AERONAUTICAL TELECOMMUNICATION NETWORK PANEL (ATNP)
WG3 – APPLICATIONS AND UPPER LAYERS
SG3 – UPPER LAYER ARCHITECTURE
Rio, March 16-20, 1998

ATN Upper layers Security

Prepared by G. Mittaux-Biron

Summary

ICAO is in the process of standardising the requirements for ATN Security. This will allow the development of a secured environment of data transfer between airborne and ground ATN systems, taking into account the various local legislation constraints.

This document only deals with the implementation of security mechanisms in the ATN upper layers (Session to Application) further studies will propose solutions for the lower layers of the ATN.

The Application Service Elements (ASE) provide application-users with the communication functions suitable to each ATN application. The architecture proposed in this document limits the impacts on the ATN ASEs of the implementation of security mechanisms. This means that most of the security mechanisms which will take place in the upper layers will be located in the application layer under the ATN ASEs.

Appendix 1 of this document identifies the management functions that the secured ATN upper layers may require as identified in the Security Framework for Open Systems (Security Audit Framework). The analysis of these requirements lead to the identification of the security related resources which have to be visible from the system managers, i.e. to define the minimum set of security related managed objects to be included in the ATN MIB to cover the management of the security

Appendix 2 of this document explains the need for canonical encoding of ASN.1 structure when secured communications take place.

1 SCOPE

This paper specifies how ATN Upper Layers can use standardised mechanisms from GULS (ISO/IEC 11586) to secure elements of the various ATN ASEs. It also describes how the ATN upper layers can use the Directory authentication framework (ISO/IEC 9594-8) to authenticate the peer systems in the ATN environment.
The aim of this document is to provide a first approach of the technical implementation of the security in the ATN: many aspects still need to be refined, ranging from the precise description of the communications which will take place between ATN "operational" entities and administrative entities like the Certifications Authorities, or the entities which aim at providing private information, or, more technically, the refinement of the formal description of the security mechanisms and their inclusion in the already existing ATN Upper Layers. Some correction will have to be added later. Evolutions and refinement should be made during the future activities which will take place in the sub-group.
Chapter 2 provides the references which have been used as a basis for this document.
Chapter 3 and 4 provides the terms definitions and abbreviations which applies all over this document.
Chapter 5 provides an overview of the upper layers architecture needed to implement the ATN security.
Chapter 6 provides a first level of the definition of the Security services, the impact on the Dialogue service, and a description of the exchanges in the ATN Application Entity.
Chapter 7 proposes a first definition of the mechanisms and exchanges of security information involved in the peer entity authentication, together with an overview of the system security functions, the formal definition of the security transformations and the security exchanges which take place during this phase.
Chapter 8 proposes a first definition of the mechanisms and exchanges of security information involved in the data exchanges in order to protect them from forgery or replay, together with an overview of the system security functions, the formal definition of the security transformations and the security exchanges which take place during this phase.
Appendix 1 provides the system requirements for the security in the upper layers of the ATN.
Appendix 2 is an explanation on the need for canonical encoding variant of PER.

2 REFERENCES

Error! Unknown switch argument.. DED1/ATNIP/STA_ATNP/DCO/42, *Overall Security Concept () - Tony Whyman, Ian Valentine, Nick Pope.*

Error! Unknown switch argument.. ISO/IEC 7498-1:1994 *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*

Error! Unknown switch argument.. ISO 7498-2:1989 *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*

Error! Unknown switch argument.. ISO/IEC 8649:1996 *Information technology — Open Systems Interconnection — Service definition for the Association Control Service Element*

Error! Unknown switch argument.. ISO/IEC 8650-1:1996 *Information technology — Open Systems Interconnection — Connection-oriented protocol for the Association Control Service Element: Protocol specification*

Error! Unknown switch argument.. ISO/IEC 8822:1994 *Information technology — Open Systems Interconnection — Presentation service definition*

Error! Unknown switch argument.. ISO/IEC 8824-1:1995 *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

Error! Unknown switch argument.. ISO/IEC 8824-4:1995 *Information technology — Abstract Syntax Notation One (ASN.1): Parameterisation of ASN.1 specifications*

Error! Unknown switch argument.. ISO/IEC 9545:1994 *Information technology — Open Systems Interconnection — Application Layer structure*

Error! Unknown switch argument.. ISO/IEC 9594-8:1995 *Information technology — Open Systems Interconnection — The Directory: Authentication framework*

Error! Unknown switch argument.. ISO/IEC 10745:1995 *Information technology — Open Systems Interconnection — Upper layers security model*

Error! Unknown switch argument.. ISO/IEC 11586-1:1996 *Information technology — Open Systems Interconnection — Generic upper layers security: Overview, models and notation*

Error! Unknown switch argument.. ITU-T Rec. X.831|ISO/IEC 11586-2:1996 *Information technology — Open Systems Interconnection — Generic upper layers security: Security Exchange Service Element (SESE) service definition*

Error! Unknown switch argument.. ISO/IEC 11586-3:1996 *Information technology — Open Systems Interconnection — Generic upper layers security: Security Exchange Service Element (SESE) protocol specification*

Error! Unknown switch argument.. ISO/IEC 11586-4:1996 *Information technology — Open Systems Interconnection — Generic upper layers security: Protecting transfer syntax specification*

Error! Unknown switch argument.. ISO/IEC 10181-7:1996 *Information technology — Open Systems Interconnection — Security Frameworks for Open Systems: Security Audit Framework*

Error! Unknown switch argument.. ITU-T Rec. X.710|ISO/IEC 9595-4:1991 *Information technology — Open Systems Interconnection — Common management information service definition*

Error! Unknown switch argument.. ITU-T Rec. X.740|ISO/IEC 10164-4:1992 *Information technology — Open Systems Interconnection — System management: Alarm reporting function.*

Error! Unknown switch argument.. ITU-T Rec. X.734|ISO/IEC 10164-5:1993 *Information technology — Open Systems Interconnection — System management: Event report management function.*

Error! Unknown switch argument.. ITU-T Rec. X.735|ISO/IEC 10164-6:1993 *Information technology — Open Systems Interconnection — System management: Log control function.*

Error! Unknown switch argument.. ITU-T Rec. X.736|ISO/IEC 10164-7:1992 *Information technology — Open Systems Interconnection — System management: Security alarm reporting function.*

Error! Unknown switch argument.. ITU-T Rec. X.721|ISO/IEC 10165-2:1992 *Information technology — Open Systems Interconnection — Structure of management information: definition of management information.*

3

4 DEFINITIONS

The following term is used as defined in ISO/IEC 7498-1:

protocol data unit.

The following terms are used as described in ITU-T Rec. X800|ISO/IEC 7498-2:

access control;
authentication exchange;
authentication;
confidentiality;
connection integrity with recovery;
connection integrity without recovery;
data integrity;
data origin authentication;
denial of service;
digital signature;
encipherment;
integrity;
key;
masquerade;
non-repudiation with proof of delivery;
non-repudiation with proof of origin;
notarisation;
passive threat;
peer-entity authentication;
replay threat;
replay;
repudiation;
security audit trail;
security mechanism;
security policy;
security service;
selective field connection integrity;
selective field protection;
signature mechanism;
signature;
threat.
trusted third party.

The following terms are used as defined in ISO/IEC 8822

abstract syntax;
presentation context;
presentation data value.

The following terms are used as defined in ISO/IEC 8824-4

parameterised type.

The following terms are used as defined in ISO/IEC 9545:

application context;
application service element;
application-association, association.

The following terms are used as defined in ISO/IEC 9594-8:

certification authority;
certification path;
public key;
strong authentication.

The following terms are used as defined in ISO/IEC 10745:

security association;
security exchange;
system security object
system security functions
security transformation.

The following term is used as described in ISO/IEC 11586-1:

PROTECTED abstract syntax notation.

For the purposes of this document, the following definitions apply:

a) strong authentication procedures: the procedures defined in ISO/IEC 9594-8 clause 10.

5 ABBREVIATIONS

AARE	ACSE Associate Response APDU
AARQ	ACSE Associate Request APDU
ACSE	Association Control Service Element (ISO/IEC 8649, 8650)
ASE	Application Service Element
ASN.1	Abstract Syntax Notation (ISO/IEC 8824)
CA	Certification Authority
GULS	Generic Upper Layer Security (ISO/IEC 11586)
MIB	Management Information Base
OSI	Open Systems Interconnection (ISO/IEC 7498)
PDU	Protocol Data Unit
PDV	Presentation Data Value
RLRE	Release Response ACSE APDU
RLRQ	Release Request ACSE APDU
SESE	Security Exchange Service Element (ISO/IEC 11586-2 and -3)
SSF	System Security Functions
SSO	System Security Object
TTP	Trusted Third Party
U-ASE	User Application Service Element

6

7 MODEL

7.1 OVERVIEW

One main constraint raised in specifying security in the ATN is to avoid any modification in the ATN ASEs (ADS, CPDLC, FIS…). Provision for security should also be able to provide interoperability of secured ATN stacks with unsecured ATN stacks.
The model proposed in this document allows security services for the secured operation of ATN end systems communications to be provided in the Application Layer. It deals with security between two peer systems in a Dialogue association. It assumes that, in an ATN association between two ASEs, the involved ATN ASEs relies on the Dialogue service for all security consideration.
The provision for security for an application using the ATN upper layers, may be provided at two different levels: the application layer for end-to-end security, and the underlying layers for internet related security. This document only deals with the former.
In order to conform to [Ref.1], for the secure communications in the ATN, it is necessary to authenticate the communicating entities and the messages which are exchanged between the entities.

7.2 ARCHITECTURE

The resulting model includes the following characteristics of the Application layer:

a) A set of ASEs which includes the ATN ASEs: ADS, CPDLC, FIS and CM, together with the SESE and the ACSE. It should be noted that the data issued from the ATN ASEs should be encoded using the canonical form of PER.

b) A modified Dialogue control function, taking into account the security exchanges and the inclusion of SESE.

c) Security specific exchanges: the dirAuthenticationTwoWays and the gulsSignedTransformation security exchanges will be supported. Although no need for confidentiality has been expressed, it may be necessary, in further studies, to include security exchanges which pertain to confidentiality: in particular, in the frame of the exchange of private information between ATN end-systems and the certified systems which are involved in this private information management.

d) SESE PDU mapping: the SE-Transfer PDUs involved in the peer entities authentication (dirAuthenticationTwoWays exchanges) can be conveyed using ACSE association establishment PDUs: AARQ and AARE PDUs, for the initial peer entity authentication, then on P-DATA, for optional intermediate authentication and on RLRQ and RLRE, for the optional peer authentication during association release. The SE-Transfer PDU involved in data origin authentication and on data integrity will be mapped on P-DATA.

e) Any error condition encountered in the dirAuthenticationTwoWay security exchange results in the aborting of the application-association.

f) The presentation context used for transferring user data PDUs must employ a protecting transfer syntax with a protection mapping which meets the requirements of the User ASE.

8

9 SECURITY SERVICES

6.1 DEFINITION OF SERVICES

The services that are provided for the secure operation of the ATN ASEs, are peer-entity and data authentication and protection against replay. These services are optional and their use will be defined through the use of Dialogue services using additional parameters. The rationale for this selection of services in based on [Ref.1].

a) Peer authentication services will be provided by the Application layer based on the strong authentication procedures described in [Ref.10].

b) Data origin authentication will be provided by the Application layer, based on the SESE service invocation primitives SE-Transfer, as defined in [Ref.13], and on a security transformation to append a seal or signature.

c) Protection against replay will be provided by the Application layer, based on a security transformation to append a unique identifier to the message.

Five kinds of secured transfers have been envisaged via the dialogue services activation:

a) Unsecured service: no protection will be envisaged on the dialogue either for its establishment, or during the exchanges.

b) Secured Dialogue service: authentication will be used for the dialogue establishment and maintenance, but not during the exchanges.

c) Forward path secured application dialogue: authentication will be used for the dialogue establishment and maintenance, and for all exchanges issued by the initiator of the dialogue.

d) Return path secured application dialogue: authentication will be used for the dialogue establishment and maintenance, and for all exchanges issued by the acceptor of the dialogue.

e) Secured application dialogue: authentication will be used for the dialogue establishment and maintenance, and for all exchanges issued by the entities involved in the dialogue.

6.2 IMPACT ON DIALOGUE SERVICE

These information may be provided through the use of the Security requirements parameter of the D-START dialogue service. It will be able to give to the two communicating entities the possibility to negotiate the level of security for the dialogue. The following negotiation rules are proposed:

6.2.1 Unsecured dialogue establishment

If the initiator of the dialogue wishes to establish a dialogue where no security mechanisms will take place, it issues a D-START Request service primitive activation with the Security Requirements set to “Unsecured service”.

Upon reception of the D-START Indication service primitive activation, the responder will only have the following possibilities:

- Accept the unsecured dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to “Accepted” and with the Security Requirements set to “Unsecured service”.
- Refuse the unsecured dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to “Rejected (transient)” or “Rejected (permanent)”, with the Reject Source parameter set to “DS User” and with the Security Requirements set to “Unsecured service”, as specified in the initial request.

Upon reception of the D-START Confirmation service primitive activation, if the dialogue establishment has been accepted, the initiator checks that the Security Requirements parameter is the one specified on the request primitive and establishes an unsecured dialogue.

6.2.2 Secured dialogue establishment

If the initiator of the dialogue wishes to establish a dialogue where only peer entity authentication takes place, it issues a D-START Request service primitive activation with the Security Requirements set to "Secured service".
Upon reception of the D-START Indication service primitive activation, the responder will only have the following possibilities:

- Accept the secured dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Accepted" and with the Security Requirements set to "Secured service".
- Accept the dialogue establishment but refuses the security by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Accepted" and with the Security Requirements set to "Unsecured service".
- Refuse the unsecured dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Rejected (transient)" or "Rejected (permanent)", with the Reject Source parameter set to "DS User" and with the Security Requirements set to "Secured service", as specified in the initial request.

Upon reception of the D-START Confirmation service primitive activation, if the dialogue establishment has been accepted, the initiator checks the Security Requirements parameter. It then has the following possibilities:

- If the Security Requirements is set to "Secured service", then the secured dialogue is established.
- If the Security Requirement is set to "Unsecured service"
  - If the initiator does not accept to continue the dialogue in an unsecured environment, it issues a D-ABORT Request service primitive activation with the Originator parameter not specified or set to "provider", if the decision is taken by the ATN ASE or "user" if the decision is taken by the ATN Application user.
  - if the initiator accepts to continue the dialogue in an unsecured environment, then the secured dialogue is considered as established.

6.2.3 Forward path secured application dialogue establishment

If the initiator of the dialogue wishes to establish a dialogue where the peer entities authentication will take place and where only the data issued by the initiator will be protected, it issues a D-START Request service primitive activation with the Security Requirements set to "Forward path secured application dialogue".
Upon reception of the D-START Indication service primitive activation, the responder will only have the following possibilities:

- Accept the forward path secured application dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Accepted" and with the Security Requirements set to " to "Forward path secured application dialogue ".
- Accept the dialogue establishment but refuses the proposed security characteristics by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Accepted" and with the Security Requirements set to
  - "Unsecured service", if no security mechanism can be used or to
  - "Secured Dialogue service" if only peer entity authentication can be used.
- Refuse the forward path secured application dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Rejected (transient)" or "Rejected (permanent)", with the Reject Source parameter set to "DS User" and with the Security Requirements set to "Forward path secured application ", as specified in the initial request.

Upon reception of the D-START Confirmation service primitive activation, if the dialogue establishment has been accepted, the initiator checks the Security Requirements parameter. It then has the following possibilities:

- If the Security Requirements is set to " Forward path secured application dialogue", then the secured dialogue is established.
- If the Security Requirement is set to "Secured Dialogue service" and
  - if the initiator does not accept to continue the dialogue with this restriction, it issues a D-ABORT Request service primitive activation with the Originator parameter not specified or set to "provider", if the decision is taken by the ATN ASE, or "user" if the decision is taken by the ATN Application user.
  - if the initiator accepts to continue the dialogue in an environment which only permits peer entity authentication, then the secured dialogue is considered as established.
- If the Security Requirement is set to "Unsecured service" and
  - if the initiator does not accept to continue the dialogue in an unsecured environment, it issues a D-ABORT Request service primitive activation with the Originator parameter not specified or set to "provider", if the decision is taken by the ATN ASE, or "user" if the decision is taken by the ATN Application user.
  - if the initiator accepts to continue the dialogue in an unsecured environment, then the secured dialogue is considered as established.

6.2.4 Return path secured application dialogue establishment

If the initiator of the dialogue wishes to establish a dialogue where the peer entities authentication will take place and where only the data issued by the acceptor will be protected, it issues a D-START Request service primitive activation with the Security Requirements set to "Return path secured application dialogue".

Upon reception of the D-START Indication service primitive activation, the responder will only have the following possibilities:

- Accept the return path secured application dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Accepted" and with the Security Requirements set to " to "Return path secured application dialogue ".
- Accept the dialogue establishment but refuses the proposed security characteristics by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Accepted" and with the Security Requirements set to
  - "Unsecured service", if no security mechanism can be used or to
  - "Secured Dialogue service" if only peer entity authentication can be used.
- Refuse the return path secured application dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to "Rejected (transient)" or "Rejected (permanent)", with the Reject Source parameter set to "DS User" and with the Security Requirements set to "Return path secured application ", as specified in the initial request.

Upon reception of the D-START Confirmation service primitive activation, if the dialogue establishment has been accepted, the initiator checks the Security Requirements parameter. It then has the following possibilities:

- If the Security Requirements is set to " Return path secured application dialogue", then the secured dialogue is established.
- If the Security Requirement is set to "Secured Dialogue service" and
  - if the initiator does not accept to continue the dialogue with this restriction, it issues a D-ABORT Request service primitive activation with the Originator parameter not specified or set to "provider", if the decision is taken by the ATN ASE, or "user" if the decision is taken by the ATN Application user.
  - if the initiator accepts to continue the dialogue in an environment which only permits peer entity authentication, then the secured dialogue is considered as established.
- If the Security Requirement is set to "Unsecured service" and
  - if the initiator does not accept to continue the dialogue in an unsecured environment, it issues a D-ABORT Request service primitive activation with the Originator parameter not

specified or set to “provider”, if the decision is taken by the ATN ASE, or “user” if the decision is taken by the ATN Application user.

- if the initiator accepts to continue the dialogue in an unsecured environment, then the secured dialogue is considered as established.

6.2.5 Secured application dialogue establishment

If the initiator of the dialogue wishes to establish a dialogue where the peer entities authentication will take place and where the data issued by both entities will be protected, it issues a D-START Request service primitive activation with the Security Requirements set to “Secured application dialogue”.
Upon reception of the D-START Indication service primitive activation, the responder will only have the following possibilities:

- Accept the secured application dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to “Accepted” and with the Security Requirements set to “ to “Secured application dialogue ”.
- Accept the dialogue establishment but refuses the proposed security characteristics by issuing a D-START Confirmation service primitive activation with the Result parameter set to “Accepted” and with the Security Requirements set to
  - “Unsecured service”, if no security mechanism can be used or to
  - “Secured Dialogue service” if only peer entity authentication can be used.
  - “Forward path secured application dialogue” if peer entity authentication can take place and only data issued from the initiator will be protected,
  - “Return path secured application dialogue” if peer entity authentication can take place and only data issued from the responder will be protected,
- Refuse the return path secured application dialogue establishment by issuing a D-START Confirmation service primitive activation with the Result parameter set to “Rejected (transient)” or “Rejected (permanent)”, with the Reject Source parameter set to “DS User” and with the Security Requirements set to “Secured application dialogue”, as specified in the initial request.

Upon reception of the D-START Confirmation service primitive activation, if the dialogue establishment has been accepted, the initiator checks the Security Requirements parameter. It then has the following possibilities:

- If the Security Requirements is set to “Secured application dialogue”, then the secured dialogue is established.
- If the Security Requirement is set to “Secured Dialogue service”, “ Forward path secured application dialogue” or “Return path secured application dialogue” and
  - if the initiator does not accept to continue the dialogue with this restriction, it issues a D-ABORT Request service primitive activation with the Originator parameter not specified or set to “provider”, if the decision is taken by the ATN ASE, or “user” if the decision is taken by the ATN Application user.
  - if the initiator accepts to continue the dialogue with the responded restrictions, then the secured dialogue is considered as established.
- If the Security Requirement is set to “Unsecured service” and
  - if the initiator does not accept to continue the dialogue in an unsecured environment, it issues a D-ABORT Request service primitive activation with the Originator parameter not specified or set to “provider”, if the decision is taken by the ATN ASE, or “user” if the decision is taken by the ATN Application user.
  - if the initiator accepts to continue the dialogue in an unsecured environment, then the secured dialogue is considered as established.

6.3

6.4 SECURED ATN APPLICATION ENTITY

6.4.1 Overall architecture

The Figure 1 explains the addition of the security functionality in the ATN upper layers architecture where:

a) SESE is the Security Exchange Service Element as defined in [Ref.14] and [Ref.15].

b) System Security Object (SSO) is an object representing a set of related security functions, as defined in [Ref.11].

c) System Security Function (SSF) is a capability of an open system to perform security related processing as defined in [Ref.11].

Figure 1: Overview of secured ATN upper layers

6.4.2 Interactions

The figures 2 to 5 explain the interactions which take place between the various ASEs during the following phases of an association lifetime:

a) Association establishment

b) Association release

c) Association abort

d) Data transfer

6.4.2.1

6.4.2.2 Association establishment:

Initiator Receiver

Figure 2: Exchanges during association establishment

6.4.2.2.1 Initiator side

The ATN-App ASE issues a D-Start Request service activation (1) specifying the appropriate level of required security. If peer authentication is required (Secured dialogue service, or forward, or return path secured application dialogue or secured application dialogue is required), then an initial encoding of the user data is done in order to obtain an ASN.1 bit string representation of the data on which the security transformation is applied, in order to obtain a transformed item (XformedDataType). The SSF generates a signature (1.1) and (1.2). The transformed item, together with local information, is encoded according to the protecting transfer syntax.
A SE-TRANSFER APDU is built by invoking the SESE SE-Transfer Request service primitive which is mapped on the field authentication-value of the ACSE AC-Associate Request service primitive (2), using the ACSE Authentication FU.
A ACSE AARQ APDU is built by ACSE and provided as user information of the P-Connect request primitive activation (3) when establishing the application-association.
The presentation service provider then transfers the P-CONNECT request and receive a response.
The presentation provider issues a P-Connect confirmation primitive to ACSE (4), confirming the establishment of a presentation connection.
ACSE issues an AC-Associate confirmation primitive with the authentication-value field set to the SE-TRANSFER APDU in reply (5).
An ASN.1 decoding is performed, according to the protecting transfer syntax which provides a transformed item, which is then decoded by activating the appropriate SSF (1.1) and (1.2), and checked using the security information as described in chapter 6.
The resulting unprotected item is then decoded using initial encoding rules and is provided to the ATN-App ASE (6).

6.4.2.2.2 Receiver side

On the reception of the P-Connect indication, the AARQ APDU is provided to ACSE (4)
ACSE issues an AC-Associate indication primitive with the field authentication-value set to the SE-TRANSFER APDU built by the remote SESE (5).
An ASN.1 decoding is performed, according to the protecting transfer syntax which provides a transformed item, which is then decoded by activating the appropriate SSF (1.1) and (1.2), and checked using the security information as described in chapter 6.
The resulting unprotected item is then decoded using initial encoding rules and is provided to the ATN-App ASE (6).
The ATN-App ASE issues a D-Start Response service activation (1) specifying the appropriate level of required security. If peer authentication is required (Secured dialogue service, or forward or return path secured application dialogue, or secured application dialogue is required), then an initial encoding of the user data is done in order to obtain a bit string representation. Then, the security transformation is applied to the resulting bit string, in order to obtain a transformed item (XformedDataType). The SSF generates a signature (1.1) and (1.2). The transformed item together with local information is encoded according to the protecting transfer syntax.
A SE-TRANSFER APDU is built by invoking the SESE SE-Transfer Request service primitive, which is mapped on the field authentication-value of the ACSE AC-Associate Response service primitive (2), using the ACSE Authentication FU.
An ACSE AARE APDU is built by ACSE and provided as user information of the P-Connect response primitive activation (3) when accepting the application-association.

The presentation service provider then transfers the P-CONNECT response and the application association is established or rejected according to the ATN-App ASE response.

6.4.2.3

6.4.2.4 Association release:

Initiator Receiver

Figure 3: Exchanges during association release

6.4.2.4.1 Initiator side

The ATN-App ASE issues a D-Release Request service activation (1).
If data origin authentication and protection against replay is required for this dialogue (Forward path secured application dialogue or secured application dialogue is required), then an initial encoding of the user data is done in order to obtain a bit string representation on which the security transformation is applied, in order to obtain a transformed item (XformedDataType). The SSF generates a unique identifier and a signature (1.1) and (1.2). The transformed item, together with local information is encoded according to the protecting transfer syntax.
If data origin authentication and data integrity is required for this dialogue (Forward path secured application dialogue, or secured application dialogue is required), a SE-TRANSFER APDU is built by invoking the SESE SE-Transfer Request service primitive, which is mapped on the user information field of the AC-Release Request service primitive (2).
A ACSE RLRQ APDU is built by ACSE, and provided as user information of the P-Release request primitive activation (3).
The Dialogue control function maps the P-Release request service activation on a P-Data request service activation. The presentation service provider then transfers the P-DATA request.
The presentation provider issues a P-Data indication primitive with an ACSE RLRE PDU which is mapped by the Dialogue control function on a AC-Release confirmation service activation to ACSE (4), confirming the release of the association.
ACSE issues an AC-Release confirmation primitive with the user information field set to the SE-TRANSFER APDU in reply (5), and a P-Abort request primitive is issued (7) to the Presentation service provider.
An ASN.1 decoding is performed, according to the protecting transfer syntax which provides with a transformed item which is then decoded by activating the appropriate SSF (1.1) and (1.2) and by checking the security information as described in chapter 7.
The resulting unprotected item is then decoded using initial encoding rules and is provided to the ATN-App ASE (6).

6.4.2.4.2 Receiver side

On the reception of the P-Data indication, the RLRQ APDU is provided to ACSE (4)
ACSE issues an AC-Release indication primitive with the field user information set to the SE-TRANSFER APDU built by the remote SESE (5).
An ASN.1 decoding is performed, according to the protecting transfer syntax which provides a transformed item, which is then decoded by activating the appropriate SSF (1.1) and (1.2), and by checking the security information as described in chapter 6.
The resulting unprotected item is then decoded using initial encoding rules and is provided to the ATN-App ASE (6).
The ATN-App ASE issues a D-Release Response service activation (1). If data origin authentication is required (Secured dialogue service, or forward or return path secured application dialogue, or secured application dialogue is required), then an initial encoding of the user data is done in order to obtain a bit string representation. Then, the security transformation is applied to the resulting bit string, in order to obtain a transformed item (XformedDataType). A signature is then generated using the appropriate SSF (1.1) and (1.2). The transformed item together with local information is encoded according to the protecting transfer syntax.
A SE-TRANSFER APDU is built by invoking the SESE SE-Transfer Request service primitive, which is mapped on the field user information of the ACSE AC-Associate Response service primitive (2), using the ACSE.

A ACSE AARE APDU is built by ACSE and provided as user information of the P-Connect response primitive activation (3) when accepting the application-association.
The presentation service provider then transfers the P-CONNECT response in order to signify the peer entity of the result of the release.
If the release was successful, a P-PABORT indication will then be received (7) which will definitively close the dialogue.

6.4.2.5

6.4.2.6 Association abort:

Initiator Receiver

Figure 4: Exchanges during association abort

6.4.2.6.1 Initiator side

The ATN-App ASE issues a D-Abort Request service activation (1).
If data origin authentication and protection against replay is required for this dialogue (Forward path secured application dialogue or secured application dialogue is required), then an initial encoding of the user data is done in order to obtain a bit string representation on which the security transformation is applied, in order to obtain a transformed item (XformedDataType). The SSF generates a unique identifier and a signature (1.1) and (1.2). The transformed item, together with local information is encoded according to the protecting transfer syntax.
If data origin authentication and data integrity is required for this dialogue (Forward path secured application dialogue or secured application dialogue is required), a SE-TRANSFER APDU is built by invoking the SESE SE-Transfer Request service primitive which is mapped on the user information field of the AC-Abort Request service primitive (2).
A ACSE ABRT APDU is built by ACSE and provided as user information of the P-Abort request primitive activation (3).

6.4.2.6.2 Receiver side

On the reception of the P-Abort indication, the ABRT APDU is provided to ACSE (4)
ACSE issues an AC-Abort indication primitive with the field user information set to the SE-TRANSFER APDU built by the remote SESE (5).
An ASN.1 decoding is performed, according to the protecting transfer syntax which provides with a transformed item which is then decoded by activating the appropriate SSF (1.1) and (1.2), and by checking the security information as described in chapter 7.
The resulting unprotected item is then decoded using initial encoding rules and is provided to the ATN-App ASE (6).

6.4.2.7

6.4.2.8 Data transfer:

Figure 5: Exchanges during data transfer

6.4.2.8.1 Initiator side

The ATN-App ASE issues a D-Data Request service activation (1).
If data origin authentication and protection against replay is required for this dialogue (Forward path secured application dialogue or secured application dialogue is required), then an initial encoding of the user data is done in order to obtain a bit string representation on which the security transformation is applied, in order to obtain a transformed item (XformedDataType). The SSF generates a unique identifier and a signature (1.1) and (1.2). The transformed item, together with local information is encoded according to the protecting transfer syntax.
If data origin authentication and data integrity is required for this dialogue (Forward path secured application dialogue or secured application dialogue is required), a SE-TRANSFER APDU is built by invoking the SESE SE-Transfer Request service primitive which is mapped on the user information field of the P-Data Request service primitive (2).

6.4.2.8.2 Receiver side

Presentation issues an P-Data indication primitive with the field user information set to the SE-TRANSFER APDU built by the remote SESE (3).
An ASN.1 decoding is performed, according to the protecting transfer syntax which provides with a transformed item which is then decoded by activating the appropriate SSF (1.1) and (1.2) and by checking the security information as described in chapter 7.
The resulting unprotected item is then decoded using initial encoding rules and is provided to the ATN-App ASE (4).

7

8 PEER ENTITIES AUTHENTICATION

8.2 INTRODUCTION

Peer entities authentication takes place when Secured Dialogue (single, forward or return path or fully secured application dialogue) has been selected. Peer entities authentication mechanism is based on the support of SESE described in [Ref.13] and [Ref.14]. The SESE supports the transfer of security exchange items to carry information required for authentication and association establishment. The approach will be based on the one developed in [Ref. 10], and implements a two-way authentication scheme.
Peer entities authentication occurs at association establishment phase and, eventually, during exchange when renewal of the authentication is needed. During association establishment phase, it will be supported via ACSE association establishment, while during exchange it will be supported via Presentation data transfer.

8.3 DESCRIPTION

The following preliminary actions are considered to be verified:

a) The initiator of the authentication has obtained the public key of the receiver, together with the return certification path from the receiver to the initiator.
b) The initiator has checked the validity of all the certificates in the certification path.
c) Initiator and receiver clocks are synchronised by bilateral agreement, using Co-ordinated Universal Time. Such an assumption is fundamental if a two-way authentication scheme is to be used.

According to [Ref.10] description, the following steps are involved:
On the initiator side:

a) The initiator of the authentication generates a unique identifier (number) which will be used in order to detect replay attacks and to prevent forgery.
b) It sends to the receiver a message which contains:
    1) A timestamp used to check the time validity of the message,
    2) The unique number described in a),
    3) The Identification of the receiver,
    4) A signature of the message.

On the receiver side, upon the reception of the message:

a) The receiver of the authentication obtains the initiator public key and checks its validity,
b) Using the initiator public key, it verifies the signature and thus validates the integrity of the received information,
c) It checks that it is the intended recipient of the message,
d) It verifies the validity of the timestamp,
e) It checks the unique identifier validity, by verifying that it has not been duplicated, and that it has not yet expired.
f) It generates a unique identifier (number) used, as for the initiator part, in order to detect replay attacks and to prevent forgery.
g) Its sends a message to the initiator which contains
    1) A timestamp used to check the time validity of the token,
    2) The unique number described in f),
    3) The Identification of the receiver,
    4) The unique number sent by the initiator and described in a),
    5) A signature of the message.

On the initiator side, upon reception of the message:

a) Using the receiver public key, it verifies the signature and thus validates the integrity of the received information,
b) It checks that it is the intended recipient of the message,
c) It verifies the validity of the timestamp,
d) It checks the unique identifier validity, by verifying that it has not been duplicated and that it has not yet expired.

The availability of the entities public keys and of certificates can be based upon access to the directory as described in [Ref. 10]. The public keys will be obtained from a trusted third party acting as a certification authority which will issue a certificate that is unforgeable and that contains the required public key of the peer system.

8.4 SYSTEM SECURITY FUNCTIONS

The authentication mechanism mainly involves the application of system security functions which pertain to the computation of an authentifier or signature which will be appended to the clear information to be sent. The overall mechanisms consists of:

a) Prior to applying the security transformation, the data should be encoded as a bit string using PER in its canonical form.
b) Applying a hashing function to the data to be signed, in order to compress the data. The choice of hashing algorithm will be negotiated between entities at the dialogue establishment phase, with possible default value.
c) Computing a signature on the hashed data, which will be appended to the data which will be sent. The signature will be computed using the message sender private key and will be remotely checked using the sender public key. As for hashing, the choice of the algorithm will be negotiated between entities at the dialogue establishment phase, with possible default value.

No need has been expressed on confidentiality of the exchanged information during the peer entities authentication phase. Although this is true for ATN communications, it should be kept in mind that, during initial phases where private information is obtained (private key, for example), local legislation may have an impact on the way used to obtain this private information. For example, in the case where only certified trusted third parties can provide this information, the need for confidentiality on the data path can be raised. This is also true for renewal of private information.

8.5 SECURITY TRANSFORMATIONS

The specification of the security transformation conforms to the gulsSignedTransformation defined in [Ref.12]. This description has been preferred to the dirSignedTransformation defined in [Ref.10] and [Ref.12] because of the following reasons:

a) The dirSignedTransformation only supports the enciphered-hashed signature or sealing technique, while the gulsSignedTransformation does not have any restriction on the mechanisms which can be used for signing or sealing.
b) The dirSignedTransformation only support BER DER. The INITIAL-ENCODING-RULES field is defined as:

```
INITIAL-ENCODING-RULES
{
        joint-iso-ccitt asn1 (1) ber-derived (2) distinguished-encoding (1)
}
```

which is not the case with gulsSignedTransformation for which, although the INITIAL-ENCODING-RULES field is defined as:

```
INITIAL-ENCODING-RULES
{
        joint-iso-ccitt asn1 (1) ber-derived (2) canonical-encoding (1)
}
```

it should be considered as an initial or default value which can be overridden using a static protected parameter initEncRules defined as follow:

```
initEncRules      OBJECT IDENTIFIER DEFAULT
{
        joint-iso-ccitt asn1 (1) packed-encoding (3) canonical (1) unaligned (1)
}
```

c) The dirSignedTransformation does not permit to parameterise the security mechanism as does the gulsSignedTransformation which gives the possibility to provide with initial encoding rules (initEncRules, as previously explained), signature or sealing algorithm identification (signOrSealAlgorithm), hashing function (hashAlgorithm) and key information (keyInformation which permit to define different supported formats of keys).

d) The dirSignedTransformation restricts the digital signature and hashing processes to a single algorithm, which is not the case with the gulsSignedTransformation, which provides with two optional different AlgorithmIdentifier for specifying the signOrSealAlgorithm and the hashAlgorithm.

Conforming to the [Ref.12], the ASN.1 description of the signed security transformation is as follow:
In a first time, the initial encoding will be applied to the IntermediateType based on the initial encoding rules specified in the initEncRules field. Then, the resulting data will be signed eventually using the hash function specified in hashAlgorithm and then the signature algorithm specified in signOrSealAlgorithm will be applied, producing an appendix which, with the IntermediateType will constitute the transformed data.
The unprotected item can be of two types: derived from an ASN.1 syntax

```
IntermediateType {KEY-INFORMATION: SupportedKIClasses} ::= SEQUENCE
{
    unprotectedItem      ABSTRACT-SYNTAX.&Type,
    initEncRules         OBJECT IDENTIFIER DEFAULT
    {
        joint-iso-ccitt asn1 (1) packed-encoding (3) canonical (1) unaligned (1)
    },
    signOrSealAlgorithm    AlgorithmIdentifier OPTIONAL,
    hashAlgorithm          AlgorithmIdentifier OPTIONAL,
    keyInformation         SEQUENCE
    {
        kiClass    KEY-INFORMATION.&kiClass ({SupportedKIClasses}),
        keyInfo    KEY-INFORMATION.&KiType ({SupportedKIClasses} Error! Bookmark not defined.)
    } OPTIONAL
}

GulsSignedTransformation {KEY-INFORMATION: SupportedKIClasses}

SECURITY-TRANSFORMATION ::=
{
    IDENTIFIER                  {securityTransformations guls-signed (4)}
    INITIAL-ENCODING-RULES
    {
        joint-iso-ccitt asn1 (1) ber-derived (2) canonical-encoding (0)
    }
```

```
    X-FORMED-DATA-TYPE SEQUENCE
    {
        intermediateValue EMBEDDED PDV (WITH COMPONENTS {
            identification (WITH COMPONENTS
                        {transfer-syntax (CONSTRAINED BY {
                        -- This field will be set with the transfer syntax
                        -- corresponding to the initEncRules
                        }) PRESENT}),
            data-value (WITH COMPONENTS
                        {notation (IntermediateType {{SupportedKIClasses}})})
            }),
        appendix BIT STRING (CONSTRAINED BY {
                    -- This field will be set to the result of the
                    -- signing function.
        })
    }
}
```

8.6 SECURITY EXCHANGES

The dirAuthenticationTwoWay defined in [Ref.12] will be used for the exchanges involved in the peer entities authentication. As previously explained, it involves two security exchanges items between the entities needing to authenticate. In the frame of the ATN, both entity should be able to initiate an authentication exchange, meaning that both shall be able to play the initiator or responder role. This also means that both shall be able to issue the security exchange items: initiatorCredentials and responderCredentials.
The associated objects will conform to the following ASN.1 description:

```
DirAuthenticationTwoWay     SECURITY-EXCHANGE ::=
{
   SE-ITEMS                         {initiatorCredentials | responderCredentials }
   IDENTIFIER                   global : {securityExchanges dir-authent-two-way (2) }
}
initiatorCredentials        SEC-EXCHG-ITEM ::=
{
        ITEM-TYPE                   DirectoryAbstractService.Credentials
        ITEM-ID                     1
        ERRORS                      { authenticationFailure }
}
responderCredentials        SEC-EXCHG-ITEM ::=
{
        PARAMETER                   DirectoryAbstractService.SecurityProblem
        ERROR-CODE                  local : 1
}
```

9 PROTECTION OF DIALOGUE USER DATA

9.2 INTRODUCTION

The protection of the dialogue user data takes place when forward or return path or fully secured application dialogue has been selected. The provision for protection of protocol data will be based upon data origin authentication. No further need for protection of protocol data has been envisaged.
Data origin authentication mechanism is based on the support of SESE described in [Ref.13] and [Ref.14]. The SESE supports the exchanges of security exchange items to carry information required for data origin authentication.

9.3 DESCRIPTION

The following preliminary actions are considered to be applied:

a) The sender of the message has obtained the public key of the receiver, together with the return certification path from the receiver to the initiator.
b) The sender of the message has checked the validity of all the certificates in the certification path.
c) The receiver of the message has obtained the public key of the receiver, together with the return certification path from the receiver to the sender.
d) The receiver of the message has checked the validity of all the certificates in the certification path.
e) Sender and receiver clocks are synchronised by bilateral agreement, using Co-ordinated Universal Time.

According to [Ref.10] description, the following steps are involved:

On sender side:

a) The sender of the message generates a unique identifier (number) which will be used in order to detect replay attacks and to prevent forgery.
b) It sends to the receiver a message which contains:
   5) A timestamp used to check the time validity of the message,
   6) The unique number described in a)
   7) The user information coded as bit string using canonical PER,
   8) A signature of the message.

On the receiver side, upon the reception of the message:

a) Using the sender public key, it verifies the signature and thus validates the integrity of the received information,
b) It verifies the validity of the timestamp,
c) It checks the unique identifier validity, by verifying that it has not been duplicated and that it has not yet expired.

The availability of the entities public keys and of certificates can be based upon access to the directory as described in [Ref. 10]. The public keys will be obtained from a trusted third party acting as a certification authority and which will issue a certificate that is unforgeable and that contains the required public key of the peer system.

9.4 SYSTEM SECURITY FUNCTIONS

The data origin authentication and integrity check mechanisms mainly involve the application of system security functions which pertain to the computation of a unique identifier for the message and of an authentifier or signature which will be appended to the message to be sent.
The overall mechanisms consists of:

a) Prior to applying the security transformation, the data should be encoded using PER in its canonical form.
b) Calculate a unique identifier to for the message

c) Applying a hashing function to the data to be signed (message, unique identifier, security information), providing a reduced amount of data. The choice of hashing algorithm will be negotiated between entities at the dialogue establishment phase, with possible default value.

d) Computing a signature on the hashed data, which will be appended to the data which will be sent. The signature will be computed using the message sender private key and will be remotely checked using its public key. As for hashing, the choice of the algorithm will be negotiated between entities at the dialogue establishment phase, with possible default value.

No need has been expressed on confidentiality of the exchanged information during the data exchanges.

9.5 SECURITY TRANSFORMATIONS

The security transformations involved in the data authentication phase will be the same as the one described in 6.4, for peer entities authentication.

9.6 SECURITY EXCHANGES

The data origin authentication only requires to append a signature to the user information. This means that the whole ATN-App PDU will be protected via signature mechanism, provided that this has been required during the establishment of the connection.
The check for data integrity will be based on the use of a unique identifier appended to the user information, prior to apply signature.
The PDU can be specified as an ASN.1 type as follow:

```
SignedATNpdu ::= PROTECTED
{
        SEQUENCE
        {
                uniqueIdentifier        INTEGER,
                clearInfo               ABSTRACT-SYNTAX.&Type,
                signedInfo              BIT STRING
        }
}
```

where uniqueIdentifier is an integer which will be used in order to detect replay attacks and to prevent forgery, clearInfo is the user information pre-encoded using the initial encoding rules negociated during the association establishment and the signedInfo is set to the result of the signing function.

Annex 1: System management requirements

1 SCOPE

As described in [REF.16], an important aspect of security management concerns the security audit which permits to:

a) Evaluate the adequacy of security policy,
b) Aid in the detection of security violations
c) Facilitates the identification of individuals or of entities acting on their behalf
d) Assist in the detection of misuse of resources,
e) Act as a deterrent to individuals who might attempt to damage the system.

In the frame of security audit, two main management functions have been identified: security audit trail and alarm reporting.

2 OVERALL DESCRIPTION

The prevention of security violation and the operational procedures which should be followed in response are not part of security audit, which essentially concentrates on:

a) The detection of event s which can be considered as abnormal events in the frame of security,
b) The recording of such events,
c) The analysis of the collected events.

When a security violation has been detected a security alarm may be generated in order to permit system security manager to:

a) Implement immediate recovery actions,
b) Start further off-line analysis in order to modify security policy for further prevention…

Security management will provide objects and functions to support:

a) The access control policy in order to protect the Directory information from illegal use,
b) The monitoring of security threats to Directory information,
c) The monitoring of security threats to ATN communications,
d) The reporting of security violation,
e) The provision for security audit trail,
f) The provision for establishment and maintenance of credentials.

3 MODEL

Traditionally, security alarm and audit involve a set of different phases which are:

- The detection phase, in which a security related event is detected,
- A discrimination phase, where a first analysis of the event is made in order to determine if it has to be recorded (security audit trail) or if an alarm should be raised for this event (security alarm). It could lead to the following actions:
  - take no action, if the event is of few importance with respect to the security policy,
  - generate a security audit message, if the event is of interest to the security policy, but, still with respect to the security policy, has not reached a level which could justify the sending of an alarm,

- generate both a security audit message and an alarm if the event has to be reported to the manager and may need an immediate corrective action.
- Alarm processing phase, where an alarm is analysed and may lead to the following:
  - Take no action, if the security alarm is of few importance with respect to the security policy,
  - Initiate a recovery action, if the alarm can be recovered without needing to audit,
  - Initiate a recovery action and generate a security audit message, if the alarm should be part of an audit.
- Analysis phase, where the event is analysed and correlated to its environment (previous events…) in order to provide with a higher level of analysis which could lead to the following actions:
  - Take no action,
  - Generate a security alarm, for example because the evolution of the system does not conform to the security policy,
  - Generate a security audit record, if, for example, the evolution of the system does not justify the issuing of an alarm, but should anyway be logged,
  - Generate both security alarm and audit record.
- Aggregation phase which permit to collect security audit records from various systems into a single one.
- Report generation phase, which, when requested provides reports of the security audit trail. This could be used in order to detect attacks against the system and the impact of this attack on the resources of the system, and provide help in the determination of the recovery procedures which have to take place.
- Archiving phase which will permit to move security audit trail to long term archiving systems.

In a first approach, which will be refined by work in progress in the frame of network management in the ATN, all these phases will involve mechanisms defined in ITU|OSI document pertaining to network management:

- Logging of events, the definition of the security audit log destination and procedures for creation and retrieval of entries in the security audit trail log will conform to [Ref.20],
- Event forwarding discriminator used to forward logged events to different destinations and procedures for conveying event reports to the system where the security audit trail is located will conform to [Ref.19],

It should be noted that security audit and alarms are subject to the same need for security as exchanges between ATN ASEs: peer origin authentication and data origin and protection against replay should be applied to their transfer.
Both security audit trail and security alarm reporting functions will use the CMISE M-EVENT-REPORT service as defined in [Ref.17].

4 DEFINITIONS

The definition of the associated objects which need to be recorded in the security audit trail log will be based on the generic event log record object class defined in [Ref.22].
The definition of security alarm will be based on the templates of notifications defined in [Ref.22].

4.1 SECURITY ALARM REPORTING

4.1.1 Definition of alarms

The following types of alarms can be defined together with their potential causes:

- Integrity violation will report a security error in the transfer of information,
  - bad signature, meaning that an integrity attack may have occurred,
  - unique identifier invalid, meaning that the received information has been modified since sent,
  - unique identifier already used, meaning that a replay attack may have occurred,

- Operational violation will report if the requested security service could not be provided due to ATN ASE incapacity to provide the requested security characteristics,
  - Failure to establish a secured dialogue,
  - Report of establishment of a secured dialogue with different characteristics than the proposed one,
  - Invalid characteristic of secured dialogue,

- Security service or mechanism violation will report the detection of a security attack by one of the security services or mechanisms.
  - Peer entity authentication failure,
  - Data origin authentication failure.
- Time domain violation will report the use of an invalid time domain during authentication phase or during information exchange phases.
  - Abnormal delay in information transfer,
  - Key expiration.

Together with this characterisation of the alarm, its severity can be defined as:

- indeterminate,
- critical,
- major,
- minor.

The originator of the alarm will be specified, it will identify the mechanism (security system function, security exchange…) which detected the alarm.

The service provider will identify the ATN ASE which requested the service that led to the alarm.
No particular security alarm text nor security alarm data has been identified for each alarm. This may be the subject of further studies.

4.1.2 Definition of services

As already stated, the security alarm reporting service will use the CMISE M-EVENT-REPORT service. The following parameters will be used in M-EVENT-REPORT service activation:

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M(=)
Mode	M	-
Managed object class	M	U
Managed object instance	M	U
Security alarm type	M	C(=)
Event time	U	-
Security alarm information Security alarm cause	M	-
Security alarm	M	-

severity		
Security alarm generator	M	-
Service user	M	-
Service provider	M	-
Notification identifier	U	-
Correlated notifications	U	-
Problem text	U	-
Problem data	U	-
Current time	-	U
Event reply	-	C
Errors	-	C

With the following correspondence between parameters and management attributes:

Parameter	Attribute name
Managed object class	ObjectClass
Managed object instance	ObjectInstance
Security alarm type	EventType
Event time	EventTime
Security alarm cause	securityAlarmCause
Security alarm severity	securityAlarmSeverity
Security alarm generator	securityAlarmGenerator
Service user	serviceUser
Service provider	serviceProvider
Notification identifier	notificationId
Correlated notifications	correlatedNotifications
Problem text	securityAlarmText
Problem data	securityAlarmData

4.2 SECURITY AUDIT TRAIL

4.2.1 Definition of security audit trail notifications

The following type of notifications can be defined:

- Service report, which will log the use (provision, denial or recovery) of a service. It will lead to the provision of additional information in the event information part of the event; its value may be as follow:
  - Request for service, which means that the event has been generated following a request for use of a security service,
  - Denial of service, which means that the event has been generated following a request for use of a security service,
  - Response from service, which means that the event has been generated after the successful completion of a security service,
  - Service failure, which means that the event has been generated by the security service provider itself (the Security ASE),

- Service recovery: this value will not be used as not recovery procedure have been envisaged in case of security service failure.
- Other reason.

- Usage report, which will contain information on security (statistics…)

4.2.2 Definition of services

As already stated, the security audit trail service will use the CMISE M-EVENT-REPORT service.

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	P	P
Mode	P	-
Managed object class	P	P
Managed object instance	P	P
Event type	M	C(=)
Event time	P	-
Event information Service report cause	C	-
Notification identifier	U	-
Correlated notifications	U	-
Additional text	U	-
Additional information	U	-
Current time	-	P
Event reply	-	-
Errors	-	P

Annex 2: Need for canonical form of PER encoding/decoding

Basically, authentication consists in:
On emitter side:

- applying to the binary data which has to be transferred a hashing function which will produce another data string from which the original data can be always reconstructed and which is much smaller than the original string,
- encrypt the hashed string obtaining an authentifier or signature,
- append the clear data and the signature and transfer the result.

On receiver side:

- applying to the received clear binary data the same hashing function as the emitter,
- encrypt the hashed string obtaining an authentifier or signature,
- compare the result of the encryption and the received signature: if they are identical, the received data is the same as the one which was sent.

Two main problems are raised by the authentication process: a structural problem, which is comes from the independence of the OSI layers and a transfer problem which comes from the need for relays at the application layer level (MHS for example).

The structural problem comes from the fact that the hashing and encrypting functions belong to the application layer, which does not have knowledge of the actual encoding/decoding process which takes place in the Presentation layer. Various solutions have been proposed ranging from calculating the signature without including presentation information, or locating the security related functions in the presentation layer, or, else proposing the definition of interactions between application and presentation layers.

The utilisation of relays at the application layer raise another problem which could be stated as follow:
The security is an end-to-end process: the negotiation and the sharing of public security related information only involve the peer entities which are involved in the dialogue.
The use of relays at the application level may lead to decoding/encoding process by the relays using different rules (different options of encoding due to different implementations, or different choices of encoding variant). This would lead to the transfer of a new data stream which would not conform to the initial signature.

The proposed solution for the ATN upper layers conforms to the one defined in X.509 [Ref.10]. It has the two main characteristics:

- it is based on a pre-encoding of the clear information which has to be sent in a binary string, which removes the needs of the application layer to have the knowledge of the encoding/decoding rules which are applied in the presentation layer.
- The pre-encoding process will be based on canonical variant of PER.

The main consequences of this solution are:

- a need for additional encoding/decoding of user information: one at the application layer using canonical variant of encoding rules and one at the presentation layer, using the negotiated transfer syntax.
- the removal of the need for ATN ASEs to be aware of the variant of encoding/decoding needed for security mechanisms. This will ensure that older versions of these ASEs will still be compatible with security mechanisms.