The ATM Forum
Technical Committee

Network Management M4 Security Requirements and Logical MIB

AF-NM-0103.000

January, 1999

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

Table of Contents

1. Introduction

This document specifies the security services for the management plane in form of functional requirements and the management of the security services for the user, control and management planes.

The management of the security services is described in form of requirements and of a protocol independent MIB.

This protocol independent MIB shall be used for the exchange of information across ATM Management Interfaces. It forms a basis from which protocol specific models for ATM can be derived.

This document shall serve as a framework as it focuses at basic capabilities thus leaving sufficient freedom for implementers to use the most current security technology in order to meet the needs of a rapidly changing network.

1.1 Scope

Management of Security includes

- ATM System Security Management
  System security management includes the definition, maintenance, enforcement and monitoring of a security policy, especially the event handling of local and remote security alarms, the security audit management of local and remote events, the interaction of security services and mechanisms and the interaction with other management areas.
- ATM Security Service Management
  This includes the enforcement of a defined security policy, the selection of used mechanisms and algorithms for each service, the assignment of selection rules for mechanisms and algorithms, the management of negotiation between communication partners and the activation / deactivation of mechanisms and algorithms by management operations.
- Security of ATM Network Management
  This includes the protection of the communication for management purposes.

ATM system security management is based on existing network management features (like event forwarding discriminator, log discriminator etc.) and the defined ATM security service management features. ATM system security management is therefore outside the scope of this paper.

The scope of this paper covers topics 2 and 3. It focuses at defining ATM security service management including a protocol independent MIB for the management of the Phase 1 ATM security services. This protocol independent MIB is defined in terms of *managed entities*. Managed entities are abstract protocol independent representations of resources and services in an ATM NE.

2. Security Services

2.1 Security Services for the User and Control Plane

The security services for the User and Control Plane which form the bases for the management capabilities described in this specification are described in [1].

2.2 Security Services for the Management Plane

The following section is based on [5].

2.2.1 Introduction

Generic Security Objectives

Security objectives are derived from the network operators', service providers' or network users' needs, from business relations, legal and regulatory constraints, contractual constraints, etc.

Four basic security objectives have been identified:

- Confidentiality (Confidentiality of stored and transferred information),
- Data Integrity (Protection of stored and transferred information),
- Accountability (Any entity should be responsible for any actions initiated) and
- Availability (All legitimate entities should experience correct access to network management).

The following list of security goals of a network operator can be understood as a combination of these basic security objectives:

- Only legitimate actors shall be able to access network management. Legitimate actors shall be able to access only management functions they are authorized to access
- All actors shall be held accountable for their own but only their own management actions
- It shall be possible to retrieve security related information
- If security violations are detected, this shall be handled in a controlled way, thus minimizing the damaged caused
- After a security breach is detected, it shall be possible to restore normal security levels

In the following text only the basic security objectives are referred to.

Threats and Risks

A threat is a potential violation of security aiming at the objective listed above. Only intentional threats are considered. They can be classified as follows:

- Masquerade ("spoofing"): the pretense by an entity to be a different entity.
- Eavesdropping ("disclosure of information"): a breach of confidentiality by monitoring communication.

- Unauthorized access: an entity attempts to access data in violation to the security policy in force.
- Loss or corruption of information: the integrity of data transferred is compromised by unauthorized deletion, insertion, modification, reordering, replay or delay.
- Repudiation: an entity involved in a communication exchange subsequently denies the fact
- Forgery ("fraud"): an entity fabricates information and claims that such information was received from another entity or sent to another entity.
- Denial of Service: an entity fails to fulfill its function or prevents other entities from performing their functions, e.g. by flooding them with faked error messages.

The following table addresses the general threats to the security objectives stated above. The concrete assessment of these threats, i.e. their probability and their potential impact is outside the scope of this specification, as it depends on the concrete management solution and is therefore highly network operator specific.

Table 2-2-1: Mapping of Threats and Objectives

Threat	**Confidentiality**	**Data Integrity**	**Accountability**	**Availability**
Masquerade	x	x	x	x
Eavesdropping	x	-	-	-
Unauthorized access	x	x	x	x
Loss or corruption of information (transferred)	-	x	x	-
Repudiation	-	-	x	-
Forgery	x	-	x	-
Denial of service	-	-	-	x

2.2.2 Functional Security Requirements

To deal with these threats a set of principal functional requirements can be identified. The requirements stated in this specification are not prioritized. Priorities are derived from the individual assessments of the security threats as stated above and depend on the respective management solution.

As a rule of thumb it can be stated that open management environments require the application of more stringent security mechanisms. In closed environments a sufficient level of security may be achieved mainly by organizational means.

The following table gives an overview of the principal requirements:

Table 2-2-2: Mapping of Functional Requirements and Threats

Functional requirement: Unauthorized access	**Masquerade**		**Eavesdropping**	
Verification of identities	x	-	x	
Controlled Access and Authorization	-	-		x
Protection of confidentiality	-	x		x
Protection of data integrity	-	-	-	
Strong accountability	-	-	-	
Activity logging	x	-	x	
Alarm reporting	x	-	x	
Audit	x	-	x	

Functional requirement: of	**Loss or corruption of information**	**Repudiation**	**Forgery**	**Denial Service**
Verification of identities	-	-	-	-
Controlled Access and Authorization	-	-	-	x
Protection of confidentiality	-	-	-	-
Protection of data integrity	x	-	-	-
Strong accountability	-	x	x	-
Activity logging	-	x	x	x
Alarm reporting	x	-	-	x
Audit	-	x	x	x

Verification of Identities

(CR) SM-1 The M4 interface shall support capabilities to establish and verify the claimed identity of the initiator of a management operation (human user of management application).

This is the most fundamental security requirement for network management. The main purpose is to support other security services and to provide accountability for actions taken.

No distinction shall be made between human users and management applications.

Whether simple or stronger (e.g. symmetric) methods for authentication are applied is up to the network operator depending on his needs.

Authentication will be performed when an association is established across the M4 interface. This shall ensure that e.g. an NE can trust the identity of a management system requesting a change of configuration information and vice versa a management system can trust the identity of an NE sending a critical alarm.

Note: Asymmetric algorithms which provide stronger security are particularly useful for external access to network management functionality, i.e. across the M3 or M5 interface. They will usually not be required across the M4 interface.

Controlled Access and Authorization

(CR) SM-2 The M4 interface shall support capabilities to ensure that users are prevented from gaining access to information or resources that they are not authorized to access.

The proposed level of granularity is access control on individual instances of managed entities.

Protection of Confidentiality

(CR) SM-3 The M4 interface shall support the capability to keep stored and communicated data confidential.

Confidentiality is needed for two purposes:
1. to protect network user related information like billing data and statistics
2. as a service used by other security services, e.g. in order to handle cryptographic keys

Protection of Data Integrity

(CR) SM-4 The M4 interface shall support granting the integrity of stored and communicated data.

Protection of data integrity is needed for two purposes:

1. to protect network user related information like billing data and statistics
2. as a service used by other security services

If this requirement is not met, it is possible to corrupt network management by damaging or altering management data or management operations. There would be a major risk that this would effect the availability of the telecommunication network.
For example, an attacker may choose to fake a critical alarm from a major network element in order to draw attention away from the actual security breach.

Strong Accountability

(CR) SM-5 The M4 interface shall support the capability that an entity can not deny the responsibility for any of its performed actions as well as their effects.
Any individual management user must be hold fully responsible for any of his/her actions.

Activity Logging

(CR) SM-6 The M4 interface shall support the capability to retrieve information about management activities stored in the Network Elements with the possibility of tracing this information to individuals or entities.

For the purpose of many management functions it is necessary to be able to log information about events that have occurred or operations that have been performed or attempted. When such information is retrieved from a log the network manager must be able to determine whether any records have been lost or whether the characteristics of the records stored in the log have been modified at any time.

Alarm Reporting

(CR) SM-7 The M4 interface shall support the capability to generate alarm notifications about certain adjustable and selective security related events.

The security alarm notification provides information regarding operational condition and quality of service, pertaining to security.

Audit

(CR) SM-8 The M4 interface shall support the capability to analyze and exploit logged data on security relevant events in order to check them on violations of system and network security.

An audit is to be seen as an independent review and examination of system information and activities in order to test for adequacy of system controls, to ensure compliance with the established security policy and operational procedures, to detect breaches in security and to recommend changes in control, policy and procedures.

Security Recovery

(CR) SM-9 The M4 interface shall support recovery from attempted breaches on security. Whenever a attempt to breach security occurs it shall be possible to handle this attempt in a controlled manner, meaning that the attempt shall not result in a severe degradation of network management availability.

Functional Classes for Security

(O) SM-10 The M4 interface should support functional classes for security. Functional classes can be applied for security assurance for the M4 interface. A minimum set of three function classes should be supported corresponding to three basic security levels:

1. minimal functional class
2. basic functional class
3. advanced functional class

The use of functional classes does not presuppose the use of standardized security mechanisms, any mechanisms that are suited to achieve the required level of security can be applied.

3. Requirements

3.1 ATM Security Service Management

ATM security service management includes the management of

- the (peer) entity authentication service in the user plane,
- the data confidentiality service in the user plane,
- the data origin authentication and integrity service in the user and control plane,
- the access control service in the user plane.

(Peer) entity authentication for the user plane is a first step in establishing secure communications between nodes. (Peer) entity authentication is performed once for a connection, at the beginning of
that connection. The decision to authenticate or not is determined on a VC- by VC basis (cf. [1], chapter 3.1).

Management of (peer) entity authentication may concern each node of an ATM network.

Note: Decision criteria for applying (peer) entity authentication (examples include: requested QoS level, general security policy) are not covered in [1] and therefore outside the scope of this document.

Data confidentiality for the user plane is defined between two nodes (ATM endpoints or intermediate nodes) (cf. [1], chapter 3.2).

Management of data confidentiality may concern each node of an ATM network.

Note: Currently, decision criteria for applying data confidentiality (examples include: requested QoS level, general security policy) are not covered in [1] and therefore outside the scope of this document.

Data origin authentication and integrity for the user plane is defined between two user-endpoints. It is available only to virtual channels, not for virtual paths. The decision to provide integrity or not is determined on a VCC- by VCC basis (cf. [1], chapter 3.3).

Management of data origin authentication and integrity for the user plane only concerns ATM user endpoints.

Note: Decision criteria for applying data origin authentication and integrity (examples include: requested QoS level, general security policy) are not covered in [1] and therefore outside the scope of this document.

Access control for the user plane is performed during connection establishment based on information contained in the security message exchange and network configuration parameters. Access control is provided for all connections on a per-VC basis. (cf. [1], chapter 3.4).

Management of access control may therefore concern each node (ATM endpoints or intermediate nodes).

Note: Management for access control is for further study.

Data origin authentication and integrity for the control plane is defined between two nodes (ATM endpoints or intermediate nodes) (cf. [1], chapter 4.1).

Management of integrity for the control plane may concern each node.

> Note: Decision criteria for applying data origin authentication and integrity are not covered in [1] and therefore outside the scope of this document.

Within the context of this document, requirements are presented as either *conditional requirements*, denoted by (CR) or *objectives*, denoted by (O). Conditional requirements refer to functions that are necessary for operational compatibility of an optional feature (i.e. if the respective security service is supported, than the CR is a requirement). Objectives are considered features that are viewed to be desirable but not essential for managing ATM security services.

3.1.1 Configuration Management for the ATM Security Services

3.1.1.1 States

The basic operations for configuration management are defined as: create, modify, read, activate, deactivate and delete.

Configuration management makes use of the (generic) operational and administrative states as defined in [4]. Following [4], "operability" and "administration" factors affect the management state of a managed object with regard to its corresponding resources' availability.

Operability is defined as: whether or not the resource (e.g. security service, security mechanism or security algorithm) is physically installed and working. The operability of a resource is described by the **operational state** attribute, which has two possible values: disabled and enabled. The enable event consists of action being taken to render the resource partially or fully operable. The enable event can occur only if the managed object's operational state is disabled. The enable event causes a transition to the enabled operational state. The disable event consists of some occurrence that renders the resource totally inoperable. The disable event causes a transition to the disabled operational state:

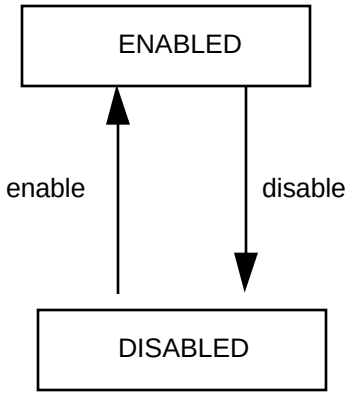

Figure 3-1: Operational state diagram

Administration is defined as: permission to use or prohibition against using the resource, imposed through the management services. The administration of managed objects operates

independently of the operability and is described by the **administrative state** attribute, which has two possible values: lock and unlock .

The unlock event consists of an operation being performed to unlock the managed object's corresponding resource. It can occur only if the managed object's administrative state is locked. It causes a transition to the unlocked administrative state.

The lock event consists of an operation being performed to lock the managed object's corresponding resource. It can occur only if the managed object's administrative state is locked. It causes a transition to the unlocked administrative state:

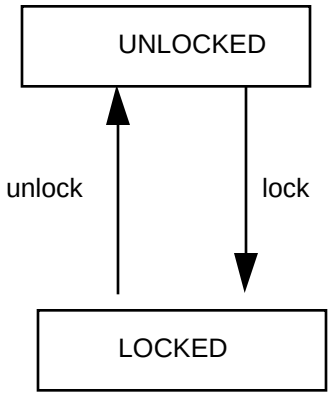

Figure 3-2: Administrative state diagram

In order to express all aspects of a security service, mechanism or algorithm life cycle, the operational state may be optionally refined by introducing a special **life cycle** state.

This state is described by an attribute with possible attribute values: not supported (the resource is not implemented), pending active (the resource is implemented, but not yet in use), active (the resource is in use) and post active (the resource is deactivated and cannot be used).

The generation event consists of an operation being performed to support the managed object's corresponding resource. It can occur only if the state is "not supported". It causes a transition to the pending active state.

The activation event consists of an operation being performed to activate the managed object's corresponding resource. It can occur only if the state is "pending active". It causes a transition to the active state.

The deactivation event consists of an operation being performed to limit the use of the corresponding resource. It can occur only if the state is "active". It causes a transition to the post active state.

The reactivation event consists of an operation being performed to activate the managed object's corresponding resource. It can occur only if the state is "post active". It causes a transition to the active state.

The destruction event consists of an operation being performed to cover logical and possibly physical destruction of the managed object's corresponding resource. It can occur only if the state is "pending active" or "post active". It causes a transition to the state "not supported".

The life cycle state corresponds to the operational state in the following way: the life cycle state values "not supported" , "pending active" and "post active" correspond to the operational state

value "disabled". The life cycle state value "active" corresponds to the operational state value "enabled".

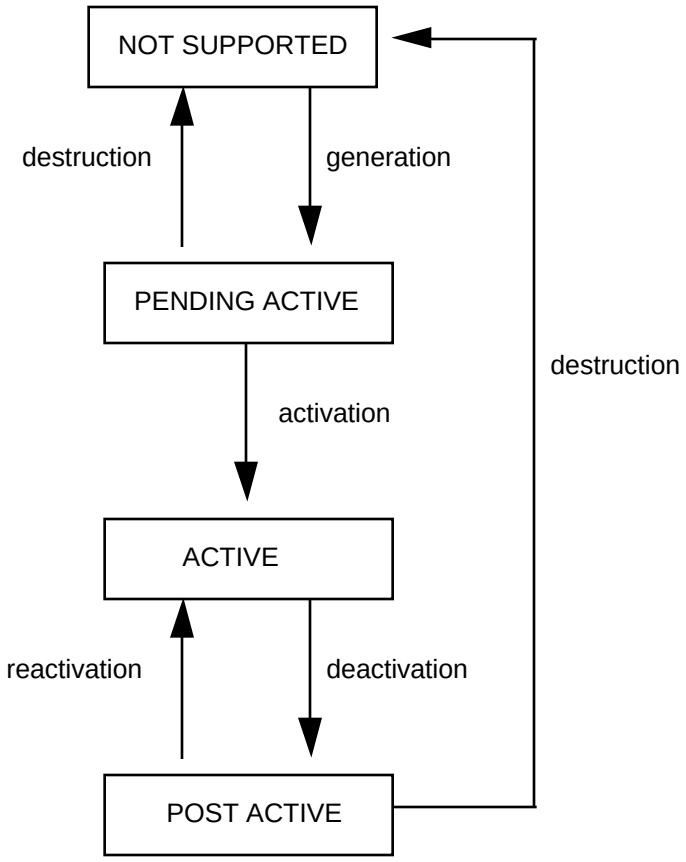

Figure 3-3: Life cycle state diagram

The state transitions correspond to the basic management and to internal operations. A state transition shall be reported .

3.1.1.2 Requirements for Configuration Management

Security is an optional feature in an ATM network. Therefore each security service will either be supported or not supported. This will be expressed by the operational state:

(CR) CM-1 The management of a security service may concern all nodes or just the user endpoints. (this depends on the security service, see above.) The management of a security service shall support the operational state as described in section 3.1.1.1. The attribute value "disabled" means "the respective security service is not supported" and the attribute value "enabled" means "the respective security service is supported".

In order to express all aspects of a security service life cycle, the operational state may be optionally refined by introducing a service life cycle state:

(O) CM-2 The management of a security service should cover all aspects of a life cycle by supporting a service life cycle state as described in section 3.1.1.1.

> Note: The guarantee of the integrity of states in an ATM network is a general network management requirement. It is not specific to security management and therefore outside the scope of this paper.

(CR) CM-3 For each security service the administrative state shall be supported as described in section 3.1.1.1. The possible attribute values are "locked" and "unlocked".

> Note: A “locked” service will only not be supported temporarily, whereas a “deactivated” service is expected not to be supported in the future. However, also a “deactivated” service may be “reactivated”. But this usually requires more (management) actions (e.g. reloading of software), than to “unlock” a locked service.

(Peer) entity authentication, possibly data origin authentication and data integrity and possibly data confidentiality require that the communication partners are named:

(CR) CM-4 It shall be possible to support the management of security entity identifiers. Each security entity identifier uniquely identifies a node (or possibly a user endpoint) as required by the message exchange protocol.

For (peer) entity authentication, various mechanisms are allowed:

(CR) CM-5 The (peer) entity authentication mechanisms differ in regard of the following parameters:

- unilateral or (optionally) mutual,
- number of message exchanges
- usage of optional fields.

It must be possible, to manage each of these different mechanisms individually per node.

(CR) CM-6 The management of a security mechanism shall support the operational state as described in section 3.1.1.1.

In order to express all aspects of a security mechanism life cycle, the operational state may be optionally refined by introducing a mechanism life cycle state.

(O) CM-7 The management of a security mechanism should cover all aspects of a life cycle by supporting a mechanism life cycle state as described in section 3.1.1.1 .

(CR) CM-8 For each security mechanism the administrative state shall be supported as described in section 3.1.1.1. The possible attribute values are "locked" and "unlocked".

For each security service, various algorithms are allowed:

(CR) CM-9 For (peer) entity authentication, mechanisms can be based on symmetric or asymmetric digital signatures (i.e. on algorithms using secret keys or private/public keys) or on hash functions .

The allowed symmetric digital signature algorithms (i.e. algorithms based on secret keys) are: DES (DES40), TripleDES and FEAL (in Cipher Block Chaining (CBC) mode).

The allowed asymmetric digital signature algorithms (i.e. algorithms based on private/public keys) are: RSA, DSA, Elliptic Curve / DSA-Like and ESIGN.

The allowed hash functions are: SHA-1, RIPEMD-160 and MD5. To guarantee (peer) entity authentication, an asymmetric digital signature (i.e. an algorithm based on private/public keys) will be applied to the result of the hash function.

It shall be possible to add further algorithms (e.g. user defined algorithms) to the allowed algorithms. Each of these different algorithms can individually be managed per mechanism.

(CR) CM-10 For confidentiality, mechanisms using enciphering algorithms based on secret keys can be used. The allowed algorithms are: DES (DES40), TripleDES and FEAL. The allowed modes are Cipher Block Chaining (CBC), Counter Mode and Electronic Codebook (ECB).

It shall be possible to add further algorithms (e.g. user defined algorithms) to the allowed algorithms. Each of these different algorithms can individually be managed per mechanism..

(CR) CM-11 For data origin authentication and integrity, mechanisms can be based on signature algorithms or message authentication codes (MAC) (i.e. cryptographic check functions). The allowed signature algorithms respectively. MAC (for the user and for the control plane) are: HMAC-MD5, HMAC-SHA-1, HMAC-RIPEMD-160, DES (DES40) MAC, Triple DES MAC and FEAL MAC (DES, TripleDES and FEAL in CBC mode).

It shall be possible to add further algorithms (e.g. user defined algorithms) to the allowed algorithms. Each of these different algorithms can individually be managed per mechanism.

(CR) CM-12 The management of a security algorithm shall support the operational state as described in section 3.1.1.1.

In order to express all aspects of a security algorithm life cycle, the operational state may be optionally refined by introducing a mechanism life cycle state:

(O) CM-13 The management of a security algorithm should cover all aspects of a life cycle by supporting a algorithm life cycle state as described in section 3.1.1.1.

(CR) CM-14 For each security algorithm, the administrative state shall be supported as described in section 3.1.1.1. The possible attribute values are "locked" and "unlocked".

(O) CM-15 For each security service the selection of a default mechanism (if applicable) and a default algorithm should be supported. The default mechanism and algorithm should be adjustable.

> Note: This kind of default mechanism serves for supporting network providers in defining a security policy in an easy way and does not imply that a specific algorithm must be supported to be in line with the standard. If such a default mechanism does not exist, it would be necessary (depending on the granularity of the security policy) to define explicitly, which mechanisms shall be used for a specific VC. Otherwise, the following strategy is possible: “If a specific mechanism is not defined e.g. for a pair of ATM endpoints , use the default mechanism”.

For the authentication, confidentiality and integrity service, it shall be optionally possible to relate (a subset of) mechanisms (if applicable) and algorithms to the parties, which are involved in communication. This can simplify the negotiation of mechanisms and algorithms between two parties. The calling party only selects for negotiation such mechanisms and algorithms, which are related to the called party. Thus, negotiation is extremely unlikely to fail.

(O) CM-16 For each communication entity pair (i.e. the two end user points or two nodes which are involved in communication) and for the (peer) entity authentication service, the confidentiality service and the data origin and integrity service it should be possible to select mechanisms and algorithms, which should be used (mechanism - algorithm selection). This selection may include the ranking of the used mechanisms and algorithms.

(O) CM-17 For each of these mechanisms - algorithm selections, it should be possible to manage the relationship to the used keys (secret key, public or private key, dependent on the algorithm).

3.1.2 Fault Management for the ATM Security Services

For each ATM security service, there is a need to generate a security alarm, if the service detects, that an entity tries to breach the security (e.g. authentication fails n times in a x minutes period).

(CR) FM-1 If an ATM security service detects a security violation, it shall be possible to forward a security alarm.

(CR) FM-2 All security management actions shall be logged.

3.1.3 Security Management for the ATM Security Services

The management of the security services shall be supported by the security services used for the M4 interface. For details see [2].

3.2 Management for the ATM Security Support Services

Beside the security services also the support services must be managed. These support services are (cf. [1] : chapter 5):

- security message exchange protocols and basic negotiation
- security messaging in the control plane
- security messaging in the user plane
- key exchange
- session key update
- certificate infrastructure.

3.2.1 General Requirements

The following requirements are valid for all support services.

> Note: The management requirements for support services are similar to those for the security services.

3.2.1.1 Configuration Management for the Support Services

(CR) SCM-1 The management of a security support service may concern all nodes or all user endpoints. The management of a security support service shall support the operational state as described in section 3.1.1.1.

In order to express all aspects of a security support service life cycle, the operational state may be optionally refined by introducing a support service state:

(O) SCM-2 The management of a security support service should cover all aspects of a life cycle by supporting a support service state. The support service state is described by a support service state attribute with the possible values "not supported", "pending active", "active" and "post active" and by the state transitions "generation", "activation", "deactivation", "reactivation" and "destruction". The state transitions correspond to the basic management operations and to internal operations. A state transition should be notified.

(CR) SCM-3 For each support service, the administrative state shall be supported as described in section 3.1.1.1. The possible attribute values are "locked" and "unlocked".

3.2.1.2 Fault Management for the Support Services

3.2.2 Key Exchange

The following additional requirements to configuration management exist.

Within the message exchange protocol different options are allowed:

It shall be possible to add further algorithms (e.g. user defined algorithms) to the allowed algorithms. Each of these different algorithms can individually be managed per node.
For each security message exchange protocol, various algorithms are allowed:

(CR) SCM- 5 For key exchange, mechanisms using symmetric or asymmetric algorithms (i.e. algorithms based on secret keys or on private / public keys) can be used.

The allowed symmetric algorithms (i.e. algorithms based on secret keys) are: DES (DES40), TripleDES and FEAL (in CBC mode). The allowed asymmetric algorithms (i.e. algorithms based on private/public keys) are: RSA, Diffie-Hellman and Elliptic Curve / Diffie-Hellman.

It shall be possible to add further algorithms (e.g. user defined algorithms) to the allowed algorithms. Each of these different algorithms can individually be managed per mechanism.

3.2.3 Management for Key Update

The following additional requirements to configuration management exist.

(CR) SCM-6 It shall be possible to enforce key update.

(CR) SCM-6 It shall be possible to manage parameters necessary for key update (e.g. update interval, L_{SKE})

3.2.4 Management for Certificate Infrastructure / Keys

Key management includes the management of local (long term) keys (in the nodes and user endpoints). Key management shall cover all aspects of a life cycle as described in ISO 11770-1 [3]. Configuration management and fault management for keys are already covered by

SCM-1 and SCM-2.

4. Protocol Independent MIB

4.1 Overview

The managed entities representing the ATM security services and resources are:

- ATM Security Endpoint
- Authentication Rule
- Authentication Service (Entity Authentication Service)
- Confidentiality Mechanism
- Confidentiality Rule
- Confidentiality Service
- DES Algorithm
- Diffie-Hellman Algorithm
- DSA Algorithm
- Elliptic Curve Algorithm
- Elliptic Curve / Diffie-Hellman Algorithm
- ESIGN Algorithm
- FEAL Algorithm
- Hash Based Authentication Mechanism
- HMAC-MD5 Algorithm
- HMAC-SHA-1 Algorithm
- HMAC-RIPEMD-160 Algorithm
- Integrity Mechanism
- Integrity Rule
- Integrity Service (Data Origin Authentication and Integrity Service)
- Key Management Rule
- Private Key
- Public Key
- Public Key Authentication Mechanism
- Public Key Management Mechanism
- RIPEMD-160
- RSA Algorithm
- Secret Key
- Secret Key Authentication Mechanism
- Secret Key Management Mechanism
- TripleDES Algorithm

Managed entities representing Certification Authorities, Trusted Third Parties or Third Parties for key distribution are outside the scope of this specification.

Notes:

1. Implementations may contain additional user defined managed entities (e.g. managed entities for representing user defined algorithms). These user defined managed entities are outside the scope of this paper.

2. A number of Managed Entities in Figures 4-1 to 4-4 does not show any Containment relationship. The reason is that the respective Managed Entity they are contained in (and consequently named after) depends on their field of application.
   As an example: an algorithm will be named by the Managed Entity "ATM Security Endpoint" if it is used in the M4 NE view, it will be named by the Managed Entity "ATM subnetwork" if it is used in the M4 Network View.

The relationships between the managed entities are shown in Figures 4-1, 4-2 ,4-3 and 4-4. All relationships shall be interpreted as being bi-directional, i.e. if Managed Entity A is related to Managed Entity B, then Managed Entity B has a reverse relationship with Managed Entity A.

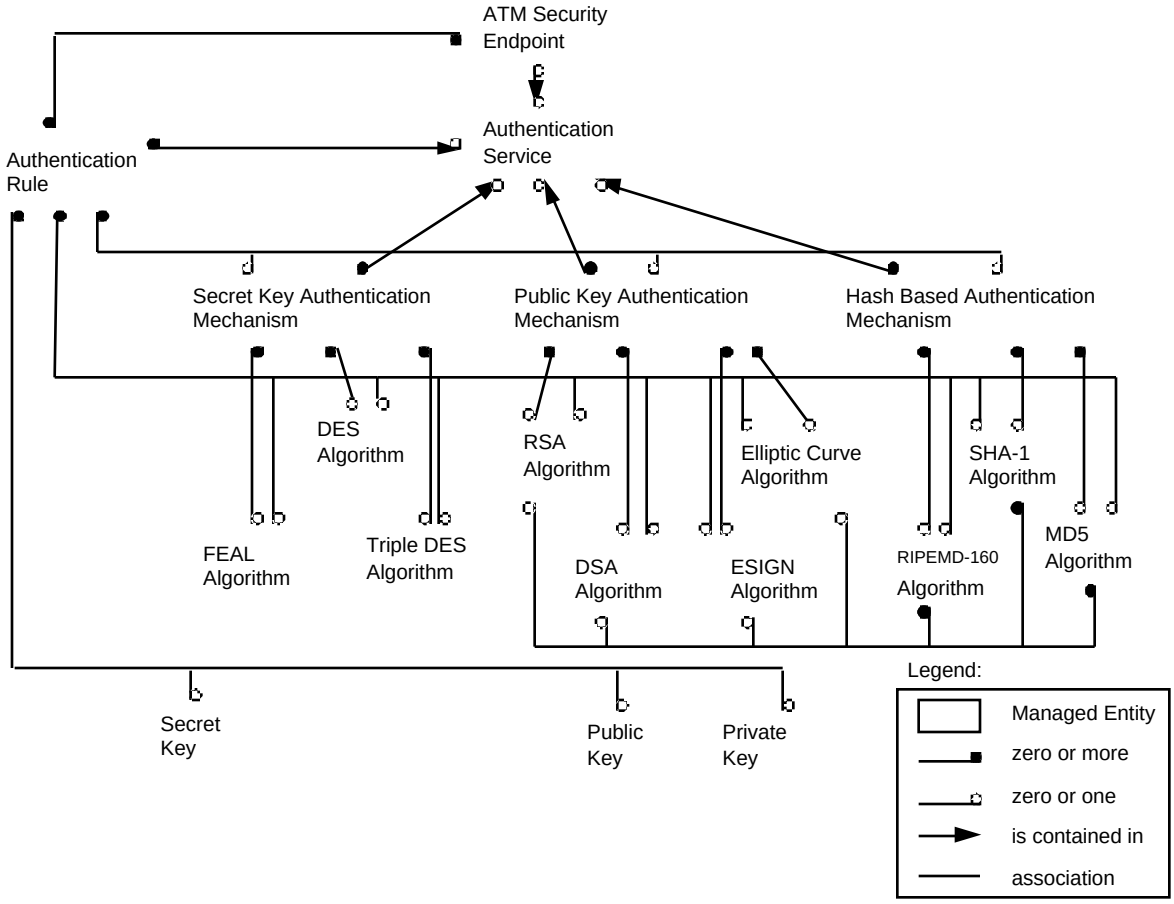

Figure 4-1: Managed Entity Relationship Diagram for the Authentication Service

Note:

The relationships between Hash algorithms und Public Key algorithms will only be used for the digital signature of a hash value (results of a hash algorithm).

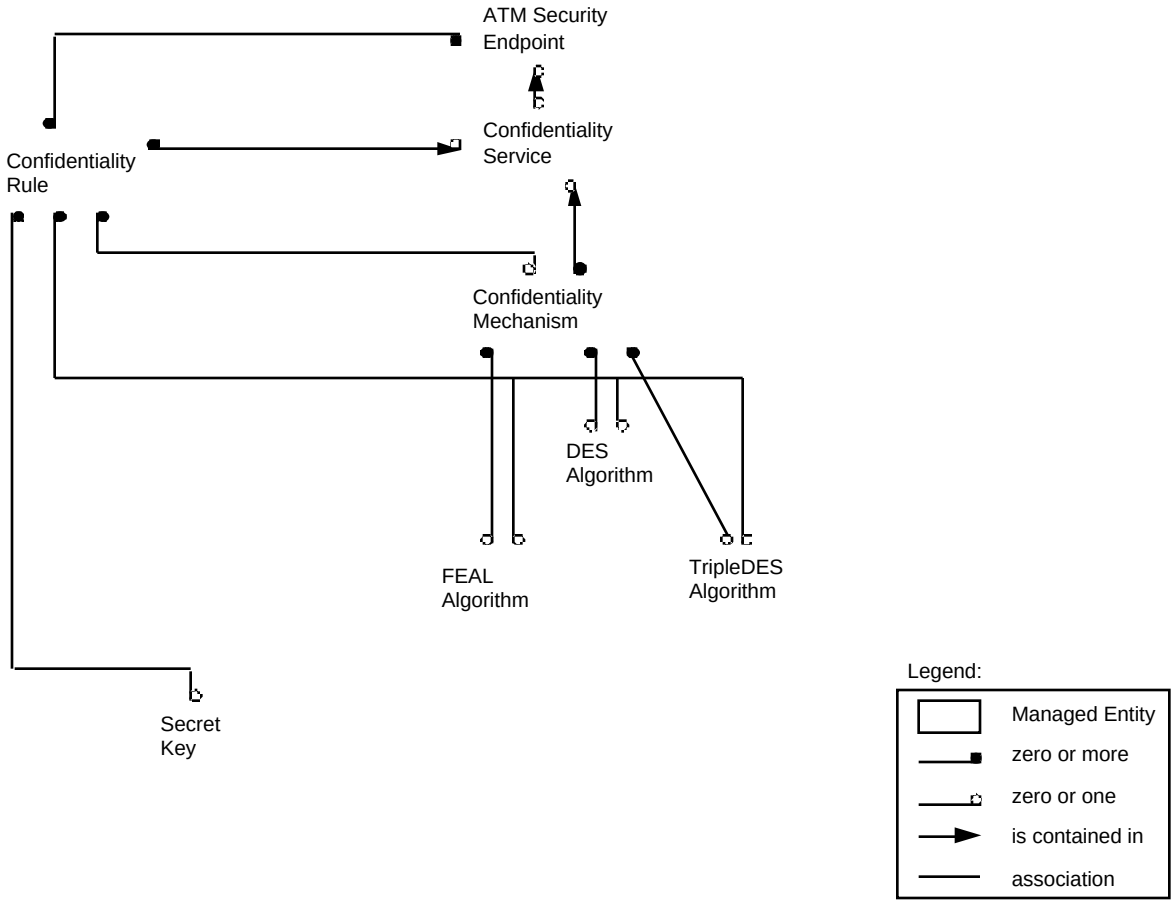

Figure 4-2: Managed Entity Relationship Diagram for the Confidentiality Service

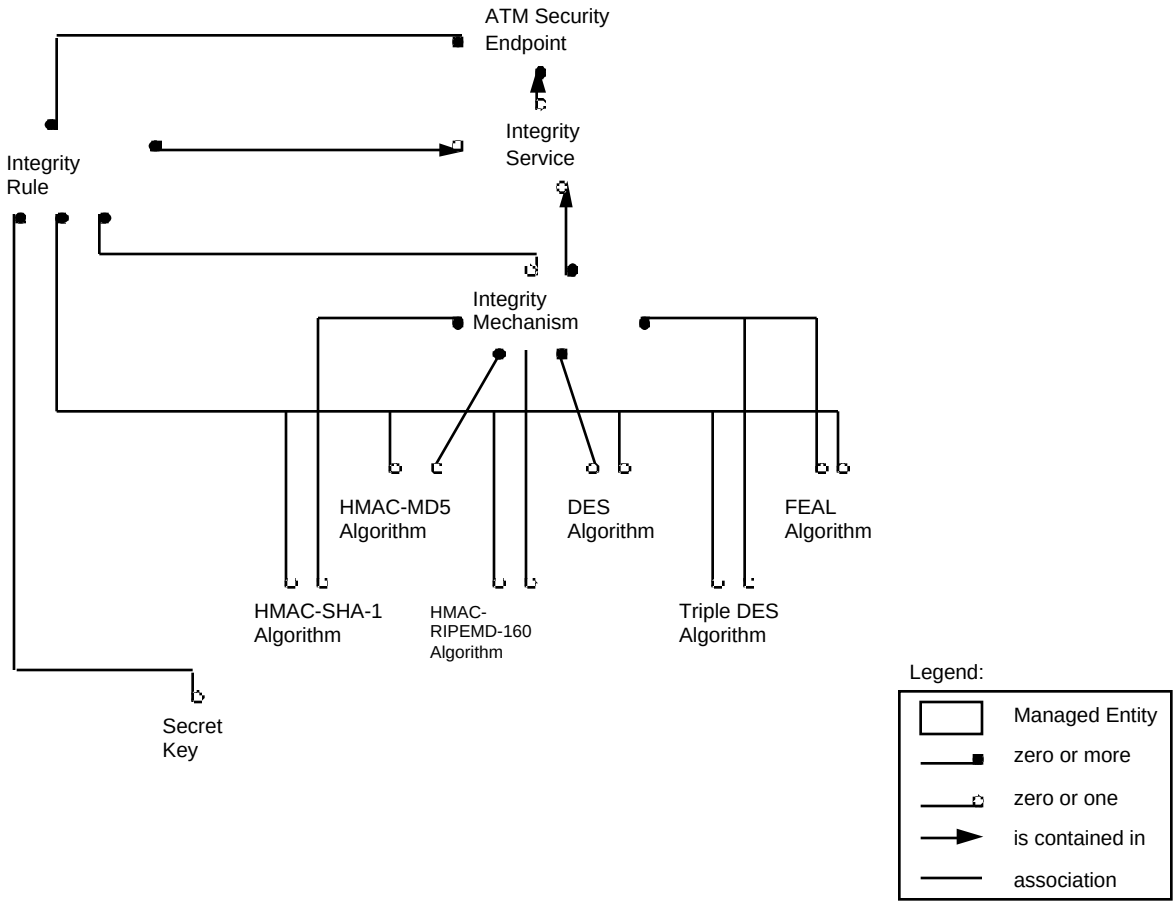

Figure 4-3: Managed Entity Relationship Diagram for the Integrity Service

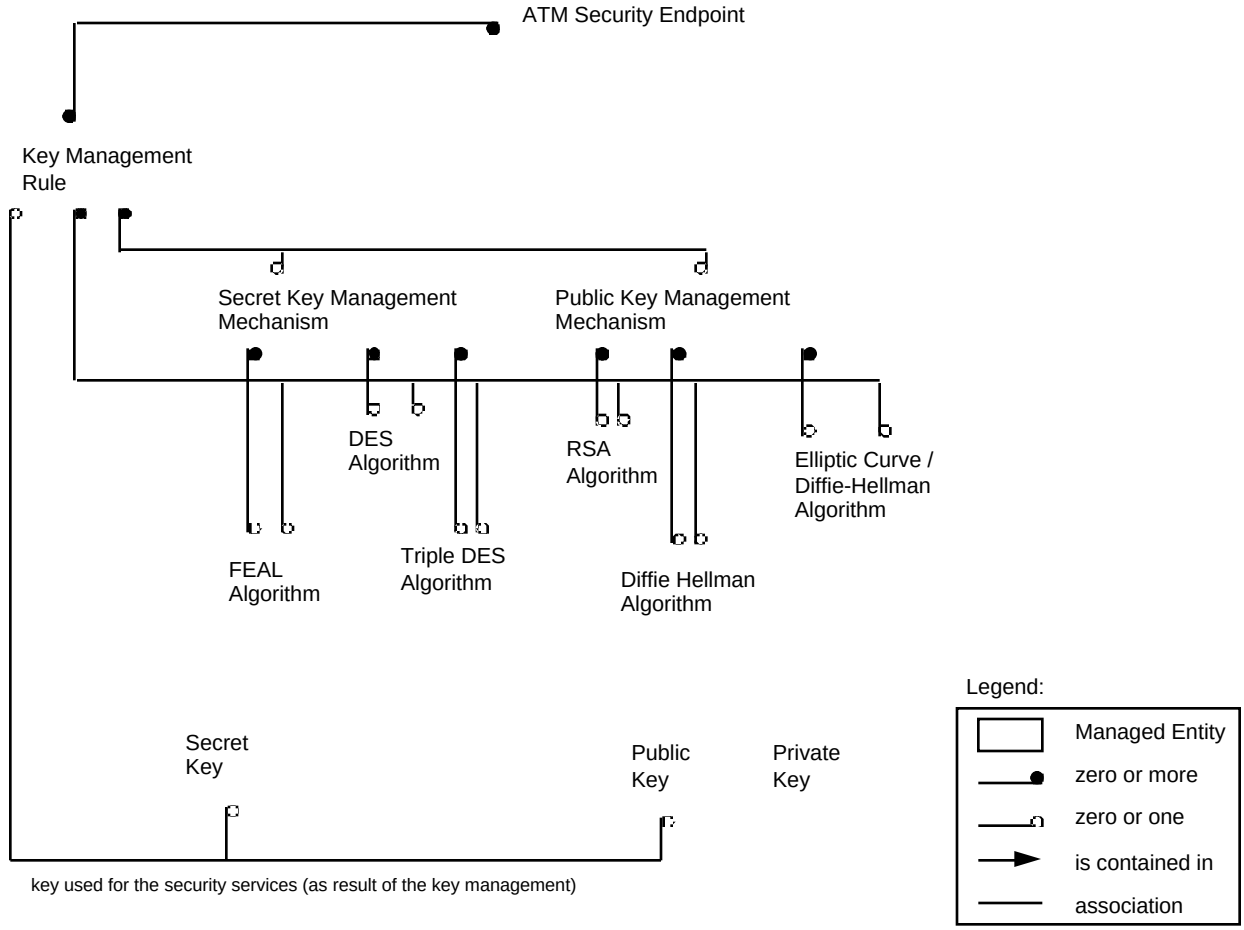

Figure 4-4 : Managed Entity Relationship Diagram for Key Management

Notes:

1. Dependent on the used mechanism, a third party (e.g. key distribution center , key translation center etc.) may be involved in key management. This third party is outside the scope of [2]. The management of this party is therefore outside the scope of this specification.
2. Dependent on the used mechanism, the key management requires a secret key, a public key (of the involved third party or of the ATM Security Endpoint instance) or a private key (of the ATM Security Endpoint instance). Key generation and exchange mechanisms for the purpose of key management are not described in [2]. The management of these mechanisms is therefore not part of this specification.
3. According to [3], key management mechanisms only provide the exchange of either secret keys or public keys. Mechanisms for private key generation and exchange are outside the scope of [3]. Therefore, the management of these mechanisms is outside the scope of this specification.

A detailed description of each managed entity is provided in the subsections that follow. The descriptions include

(1) the purpose of the entity,

(2) the attributes of the entity,

(3) the management operations (actions) that may be performed on the entity

(4) the notifications generated by the managed entity,

(5) the relationship(s) that the entity supports with other entities, and

(6) the behavior of the entity.

4.2 ATM Security Endpoint

(1) Purpose

The ATMSecurity Endpoint entity is used to represent the parties of an ATM Network, which may be involved in security.

(2) Attributes

mandatory:

Security Identity Identifier: as described in the ATM Phase 1 Security Specification [1].

(3) Operations

create, read, delete executed by security management and key management

(4) Notifications

managed entity creation

managed entity deletion

(5) Relationships

Each ATM security endpoint instance may support an authentication service, an access control service, a confidentiality service and / or a data origin authentication and integrity service for the user plane and a data origin authentication and integrity service for the control plane.

Each ATM security endpoint instance may be involved in a secure communication either as a calling party or as a called party (relationships to the authentication rule, confidentiality rule and integrity rule).

A key management rule instance is related to at least two ATM security endpoint instances, if a secret key is managed (all instances, which share the same secret key).

Each ATM security endpoint instance may be related to zero or more key management rules.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and related authentication rule, confidentiality rule, integrity rule, authentication service, confidentiality service and integrity service instances must be deleted.

Before an instance of this entity can be deleted, all of its relationships and the related key management rule instances must be deleted.

4.3 Authentication Rule

(1) Purpose

This managed entity is used to represent

- for an ATM node security endpoint instance: the default authentication mechanism and the default authentication algorithm
- for two ATM node security endpoint instances communicating via a communication link : the possible authentication mechanisms and for each authentication mechanism: the possible authentication algorithms and the keys, which will be used for authentication service

(2) Attributes

mandatory:

Authentication Rule Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

(3) Operations

create, read, delete executed by security management

(4) Notifications

managed entity creation

managed entity deletion

(5) Relationships

Each instance may be related to:

one or two ATM node security endpoint instances; these relationships represent the called party and (if a second relationship is existing) the calling party

an authentication service instance which shall be used by the called party / calling party

an authentication mechanism instance which shall be used by the called party / calling party

an authentication algorithm instance which shall be used by the called party / calling party

either a secret key instance which shall be used for authentication by the called party / calling party

or a private key instance which shall be used for authentication by the called party and a public key which shall be used for authentication by the calling party.

(6) Behavior

If a relationship to a calling party exists: Instances will only be created,

- if the calling party supports authentication service, and
- if the chosen authentication mechanism and the chosen authentication algorithm will be used both by the called and by the calling party and
- if the chosen secret key will be shared between called and calling party respectively if the chosen public key corresponds to the chosen private key.

Before an instance of this entity can be deleted, all of its relationships must be deleted.

4.4 Authentication Service

(1) Purpose

This managed entity is used to represent a (peer to peer) entity authentication service used for user plane authentication of an ATM node security endpoint.

(2) Attributes

mandatory:

Authentication Service Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not an authentication service is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the authentication service. Possible values are: lock and unlock.

Scope: This attribute describes the scope of the authentication service. Possible value: user plane

optional:

Authentication Service State: This attribute describes all aspects of an authentication service life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

authentication service or mechanism or algorithm violation, if authentication fails caused by the usage of a wrong key, by a replay attack (duplicate authentication token) ...

time violation, if authentication fails caused by delay (time out), ...

(5) Relationships

Each authentication service instance can use one or more secret key, public key or hash based authentication mechanism instances.
An authentication service instance may be related to zero or more authentication rule instances.
Each authentication service instance refers to an ATM node security endpoint instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and related authentication rules must be deleted.

4.5 Confidentiality Mechanism

(1) Purpose

This managed entity is used to represent the confidentiality mechanisms which can be used by a confidentiality service.

(2) Attributes

mandatory:

Confidentiality Mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a confidentiality mechanism is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the confidentiality mechanism. Possible values are: lock and unlock.

optional:

Confidentiality Mechanism State: This attribute describes all aspects of a confidentiality mechanism life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each confidentiality mechanism instance can be used by one or more confidentiality services.

A confidentiality mechanism instance may be related to zero or more confidentiality rule instances.

Each confidentiality mechanism instance uses for encryption zero or one of the following algorithm instances: DES(DES40), TripleDES or FEAL.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and all related confidentiality rules must be deleted. Confidentiality Rule

4.6 Confidentiality Rule

(1) Purpose

This managed entity is used to represent

- for an ATM node security endpoint instance: the default confidentiality mechanism and the default confidentiality algorithm
- for two ATM node security endpoint instances communicating via a communication link: the possible confidentiality mechanisms and for each confidentiality mechanism: the possible confidentiality algorithms and the keys, which will be used for confidentiality service.

(2) Attributes

mandatory:

Confidentiality Rule Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

(3) Operations

create, read, delete executed by security management

(4) Notifications

managed entity creation

managed entity deletion

(5) Relationships

Each instance may be related to:

one or two ATM node security endpoint instances; these relationships represent the called party and (if a second relationship is existing) the calling party

a confidentiality service instance which shall be used by the called party / calling party

a confidentiality mechanism instance which shall be used by the called party / calling party

a confidentiality algorithm instance which shall be used by the called party / calling party

a secret key instance which shall be used by the called party / calling party.

(6) Behavior

If a relationship to a calling party exists: Instances will only be created, if

- if the calling party supports confidentiality service, and
- if the chosen confidentiality mechanism and the chosen confidentiality algorithm will be used both by the called and by the calling party and
- if the chosen secret key will be shared between called and calling party.

Before an instance of this entity can be deleted, all of its relationships must be deleted.

4.7 Confidentiality Service

(1) Purpose

This managed entity is used to represent a confidentiality service used for user plane confidentiality of an ATM node security endpoint.

(2) Attributes

mandatory:

Confidentiality Service Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a confidentiality service is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the confidentiality service. Possible values are: lock and unlock.

Scope: This attribute describes the scope of the confidentiality service. Possible value: user plane

optional:

Confidentiality Service State: This attribute describes all aspects of an authentication service life cycle. Possible values are: "not supported“, "pending active", "active, "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

confidentiality service or mechanism or algorithm violation, if confidentiality fails

(5) Relationships

Each confidentiality service instance can use one or more confidentiality mechanism instances.

A confidentiality service instance may be related to zero or more confidentiality rule instances.

Each confidentiality service instance refers to an ATM node security endpoint instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related confidentiality rules must be deleted.

4.8 DES Algorithm

(1) Purpose

This managed entity is used to represent the DES algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

Key length: This attribute describes the used key length. Possible values are: 40 and 56.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management and key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more secret key authentication, confidentiality or integrity mechanism instances.

An algorithm may be related to zero or more authentication rule, confidentiality rule or integrity rule instances.

Each algorithm instance can be used by zero or more key management rule instances.
A key management rule instance may be related to zero or one algorithm instance.
Each algorithm instance may be used in zero or more secret key management mechanism instances.
Each secret key management mechanism instance may be related to zero or one algorithm instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rules, confidentiality rules and integrity rules must be deleted.

Before an instance of this entity can be deleted, all of its related key management rule instances must be deleted.

4.9 Diffie-Hellman Algorithm

(1) Purpose

This managed entity is used to represent the Diffie-Hellman algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by zero or more key management rule instances.
A key management rule instance may be related to zero or one algorithm instance.
Each algorithm instance may be used in zero or more public key management mechanism instances.
Each public key management mechanism instance may be related to zero or one algorithm instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its related key management rule instances must be deleted.

4.10 DSA Algorithm

(1) Purpose

This managed entity is used to represent the DSA algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more public key authentication mechanism instances.

An algorithm instance may be related to zero or more SHA-1 algorithm or MD5 algorithm or RIPEMD-160 algorithm instances.

An algorithm may be related to zero or more authentication rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rules must be deleted.

4.11 Elliptic Curve Algorithm

(1) Purpose

This managed entity is used to represent the Elliptic Curve algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more public key authentication mechanism instances.

An algorithm instance may be related to zero or more SHA-1 algorithm or MD5 algorithm or RIPEMD-160 algorithm instances. An algorithm may be related to zero or more authentication rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rules must be deleted.

4.12 Elliptic Curve / Diffie-Hellman Algorithm

(1) Purpose

This managed entity is used to represent the Elliptic Curve / Diffie-Hellman algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by zero or more key management rule instances.
A key management rule instance may be related to zero or one algorithm instance.
Each algorithm instance may be used in zero or more public key management mechanism instances.
Each public key management mechanism instance may be related to zero or one algorithm instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its related key management rule instances must be deleted.

4.13 ESIGN Algorithm

(1) Purpose

This managed entity is used to represent the ESIGN algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more public key authentication mechanism instances.

An algorithm instance may be related to zero or more SHA-1 algorithm or MD5 algorithm or RIPEMD-160 algorithm instances.

An algorithm may be related to zero or more authentication rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rules must be deleted.

4.14 FEAL Algorithm

(1) Purpose

This managed entity is used to represent the FEAL algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management and key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more secret key authentication, confidentiality or integrity mechanism instances.

An algorithm may be related to zero or more authentication rule, confidentiality rule or integrity rule instances.

Each algorithm instance can be used by zero or more key management rule instances.
A key management rule instance may be related to zero or one algorithm instance.
Each algorithm instance may be used in zero or more secret key management mechanism instances.
Each secret key management mechanism instance may be related to zero or one algorithm instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rules, confidentiality rules and integrity rules must be deleted.

Before an instance of this entity can be deleted, all of its related key management rule instances must be deleted.

4.15 Hash Based Authentication Mechanism

(1) Purpose

This managed entity is used to represent the hash based authentication mechanisms which can be used by an authentication service.

(2) Attributes

mandatory:

Authentication Mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a hash based authentication mechanism is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the hash based authentication mechanism. Possible values are: lock and unlock.

Authentication Token Construction: This attribute describes the rule how to construct an authentication token.

Authentication Mode: Possible values: unilateral (default), mutual

Number of Message Exchanges: possible values: 2 (default), 3

optional:

Authentication Mechanism State: This attribute describes all aspects of an authentication mechanism life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each hash based authentication mechanism instance can be used by one or more authentication services.

A hash based authentication mechanism may be related to zero or more authentication rule instances.

Each hash based authentication mechanism instance uses for the construction of the authentication token zero or one of the following algorithms: SHA-1 or MD5 or RIPEMD-160

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and all related authentication rules must be deleted.

4.16 HMAC-MD5 Algorithm

(1) Purpose

This managed entity is used to represent the HMAC-MD5 algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more integrity mechanism instances.

An algorithm may be related to zero or more integrity rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related integrity rules must be deleted.

4.17 HMAC-SHA-1 Algorithm

(1) Purpose

This managed entity is used to represent the HMAC-SHA-1 algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more integrity mechanism instances.

An algorithm may be related to zero or more integrity rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related integrity rules must be deleted.

4.18 HMAC-RIPEMD-160 Algorithm

(1) Purpose

This managed entity is used to represent the HMAC-RIPEMD-160 algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more integrity mechanism instances.

An algorithm may be related to zero or more integrity rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related integrity rules must be deleted.

4.19 Integrity Mechanism

(1) Purpose

This managed entity is used to represent the integrity mechanisms which can be used by an integrity service.

(2) Attributes

mandatory:

Integrity Mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not an integrity mechanism is physically installed and working. Possible values are: disabled enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the integrity mechanism. Possible values are: lock and unlock.

Integrity Check Value Construction: This attribute describes the rule how to construct a data integrity check value.

optional:

Integrity Mechanism State: This attribute describes all aspects of an integrity mechanism life cycle.
Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each integrity mechanism instance can be used by one or more integrity service instances.

An integrity mechanism may be related to zero or more integrity rule instances.

Each integrity mechanism instance uses for the construction of an integrity check value zero or one of the following algorithms:

HMAC-MD5, HMAC-SHA-1, HMAC-RIPEMD-160, DES (DES40), Triple DES or FEAL (DES, Triple DES and FEAL only in the CBC mode).

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and related integrity rules must be deleted.

4.20 Integrity Rule

(1) Purpose

This managed entity is used to represent

- for an ATM node security endpoint instance: the default integrity mechanism and the default integrity algorithm

- for two ATM node security endpoint instances communicating via a communication link : the possible integrity mechanisms and for each integrity mechanism: the possible integrity algorithms and the keys, which will be used for integrity service .

(2) Attributes

mandatory:

Integrity Rule Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

(3) Operations

create, read, delete executed by security management

(4) Notifications

managed entity creation

managed entity deletion

(5) Relationships

Each instance may be related to:

one or two ATM node security endpoint instances; these relationships represent the called party and (if a second relationship is existing) the calling party

an integrity service instance which shall be used by the called party / calling party

an integrity mechanism instance which shall be used by the called party / calling party

an integrity algorithm instance which shall be used by the called party / calling party

a secret key instance which shall be used by the called party / calling party

(6) Behavior

If a relationship to a calling party exists: Instances will only be created,

- if the calling party supports integrity service, and
- if the chosen integrity mechanism and the chosen integrity algorithm will be used both by the called and by the calling party and
- if the chosen secret key will be shared between called and calling party.

Before an instance of this entity can be deleted, all of its relationships must be deleted.

4.21 Integrity Service

(1) Purpose

This managed entity is used to represent an integrity service used for user plane and control plane data authentication and integrity of an ATM node security endpoint.

(2) Attributes

mandatory:

Integrity Service Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not an integrity service is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the integrity service. Possible values are: lock and unlock.

Scope: This attribute describes the scope of the integrity service. Possible values: user plane, control plane

optional:

Integrity Service State: This attribute describes all aspects of an integrity service life cycle.

Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

integrity service or mechanism or algorithm violation, if integrity fails

time violation, if integrity fails caused by delay (time out), ...

(5) Relationships

Each integrity service instance can use one or more integrity mechanism instances.

An integrity service may be related to zero or more integrity rule instances.

Each integrity service instance refers to an ATM node security endpoint instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and related integrity rules must be deleted.

4.22 Key Management Rule

(1) Purpose

This managed entity is used to represent the key management procedure.

(2) Attributes

mandatory:

Key Management Rule Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Key Update Period: This attribute describes the period (e.g. in the form of a time period or an application specific counter), after which a key update will be enforced.

(3) Operations

create, modify, read, delete executed by key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each key management rule instance is related either to a secret key management mechanism instance or to a public key management mechanism instance.

A secret key management mechanism instance may be related to one or more key management rule instances.

A public key management mechanism instance may be related to one or more key management rule instances.

If the key management rule instance is related to a secret key management mechanism, then:

- This key management rule instance is related either to a FEAL algorithm, DES algorithm or TripleDES algorithm instance. A FEAL algorithm, DES algorithm or TripleDES algorithm instance may be related to zero or more key management rule instances.
- This key management rule instance is related to exactly one secret key instance, which will be shared between two ATM Security Endpoint instances (result of the secret key management mechanism).

If the key management rule instance is related to a public key management mechanism, then:

- This key management rule instance is related either to a RSA algorithm, Diffie-Hellman algorithm or Elliptic Curve / Diffie-Hellman algorithm instance. A RSA algorithm, Diffie-Hellman algorithm or Elliptic Curve / Diffie-Hellman algorithm instance may be related to zero or more key management rule instances.
- This key management rule instance is related to exactly one secret key or one public key instance (result of the public key management mechanism). This key instance is only related to one key management rule.

A key management rule instance is related to at least two security endpoint instances. If a secret key is distributed, then the related instances share the secret key. If a public key is distributed, it is sufficient to indicate, which instance is the originator of the public key. Each security endpoint instance may be related to zero or more key management rules.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related secret key or public key instances, which are used for key management, must be deleted.

4.23 MD5 Algorithm

(1) Purpose

This managed entity is used to represent the MD5 algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more hash based authentication mechanism instances.

An algorithm must be related either to a RSA algorithm or to a DSA algorithm or to an ESIGN algorithm or to a Elliptic Curve algorithm instance.

An algorithm may be related to zero or more integrity rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related integrity rules must be deleted.

4.24 Private Key

(1) Purpose

This managed entity is used to represent the private keys which are involved in a secure communication between two communication partners.

(2) Attributes

mandatory:

Key Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Key Value: This attribute contains the key value. The key value must be stored in a confidential way.

optional:

valid from: This attribute fixes the date and time, from which on the key is valid

valid until: This attribute fixes the date and time, when key validation expires

act key update counter: This attribute contains the actual counter value, which will be used for key update enforcement.

comp key update counter: This attribute contains the comparison counter value, which will be used for key update enforcement.

(3) Operations

create, modify, read, delete executed by security management and key management

(4) Notifications

attribute value change (the key value must be stored in a confidential way)

managed entity creation

managed entity deletion

validity period exceeded

(5) Relationships

A private key instance may be involved in zero or more authentication rule instances.

(6) Behavior

A private key can only be used for algorithms related to public key authentication mechanism instances.

Before an instance of this entity can be deleted, all of its relationships must be deleted.

Before an instance of this entity can be deleted, the related key management rule instance must be deleted.

The value of "comp key update counter" shall be the same as that of "key update period" (ME key management rule).

4.25 Public Key

(1) Purpose

This managed entity is used to represent the public keys which will be used for a secure communication between two communication partners.

(2) Attributes

mandatory:

Key Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Key Value: This attribute contains the key value. The key value should be stored in a confidential way.

optional:

party name: This attribute identifies the party, which holds the corresponding private key. This attribute shall only be used for key management.

valid from: This attribute fixes the date and time, from which on the key is valid

valid until: This attribute fixes the date and time, when key validation expires

((note: the operations modify, activate, deactivate acc. BTD shall not be supported))

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management and key management

(4) Notifications

attribute value change (the key value must be stored in a confidential way)

managed entity creation

managed entity deletion

(5) Relationships

A public key instance may be related to zero or more authentication rule instances.

Zero or one public key instance (the result of the key management) may be related to a key management rule instance. This public key instance may be related to authentication rule instances.

A key management rule may be related to zero or one public key instance (the result of the key management).

(6) Behavior

A public key can only be used for algorithms related to public key authentication mechanism instances.

Before an instance of this entity can be deleted, all of its relationships must be deleted.

Before an instance of this entity can be deleted, the related key management rule instance must be deleted.

4.26 Public Key Authentication Mechanism

(1) Purpose

This managed entity is used to represent the public key authentication mechanisms which can be used by an authentication service.

(2) Attributes

mandatory:

Authentication Mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a public key authentication mechanism is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the public key authentication mechanism. Possible values are: lock and unlock.

Authentication Token Construction: This attribute describes the rule how to construct an authentication token.

Authentication Mode: Possible values: unilateral (default), mutual

Number of Message Exchanges: possible values: 2 (default), 3

optional:

Authentication Mechanism State: This attribute describes all aspects of an authentication mechanism life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each public key authentication mechanism instance can be used by one or more authentication service instances.

A public key authentication mechanism instance may be related to zero or more authentication rule instances.

Each public key authentication mechanism instance uses for the construction of an authentication token zero or one instances of the following algorithms: RSA, DSA, Elliptic Curve or ESIGN.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and all related authentication rules must be deleted.

4.27 Public Key Management Mechanism

(1) Purpose

This managed entity is used to represent the public key management mechanisms which can be used for key management.

(2) Attributes

mandatory:

Key management mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a public key management mechanism is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the public key management mechanism. Possible values are: lock and unlock.

optional:

Key Management Mechanism State: This attribute describes all aspects of an key management mechanism life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each public key management mechanism instance can be used by one or more key management rule instances.

A key management rule instance is related to zero or one public key management mechanism instance.

Each public key management mechanism instance uses for a secure key management zero or one instances of the following algorithms: RSA, Diffie-Hellman, Elliptic Curve / Diffie-Hellman; at least one relationship must exist.

An instance of the RSA, Diffie-Hellman or Elliptic Curve / Diffie-Hellman algorithm may be related to zero or more public key management mechanism instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and all related key management rules must be deleted.

4.28 RIPEMD-160 Algorithm

(1) Purpose

This managed entity is used to represent the RIPEMD-160 algorithm.

(2) Attributes

mandatory:

Algorithm Id:	This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.
Operational State:	This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled
Administrative State:	This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State:	This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more hash based authentication mechanism instances.

An algorithm must be related either to a RSA algorithm or to a DSA algorithm or to an ESIGN algorithm or to a Elliptic Curve algorithm instance.

An algorithm may be related to zero or more integrity rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related integrity rules must be deleted.

4.29 RSA Algorithm

(1) Purpose

This managed entity is used to represent the RSA algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management and key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more public key authentication mechanism instances.

An algorithm instance may be related to zero or more SHA-1 algorithm or MD5 algorithm or RIPEMD-160 algorithm instances.

An algorithm may be related to zero or more authentication rule instances.

Each algorithm instance can be used by zero or more key management rule instances.

A key management rule instance may be related to zero or one algorithm instance.

Each algorithm instance may be used in zero or more public key management mechanism instances.

Each public key management mechanism instance may be related to zero or one algorithm instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rules must be deleted.

Before an instance of this entity can be deleted, all of its related key management rule instances must be deleted.

4.30 Secret Key

(1) Purpose

This managed entity is used to represent the secret keys which will be used for a secure communication between two communication partners.

(2) Attributes

mandatory:

Key Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Key Value: This attribute contains the key value. The key value must be stored in a confidential way.

optional:

party name: This attribute identifies the party, which holds the corresponding secret key. This attribute shall only be used for key management.

valid from: This attribute fixes the date and time, from which on the key is valid

valid until: This attribute fixes the date and time, when key validation expires.

act key update counter: This attribute contains the actual counter value, which will be used for key update enforcement.

comp key update counter: This attribute contains the comparison counter value, which will be used for key update enforcement.

(3) Operations

create, modify, read, delete executed by security management and key management

(4) Notifications

attribute value change (key value must be stored in a confidential way)

managed entity creation

managed entity deletion

validity period exceeded

(5) Relationships

A secret key instance may be related to zero or more authentication rule, confidentiality rule or integrity rule instances.

Zero or one secret key instance (the exchanged or agreed key as result of the key management) may be related to a key management rule instance. This secret key instance may be related to authentication rule, integrity rule or confidentiality rule instances.

A key management rule may be related to zero or one secret key instance (the exchanged or agreed key as result of the key management).

(6) Behavior

A secret key can only be used for algorithms related to secret key authentication, confidentiality or integrity mechanisms.

Before an instance of this entity can be deleted, all of its relationships must be deleted.

Before an instance of this entity can be deleted, the related key management rule instance must be deleted.

The value of "comp key update counter" shall be the same as that of "key update period" (ME key management rule).

4.31 Secret Key Authentication Mechanism

(1) Purpose

This managed entity is used to represent the secret key authentication mechanisms which can be used by an authentication service.

(2) Attributes

mandatory:

Authentication Mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a secret key authentication mechanism is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the secret key authentication mechanism. Possible values are: lock and unlock.

Authentication Token Construction: This attribute describes the rule how to construct an authentication token.

Authentication Mode: Possible values: unilateral (default), mutual

Number of Message Exchanges: possible values: 2 (default), 3

optional:

Authentication Mechanism State: This attribute describes all aspects of a public key authentication mechanism life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each secret key authentication mechanism instance can be used by one or more authentication services.

A secret key authentication mechanism may be related to zero or more authentication rule instances.

Each secret key authentication mechanism instance uses for the construction of an authentication token zero or one of the following algorithm instances: DES(DES40), TripleDES or FEAL.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and all related authentication rules must be deleted.

4.32 Secret Key Management Mechanism

(1) Purpose

This managed entity is used to represent the secret key management mechanisms which can be used for key management.

(2) Attributes

mandatory:

Key management mechanism Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not a secret key management mechanism is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the secret key management mechanism. Possible values are: lock and unlock.

optional:

Key Management Mechanism State: This attribute describes all aspects of an key management mechanism life cycle. Possible values are: "not supported“, "pending active", "active", "post active"

(3) Operations

create, modify, read, activate, deactivate, delete executed by key management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each secret key management mechanism instance can be used by one or more key management rule instances..

A key management rule instance is related to zero or one secret key management mechanism instance.

Each public key management mechanism instance uses for a secure key management zero or one instances of the following algorithms: DES, FEAL, TripleDES; at least one relationship must exist..

An instance of the DES, FEAL or TripleDES algorithm may be related to zero or more secret key management mechanism instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and all related key management rules must be deleted.

4.33 SHA-1 Algorithm

(1) Purpose

This managed entity is used to represent the SHA-1 algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more hash based authentication mechanism instances.

An algorithm must be related either to a RSA algorithm or to a DSA algorithm or to an ESIGN algorithm or to a Elliptic Curve algorithm instance.

An algorithm may be related to zero or more integrity rule instances.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related integrity rules must be deleted.

4.34 TripleDES Algorithm

(1) Purpose

This managed entity is used to represent the TripleDES algorithm.

(2) Attributes

mandatory:

Algorithm Id: This is an attribute which distinguished value can be used as an identifier. The attribute values must be unique.

Operational State: This attribute describes whether or not the algorithm is physically installed and working. Possible values are: disabled and enabled

Administrative State: This attribute describes the permission to use or the prohibition against using the algorithm. Possible values are: lock and unlock.

optional:

Algorithm State: This attribute describes all aspects of an algorithm life cycle. Possible values are: "not supported“, "pending active", "active", "post active".

(3) Operations

create, modify, read, activate, deactivate, delete executed by security management

(4) Notifications

attribute value change (including state transition)

managed entity creation

managed entity deletion

(5) Relationships

Each algorithm instance can be used by one or more secret key authentication mechanism, integrity mechanism or confidentiality mechanism instances.

An algorithm may be related to zero or more authentication rule, confidentiality rule or integrity rule instances.

Each algorithm instance can be used by zero or more key management rule instances.

A key management rule instance may be related to zero or one algorithm instance.

Each algorithm instance may be used in zero or more secret key management mechanism instances.

Each secret key management mechanism instance may be related to zero or one algorithm instance.

(6) Behavior

Before an instance of this entity can be deleted, all of its relationships and the related authentication rule, confidentiality rule and integrity rule instances must be deleted.

Before an instance of this entity can be deleted, all of its related key management rule instances must be deleted.

5. References

(1) ATM Forum Technical Committee, Security, ATM Security Specification Version 1.0, STR-SEC-01.04, October 1998

(2) ATM Forum Technical Committee, STR-NM-M4NE-REQ-02.00; "M4 Interface Requirements and Logical MIB: ATM Network Element View", July 1998

(3) ISO/IEC 11770-1: Information technology - Security techniques - Key management, Part 1: Framework, 1996

(4) ITU-T X.731 (ISO/IEC 10164-2): Information technology - Open systems interconnection - systems management: state management function, January 1992

(5) ETSI TR NA-043208: Network Aspects (NA); Telecommunication Network Management (TMN); Introduction to Standardizing Security for TMN, 1996

6. Abbreviations

CBC	Cipher Block Chaining
CBC-MAC	CBC Message Authentication Code
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ESIGN	Efficient digital signature scheme
FEAL	Fast data encipherment algorithm
ITU	International Telecommunication Union
MD5	Message Digest Algorithm No.5
RSA	Rivest, Shamir and Adleman (algorithm)
SHA	Secure Hash Algorithm