5 IP Address

This mandatory 8 octet field contains the IP address for the switch. If this field contains an IPv4 address, the address will be right justified with the extra 4 octets coded as “0”s.

4.2.6 ATM Address

This mandatory 20 octet field contains the ATM address of the switch. The ATM address type is specified by the Addressing Identification field.

4.2.7 Vendor-Specific Field

This optional field contains vendor-specific information. The first 3 octets of this field contain the Organizationally Unique Identifier (OUI).

4.3 Call Section

When call auditing is enabled, the node will audit any signaling messages that it receives that are specified as being audited in the MIB object Callaudit . When an auditable message is received the entire signaling message, including the protocol discriminator, call reference number, any IEs contained within the message, and the DTL, are written to the audit log.

4.3.1 Vendor-Specific Field

This optional field contains vendor-specific information. The first 3 bits of the vendor-specific field are the OUI.

4.4 Call Statistics

The following table contains statistics related to each call. The signature on this field is not be computed each time that one of the values changes. The signature is computed every X seconds, where X is an implementation specific value based on security policy. Before a new signature is computed all parameters in the call statistics section are updated.

Bits 8	7	6	5	4	3	2	1	octet
Audit Record Type Indicator								
Incoming Port								1
Incoming VPI								2
Incoming VPI (cont)				0	0	0	0	3
Incoming VCI								4
Incoming VCI (cont)								5
Outgoing Port								6
Outgoing VPI								7
Outgoing VPI (cont)				Call Type				8
Outgoing VCI								9
Outgoing VCI (cont)								10
Dropped Cell Count								11
Count Dropped Cells (cont)								12
Count Dropped Cells (cont)								13
Count Dropped Cells (cont)								14
Count Dropped Cells (cont)								15
Count Tagged Cells								16
Count Tagged Cells (cont)								17
Count Tagged Cells (cont)								18
Count Tagged Cells (cont)								19
Count Total Passed Cells								20
Count Total Passed Cells (cont)								21
Count Total Passed Cells (cont)								22
Count Total Passed Cells (cont)								23
Count Security OAM Cells								24
Count Security OAM Cells (cont)								25
Count Security OAM Cells (cont)								26
Count Security OAM Cells (cont)								27
Count OAM Cells (cont)								28
Count OAM Cells (cont)								29
Count OAM Cells (cont)								30
Count OAM Cells (cont)								31
Connection Start Time								32
Connection Start Time (cont)								33
Connection Start Time (cont)								34
Connection Start Time (cont)								35
Connection End Time								36
Connection End Time (cont)								37
Connection End Time (cont)								38
Connection End Time (cont)								39
0	0	0	0	0	0	0	Call End	40
Count Security Setup Failures								41
Count Security Setup Failures (cont)								42

Bits								
8	7	6	5	4	3	2	1	octet
Count Security Setup Failures (cont)								43
Count Security Setup Failures (cont)								44

4.4.1 Incoming Port

When call auditing is enabled, the ATM node records the incoming port in the 2 octet Incoming Port field in the audit file for each connection. The capability to audit the incoming port is mandatory.

4.4.2 Incoming VPI

When call auditing is enabled, the ATM node records the incoming Virtual Path Indicator (VPI) in the 12 bit Incoming VPI field in the audit file for each connection. The capability to audit the incoming VPI is mandatory.

4.4.3 Incoming VCI

When call auditing is enabled, the ATM node records the incoming Virtual Channel Indicator (VCI) in the 2 octet Incoming VCI field in the audit file for each connection. The capability to audit the incoming VCI is mandatory.

4.4.4 Outgoing Port

When call auditing is enabled, the ATM node records the outgoing port in the 2 octet Outgoing Port field in the audit file for each connection. The capability to audit the outgoing port is mandatory.

4.4.5 Outgoing VPI

When call auditing is enabled, the ATM node records the outgoing VPI in the 12 bit Outgoing VPI field in the audit file for each connection. The capability to audit the outgoing VPI is mandatory.

4.4.6 Outgoing VCI

When call auditing is enabled, the ATM node records the outgoing VCI in the 2 octet Outgoing VCI field in the audit file for each connection. The capability to audit the outgoing VCI is mandatory.

4.4.7 Call Type

When call auditing is enabled, the ATM node records the call type for each connection. The capability to audit the call type is mandatory.

Bits 4321	Definition
0001	SVC
0010	PVC
0011	PVP
0100	SPVC

4.4.8 Count Dropped Cells

When call auditing is enabled and Count Dropped Cells is enabled, the ATM node counts the number of dropped cells for each port/VPI/VCI and writes the count to the 4 octet Dropped Cell Count field in the audit file. The cell count is for the lifetime of the connection and the counter wraps. The capability to audit the dropped cell count is optional.

4.4.9 Count Tagged Cells

When call auditing is enabled and Count Tagged Cells is enabled, the ATM node counts the number of dropped cells for each port/VPI/VCI and write the count to the 4 octet Tagged Cell Count field in the audit file. The cell count is for the lifetime of the connection and the counter wraps. The capability to audit the tagged cell count is optional.

4.4.10 Count Total Passed Cells

When call auditing is enabled and Count Total Passed Cells is enabled, the ATM node counts the number of passed cells for each port/VPI/VCI and writes the count to the 4 octet Total Passed Cell Count in the audit log. The cell count is for the lifetime of the connection and the counter wraps. The capability to audit the total passed cell count is optional.

4.4.11 Count Security OAM Cells

When call auditing is enabled and Count Security Operations, Administration & Maintenance (OAM) Cells is enabled, the ATM node counts the number of Security OAM cells for each port/VPI/VCI and write the count to 4 octet Security OAM Cell Count in the the audit log. The cell count is for the lifetime of the connection and the counter wraps. The capability to audit the security OAM cell count is optional.

4.4.12 Count OAM Cells

When call auditing is enabled and Count OAM Cells is enabled, the ATM node counts the number of OAM cells for each port/VPI/VCI and write the count to the 4 octet OAM Cell Count field in the audit log. The cell count is for the lifetime of the connection and the counter wraps. The capability to audit the total OAM cell count is optional.

4.4.13 Connection Start Time

When call auditing is enabled, the ATM node records the time the connection was established and write it to the 4 octet Connection Start Time Field in the audit log. This value is the binary encoding of the number of seconds since 00:00:00 GMT on January 1, 1970 (same as UNIX time). The capability to audit the start hour of the connection is mandatory.

4.4.14 Call End

When call auditing in enabled, the ATM node writes a "1" to this 1 bit field if the call has not terminated during the audit period specified in the switch specific section. If the call does continue longer than the auditing period, the end hour and end minute fields contain the end hour and end minute for the auditing period. If the call has ended before the audit period has ended, this field contains a "0". Note, it is possible for a connection to continue longer than the auditing period. Support for the call end bit is mandatory.

4.4.15 Connection End Time

When call auditing is enabled, the ATM node records the time that the connection was terminated and writes it to the 4 octet Connection End Time Field in the audit log. This value is the binary encoding of the number of seconds since 00:00:00 GMT on January 1, 1970 (same as UNIX time). The capability to audit the end hour of the connection is mandatory.

4.5 Routing Section

When Routing audit is enabled, the ATM node writes PNNI packets specified in the MIB object Routingaudit to the audit log.

4.6 Security Section

When Security Message Exchange auditing is enabled the Security Services Information Element (SSIE) for each flow of the security message exchange, as defined in [3] Section 5.1.3, is written to the audit log.

4.6.1 Security Failures

When Security Message Exchange auditing is enabled, and the initiator or responder detects an error, the fault message that is sent to the peer is written to the audit log. The fault message will be as defined in [3] Section 5.1.5.3.3.5, with cause codes defined in [3] Section 5.1.5.3.6.

4.7 Vendor-Specific Field

This optional field contains vendor-specific information. The first 3 octets of this field are the OUI.