AERONAUTICAL TELECOMMUNICATION NETWORK PANEL (ATNP)
WG3 – Applications and upper layers
SG3 – Upper Layer Architecture
Gran Canaria, September the 27th, October the 8th, 1999

Proposed mapping of WG1/SG2 security requirements on upper layers mechanisms

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 presents an overall description of the mechanisms involved in the ATN upper layers, by the implementation of the security in the ATN. It is intended to serve as a basis for the upper layers SARPs chapter 4.8 which specifies the implementation of security in the ATN Upper Layers.

1 INTRODUCTION

This document describes how the requirements provided to WG3/SG3 by WG1/SG2 can be mapped to upper layers mechanisms. Only those requirements which can be fulfilled by the Dialogue control function or the security ASO have been translated.
The documents used input are the following requirement documents:
Application Security Solution for the Aeronautical Telecommunication Network (WG1/SG2/WP1406a)
Upper Layer Authentication using Hybrid Elliptic Curve Public Key and HMAC Symmetric Key Approach (WG1/SG2/WP1308) and its last update issued after WG1/SG2 Atlantic meeting in July 1999.

2 OVERVIEW

The implementation of security in the ATN upper layers involves two main mechanisms:

- the first one intends at setting agreed information between the two communicating entities. One of the most important information being the key which will be used for subsequent exchanges and the information needed for key computation at the two end systems.
- The second one uses the previously agreed and established security information to secure the exchanges which take place between various communicating entities.

The architecture which has been foreseen in order to implement these exchanges is based on the addition, in the ATN Application Layer of a specific ASO which is in charge of security transformations and exchanges: the Security ASO.
It contains:

- A Security Exchange ASE, which supports all the exchanges of information specific to security,
- A System Security Object, which incorporates all basic security related functions (Hashing, Key derivation…),
- A Control Function .

This architecture conforms with the model described in the OSI/ITU-T documents serie which describes the GULs.

3 SOLUTION OPERATION

This section describes the main mechanisms involved in the upper layers and should be read together with its reference documents. In order to provide link with the requirements, the associated chapter of "Application Security Solution for the ATN" for each set of basic actions.

3.1 Dialogue user supporting key management exchanges

3.1.1 Requirement:

The aircraft entity initiates secure communication with a CM domain by performing CMA login as follows.

1. The aircraft entity CMA forms a CMA login request message $Data_1$ consisting of some login data including an indication that it is requesting a secure connection, its identity *U*, the ground CMA entity's identity *V*, a time field T_U, and a field *Addresses* containing the identities of the ground application entities within the CM domain that it wishes to communicate with. It signs $Data_1$ using ECDSA under its private signing key $d_{sig,U}$:

   $$s_U = Sign\ (d_{sig,U};\ Data_1)$$

   The aircraft entity sends the CMA login request message along with s_U to the ground CMA. The aircraft entity retains its signature s_U for use later.

   The signature on this message confirms the identity of the aircraft to the ground CMA. Inclusion of T_U in the message ensures that it is fresh, while inclusion of *U* and *V* confirms the intended source and recipient. The aircraft must sign this message rather than MACing it under a session key because it does not yet possess the ground CMA entity's public key and is therefore not yet able to compute the session key.

   This calculation employs the ECDSA signing transformation specified in Section 5.9.1 of ANSI X9.63 [2].

3.1.2 Solution proposal:

3.1.2.1 Initiator side: Request phase

a) The Dialogue user issues a D-START request at the dialogue interface, with the Security Requirements parameter set to the abstract value “Exchange supporting key management” and the User Data containing the user data provided by the user of the dialogue service.

b) The Dialogue control function maps the D-START request on a SA-START request at the Security ASO interface, with the following parameters:
- The identification of the Calling Peer ID,
- The identification of the Called Peer ID,
- The D-START request user data

c) Upon reception of the SA-START request, the Security ASO control function:
- retrieves from the Calling Peer ID the identification of the algorithm, which contains the identifier of the algorithm together with its parameters,

- Activates the SSO function SSO-Signature to compute the signature on the user data provided in the SA-START request, using the following parameters:
  - ECDSA as the algorithm identifier,
  - the Calling Peer ID,
  - the Called Peer ID,
  - the User Data provided in the SA-START request.

- Activates the SESE SE-TRANSFER request, using the following parameters:
  - Security exchange identifier set to the identifier of the atnEtablishSE security exchange.
  - Invocation identifier set to an unambiguous identifier of the exchange.
  - Security exchange item contains the signature computed using the SSO-Signature function, under the form of an algorithm identifier together with encrypted data, and the time stamp
  - Item identifier set to the value 1, which is the identifier of the atnEstablish security exchange item.
  - Start flag set to true in order to specify that this is the first request of the exchange.
  - End flag set to false.

- Transfer the resulting SETR PDUs to the dialogue control function,

d) Upon reception of the request to send the SETR PDU, the Dialogue control function:
   a) Retrieves the AE-qualifier as defined for the ATN-App AE,

   b) Constructs the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and zero otherwise,

   c) Looks up the calling presentation address,

   d) Looks up the called presentation address from the Called Peer Id parameter,

   e) Retrieves the Calling AP Title and Calling AE-Qualifier from the Calling Peer Id parameter.

   f) Retrieves the Called AP Title and Called AE-Qualifier from the Called Peer Id parameter.

   g) Sets the ACSE Requirements parameter to the symbolic value “Authentication functional unit ”,

   h) Maps the Security Requirements value to the A-ASSOCIATE Authentication mechanism name parameter.

i) Maps the SETR PDU to the A-ASSOCIATE Authentication value parameter,

j) Construct an A-ASSOCIATE Request primitive with the following parameters:

A-ASSOCIATE Request parameter	**ISO Status**	**ATN Status**
Mode	U	Not used (default value)
Application Context Name	M	As derived in b) above
Application Context Name List	C	Not used
Calling AP Title	U	As derived in e) above
Calling AE Qualifier	U	As derived in e) above
Calling AP Invocation Identifier	U	Not used
Calling AE Invocation Identifier	U	Not used
Called AP Title	U	As derived in f) above
Called AE Qualifier	U	As derived in f) above
Called AP Invocation Identifier	U	Not used
Called AE Invocation Identifier	U	Not used
ACSE Requirements	U	As derived in g) above
Authentication mechanism name	U	As derived in h) above
Authentication value	U	As derived in i) above
User Information	U	D-START User Data Parameter
Calling Presentation Address	M	As derived in c) above
Called Presentation Address	M	As derived in d) above
Presentation Context Definition List	U	Not used
Default Presentation Context Name	U	Not used
Quality of Service	M	See SV4
Presentation Requirements	U	Not Used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not Used
Session-connection identifier	U	Not used

k) Invoke the A-ASSOCIATE Request primitive.

3.1.3 Requirement:

2. The ground CMA entity receives the CMA login request message along with s_U and sees that the message requests a secure connection. It recovers from the message the aircraft entity's identity *U*, its identity *V*, the time field T_U, and the application entity identities. The ground CMA entity then retrieves the public key certificates of aircraft entity *U*, the public key certificates of the specified ground application entities, and the current CRL from the publicly accessible certificate directory. It checks none of the public key certificates have been revoked using the CRL, and it verifies the public key certificates are valid using its copy of the CA's public key $Q_{sig,CA}$. It retrieves the aircraft entity's public signing key $Q_{sig,U}$ from its signature key certificate. It verifies the received signature s_U is a valid ECDSA signature on $Data_1$ using $Q_{sig,U}$, and it verifies that *V* is correct and T_A corresponds with the current time. (Note that the ground CMA entity may optionally cache s_A and use it to check that CMA login requests are not replayed even during a single time period.) If all these checks are successful, the ground CMA entity accepts the aircraft entity's login request.

   These calculations employ the ECDSA verifying transformation as specified in Section 5.9.2 of ANSI X9.63 [2].

3. The ground CMA entity calculates a shared public value $X_{U,V}$ which will be sent to other application entities in the CM domain and used to ensure that the application entities use session keys which are unique to this CMA session. To do this it first retrieves the aircraft entity's key agreement key $Q_{s,U}$ from its key agreement key certificate. Then it calculates the shared secret value $Z_{U,V}$ from the *x*-coordinate of the point $d_{s,V}Q_{s,U}$ using its private key agreement key $d_{s,V}$ as specified in ANSI X9.63 [2]. Then it selects a random challenge $Rand_V$. Finally it computes the 80-bit $X_{U,V}$ using the SHA-1 based ANSI X9.63 key derivation function from $Z_{U,V}$, the single octet 00_{16}, s_U, and $Rand_V$ as:

$$X_{U,V} = KDF\ (Z_{U,V};\ 80;\ 00_{16} \parallel s_U \parallel Rand_V)$$

   Inclusion of s_U and $Rand_V$ in the generation of $X_{U,V}$ ensures that $X_{U,V}$ is unique to this CMA session while inclusion of the octet 00_{16} ensures that $X_{U,V}$ is distinct from the session key $MacKey_{U,V}$ computed later on.

   This calculation employs the static unified model key agreement scheme specified in Section 6.3 of ANSI X9.63 [2].

4. The ground CMA entity next calculates the CMA session key $MacKey_{U,V}$. To do this it computes the 80-bit $MacKey_{U,V}$ using the SHA-1 based ANSI X9.63 key derivation function from the shared secret value $Z_{U,V}$, the single octet 01_{16}, the shared public value $X_{U,V}$, an indication *CMA* of the application the session key is for, the aircraft entity identity *U*, and the ground CMA entity identity *V* as:

$$MacKey_{U,V} = KDF\ (\ Z_{U,V};\ 80;\ 01_{16} \parallel X_{U,V} \parallel CMA \parallel U \parallel V)$$

   Including $X_{U,V}$ in the key derivation process ensures the session key $MacKey_{U,V}$ is unique to this CMA session. Including the single octet 01_{16} ensures the session key $MacKey_{U,V}$ is distinct from the shared public value $X_{U,V}$ computed earlier. Finally including *CMA*, *U*, and *V* ensures that the session key is specific to a CMA session between *U* and *V*.

   This calculation also employs the static unified model key agreement scheme specified in Section 6.3 of ANSI X9.63 [2].

3.1.4 Solution proposal:

3.1.4.1 Receiver side: Indication phase

a) The ACSE Protocol Machine (ACPM) invokes the A-ASSOCIATE Indication primitive with:
   - the Calling and Called AP Titles set,
   - the ACSE Requirements parameter set to the symbolic value "Authentication

functional unit ",
- the Authentication mechanism name parameter set to the abstract value "Exchange supporting key management",
- the Authentication value parameter set to a SETR PDU,

b) The Dialogue control function:
- extracts the Calling Peer Id from the Calling AP Title,
- extracts the Called Peer Id from the Called AP Title,
- extracts the Dialogue user data, from the A-ASSOCIATE Indication user information field,
- extract the SESE PDU from the A-ASSOCIATE Indication Authentication value field,
- provides the SESE PDU to the Security ASO, together with the Calling and Called Peer Id and the Dialogue user data,

c) Upon reception of the SESE PDU, the Security ASO control function provides it to the SESE Protocol Machine (SEPM),

d) The SEPM activates the SE-TRANSFER indication, using the following parameters, as set by the peer SEPM:
- Security exchange identifier set to the identifier of the atnEstablishSE security exchange.
- Invocation identifier set to an unambiguous identifier of the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, under the form of an algorithm identifier together with encrypted data, and a time stamp,
- Item identifier set to the value of the atnEstablish security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to false.

e) Upon reception of the SE-TRANSFER indication, the Security ASO control function checks the signature is valid by activating the SSO-SignCheck function, with the atnSignCheckST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,
- the User Data provided in the SA-START request.
- The signature together with the algorithm identifier contained in the atnEstablish security exchange item.

The SSO-SignCheck function is in charge of the following:
- read, from the ATN Directory, the information associated to the Calling Peer ID, by selecting its signed Public Key Certificate attribute, under the following form:
  - version of the certificate,
  - certificate serial number,
  - the identification of the algorithm used for signing the certificate,
  - the name of the certification authority which issued the certificate,
  - the period of validity of the certificate,
  - the identity of the Calling Peer Id,
  - the public key of the Calling Peer Id,
- read, from the ATN Directory, the Certificate Revocation List,
- check that the Certificate Revocation List does not contain, in its list of revoked certificates, the serial number of the certificate associated with the Calling Peer ID,
- check the validity of the Calling Peer ID certificate, using the following parameters:
  - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
  - The Certificate Authority public key, as signing key,
  - The public key of the Calling Peer Id as clear data,
  - The signature of the certificate.
- verify that the time stamp corresponds to a valid period,
- check the validity of the signature contained in the atnEstablish security exchange item,

using the following parameters:
- The algorithm identifier included in the received signature,
- The public key of the Calling Peer Id,
- The Calling Peer ID,
- The Called Peer ID,
- The User Data provided in the SA-START request.
- The signature contained in the atnEstablish security exchange item.

f) the Security ASO control function triggers the computation of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,
- the User Data provided in the SA-START request.
- The signature together with the algorithm identifier contained in the atnEstablish security exchange item.

The SSO-SessionKey function is in charge of the following:
- read, from the ATN Directory, the information associated to the Called Peer ID, by selecting its signed key agreement certificate attribute, under the following form:
  - version of the certificate,
  - certificate serial number,
  - the identification of the algorithm used for signing the certificate,
  - the name of the certification authority which issued the certificate,
  - the period of validity of the certificate,
  - the identity of the Called Peer Id,
  - the public key of the Called Peer Id,
- compute a random number,
- compute the shared secret value,
- compute a shared public value using key derivation mechanisms, with the following parameters:
  - The abstract value associated to SHA-1 as the algorithm identifier,
  - The shared secret value,
  - The signature contained in the atnEstablish security exchange item initially received,
  - The random number,
- compute the session key which will be used during the exchange, using key derivation mechanisms with the following parameters:
  - The abstract value associated to SHA-1 as the algorithm identifier,
  - The shared secret value,
  - The shared public value,
  - The Calling Peer ID,
  - The Called Peer ID

g) the Security ASO control function issues a SA-START indication with the following parameters:
- The identification of the Calling Peer ID,
- The identification of the Called Peer ID,

h) Upon reception of the SA-START indication, the Dialogue control function issues a D-START indication with:
- the Security requirements parameter set to the abstract value received in the Authentication mechanism name of the A-ASSOCIATE indication,
- the user data set to the user information of the A-ASSOCIATE indication,

3.1.5 Requirement:

5. The ground CMA entity forms a CMA login response message $Data_2$ including an indication that it has accepted the aircraft entity's request for a secure connection, the aircraft entity identity *U*, the ground CMA entity identity *V*, the ground CMA entity's random challenge $Rand_V$, and the ground application entity identities and public keys that the aircraft entity requested. It calculates the tag on its identity *V*, a sixteen bit counter value *Count* which is initially 0001_{16}, $Data_2$ and s_U using HMAC under the CMA session key $MacKey_{U,V}$ as:

$$MAC\ (MacKey_{U,V};\ V \parallel Count \parallel Data_2 \parallel s_U)$$

The ground CMA entity sends the CMA login response $Data_2$ along with its key agreement certificate $Cert(V;\ Q_{s,V})$ and the tag to the aircraft entity.

The tag on this message confirms the identity of the ground CMA entity to the aircraft entity. Inclusion of the 'random challenge' s_U in the MAC ensures that the response is fresh, inclusion of the ground CMA entity's identity *V* and the aircraft entity's identity *U* confirms the intended source and recipient, inclusion of the ground CMA entity's random challenge $Rand_V$ enables the aircraft entity to compute the shared public value $X_{U,V}$, inclusion of the ground application entity identities and public keys the aircraft entity requested transfer these keys securely to the aircraft entity, and inclusion of the counter prevents replay of messages secured under $MacKey_{U,V}$. The ground CMA entity MACs the response message instead of signing it to save bandwidth.

[[Note that *Count* is incremented each time a message is sent from *V* to *U* authenticated under $MacKey_{U,V}$. *Count* must therefore be maintained across CMA dialogues within a CMA session along with $MacKey_{U,V}$. *Count* may either be sent along with secured messages, or may be inferred from previous communications. Inference of *Count* is preferred in order to save bandwidth.]]

3.1.6 Solution proposal:

3.1.6.1 Receiver side: Response phase

a) The Dialogue user issues a D-START response at the dialogue interface, with the Security Requirements parameter set to the abstract value "Exchange supporting key management" and the User Data containing the user data provided by the user of the dialogue service.

b) The Dialogue control function maps the D-START response on a SA-START response at the Security ASO interface, with the following parameters:
   - The identification of the Calling Peer ID,
   - The identification of the Called Peer ID,
   - The D-START response user data

c) Upon reception of the SA-START response, the Security ASO control function activates the SSO function SSO-GetPublicKeyCertificate to retrieve the public key certificate of the called peer identifier provided in the SA-START response, using the Called Peer ID as parameter,

The SSO- GetPublicKeyCertificate function is in charge of the following:
- read, from the ATN Directory, the information associated to the Called Peer ID, by selecting its signed Public Key Certificate attribute, under the following form:
  - version of the certificate,
  - certificate serial number,
  - the identification of the algorithm used for signing the certificate,
  - the name of the certification authority which issued the certificate,
  - the period of validity of the certificate,
  - the identity of the Called Peer Id,
  - the public key of the Called Peer Id,
- read, from the ATN Directory, the Certificate Revocation List,

- check that the Certificate Revocation List does not contain, in its list of revoked certificates, the serial number of the certificate associated with the Called Peer ID,
- check the validity of the Called Peer ID certificate, using the following parameters:
  - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
  - The Certificate Authority public key, as signing key,
  - The public key of the Called Peer Id as clear data,
  - The signature of the certificate.

d) the Security ASO control function activates the SSO function SSO-Signature to compute a Signature on the user data provided in the SA-START response, using the following parameters:
   - HMAC as the algorithm identifier,
   - the Calling Peer ID,
   - the Called Peer ID,
   - the User Data provided in the SA-START response.

e) the Security ASO control function activates the SESE SE-TRANSFER request, using the following parameters:
   - Security exchange identifier set to the identifier of the atnEstablishSE security exchange.
   - Invocation identifier set to an unambiguous identifier of the exchange.
   - Security exchange item containing the signature computed using the SSO-Signature function, under the form of an algorithm identifier together with encrypted data, the random number, the Calling Peer ID, the Called Peer ID key agreement certificate,
   - Item identifier set to the value of the antEstablish security exchange item.
   - Start flag set to true in order to specify that this is the first request of the exchange.
   - End flag set to true.

f) the Security ASO control function transfers the SESE PDUs to the dialogue control function,

g) Upon reception of the request to send the SESE PDU, the Dialogue control function:
   a) Construct the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and set to zero otherwise,
   b) Retrieves the responding Presentation address,
   c) Retrieves the Responding AP Title and Responding AE-Qualifier from the Called Peer Id parameter.
   d) Sets the ACSE Requirements parameter to the symbolic value “Authentication functional unit ”,
   e) Maps the Security Requirements value to the A-ASSOCIATE Authentication mechanism name parameter, together with the session key,
   f) Maps the SETR PDU to the A-ASSOCIATE Authentication value parameter,
   g) Construct an A-ASSOCIATE Response primitive with the following parameters:

A-ASSOCIATE Response parameter	**ISO Status**	**ATN Status**
Application Context Name	M	As derived in a) above
Application Context Name List	C	Not used
Responding AP Title	U	As derived in c) above
Responding AE Qualifier	U	As derived in c) above
Responding AP Invocation Identifier	U	Not used
Responding AE Invocation Identifier	U	Not used
ACSE Requirements	U	As derived in d) above
Authentication mechanism name	U	As derived in e) above
Authentication value	U	As derived in f) above
User Information	U	D-START User Data Parameter
Result	M	D-START Result parameter
Diagnostic	U	Not used
Responding Presentation Address	M	As derived in b) above
Presentation Context Definition List	C	Not used

Default Presentation Context Result	C	Not used
Quality of Service	M	As for D-START Request Request (see preceding section)
Presentation Requirements	U	Not Used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not Used
Session-connection identifier	U	Not used

h) If the D-START Response *Result*, parameter has the abstract value “accepted”, invoke an A-ASSOCIATE Response primitive with the Result parameter set to “accepted”, and remain in the ASSOCIATION PENDING state,

i) If the D-START Response *Result* parameter has the abstract value “rejected (permanent)” or “rejected (transient)”, invoke an A-ASSOCIATE Response primitive with the Result parameter set to the same abstract value.

3.1.7 Requirement:

6. The aircraft entity CMA receives the CMA login response along with the tag and $Cert(V; Q_{s,V})$, and sees that the message accepts a secure connection. It recovers from the CMA login response the ground CMA entity's identity V, it's identity U, the application entities' public keys, and the random challenge $Rand_V$. It verifies the public key certificate is valid using its copy of the CA's public key $Q_{sig,CA}$. It checks the received copy of its identity U is correct. It retrieves the ground CMA entity's public key agreement key $Q_{s,V}$ from the certificate.

   The aircraft entity CMA then calculates the shared public value $X_{U,V}$ which will be sent to other applications invoked during this CMA session and used to ensure that the applications use session keys which are unique to this CMA session. To do this it calculates the shared secret value $Z_{U,V}$ from the x-coordinate of the point $d_{s,U}Q_{s,V}$ using its private key agreement key $d_{s,U}$ as specified in ANSI X9.63 [2]. Then it computes the 80-bit $X_{U,V}$ using the SHA-1 based ANSI X9.63 key derivation function from $Z_{U,V}$, the single octet 00_{16}, s_U, and the received $Rand_V$ as:

   $$X_{U,V} = KDF\ (Z_{U,V};\ 80;\ 00_{16} \| s_U \| Rand_V)$$

   The mathematical properties of elliptic curves ensure that the value of $Z_{U,V}$ computed by the aircraft entity CMA is the same as the value of $Z_{U,V}$ computed earlier by the ground CMA entity (and hence the value $X_{U,V}$ is also the same).

   This calculation employs the ECDSA verification transformation specified in Section 5.9.2 and the static unified model key agreement scheme specified in Section 6.3 of ANSI X9.63 [2].

7. The aircraft entity CMA next calculates the CMA session key $MacKey_{U,V}$. To do this it computes the 80-bit $MacKey_{U,V}$ using the SHA-1 based ANSI X9.63 key derivation function from the shared secret value $Z_{U,V}$, the single octet 01_{16}, the shared public value $X_{U,V}$, an indication CMA of the application the session key is for, the aircraft entity identity U, and the ground CMA entity identity V as:

   $$MacKey_{U,V} = KDF\ (\ Z_{U,V};\ 80;\ 01_{16} \| X_{U,V} \| CMA \| U \| V)$$

   Again the mathematical properties of elliptic curves ensure that the value of $MacKey_{U,V}$ computed by the aircraft entity CMA is the same as the value of $MacKey_{U,V}$ computed earlier by the ground CMA entity.

   This calculation also employs the static unified model key agreement scheme specified in Section 6.3 of ANSI X9.63 [2].

8. Finally the aircraft entity CMA reconstructs $V \| Count \| Data_2 \| s_U$ and checks the tag it received is valid using HMAC with SHA-1 under the session key $MacKey_{U,V}$. If this check is successful, the aircraft entity CMA accepts the ground CMA entity's login response and concludes that the CMA login security check has been successful.

3.1.8 Solution proposal:

3.1.8.1 Initiator side: Confirmation phase

a) The ACSE Protocol Machine (ACPM) invokes the A-ASSOCIATE Confirmation primitive with:

- the Result parameter set to the abstract value "accepted",
- the Responding AP Title set,
- the ACSE Requirements parameter set to the symbolic value "Authentication functional unit ",
- the Authentication mechanism name parameter set to the abstract value "Exchange supporting key management",

- the Authentication value parameter set to a SESE PDU,

b) The Dialogue control function:
- extracts the Called Peer Id from the Responding AP Title,
- extract from the Authentication value the SESE PDU,
- provides the SESE PDU to the Security ASO, together with the Called Peer Id, the Calling Peer Id, the SESE PDU and the A-ASSOCIATE user information,

c) Upon reception of the SESE PDU, the Security ASO control function provides it to the SEPM,

d) The SESE Protocol Machine (SEPM) activates the SE-TRANSFER indication, using the following parameters, as set by the peer SEPM:
- Security exchange identifier set to the identifier of the atnEstablishSE security exchange.
- Invocation identifier set to an unambiguous identifier of the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, under the form of an algorithm identifier together with encrypted data, the random number, the Calling Peer ID and the Called Peer ID key agreement certificate,
- Item identifier set to the value of the atnEstablish security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

e) The Security ASO control function triggers the computation of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,
- The signature together with the algorithm identifier contained in the atnEstablish security exchange item.
- The certificate of the Called Peer ID contained in the atnEstablish security exchange item.
- the User Data provided in the SA-START request.

The SSO-SessionKey function is in charge of the following:
- retrieve its CA's public key
- check the validity of the Called Peer ID certificate, using the following parameters:
  - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
  - The Certificate Authority public key, as signing key,
  - The public key of the Calling Peer Id as clear data,
  - The signature of the certificate.
- compute a random number,
- compute the shared secret value,
- compute a shared public value using key derivation mechanisms, with the following parameters:
  - The abstract value associated to SHA-1 as the algorithm identifier,
  - The shared secret value,
  - The signature contained in the atnEstablish security exchange item initially received,
  - The random number,
- compute the session key which will be used during the subsequent exchanges, using key derivation mechanisms with the following parameters:
  - The abstract value associated to SHA-1 as the algorithm identifier,
  - The shared secret value,
  - The shared public value,
  - The Calling Peer ID,
  - The Called Peer ID

f) the Security ASO control function activates the SSO function SSO-SignCheck to check the validity of the Called Peer ID key agreement certificate, using the following parameters:
- the Calling Peer ID,

- the Called Peer ID,
- the User Data,
- The signature together with the algorithm identifier contained in the atnEstablish security exchange item.

The SSO-SignCheck function is in charge of the following:
- verify that the time stamp corresponds to a valid period,
- check the validity of the signature contained in the atnEstablish security exchange item, using the following parameters:
  - The algorithm identifier included in the received signature,
  - The public key of the Calling Peer Id,
  - The Calling Peer ID,
  - The Called Peer ID,
  - The User Data provided in the SA-START request.
  - The signature contained in the atnEstablish security exchange item.

g) the Security ASO control function issues a SA-START confirmation with the following parameters:
- The identification of the Calling Peer ID,
- The identification of the Called Peer ID,

h) Upon reception of the SA-START confirmation, the Dialogue control function issues a D-START confirmation with:
- the Security requirements parameter set to the abstract value received in the Authentication mechanism name of the A-ASSOCIATE confirmation,
- the user data set to the user information of the A-ASSOCIATE confirmation,

3.1.9 Requirement:

Subsequent to CMA login when the ground CMA entity and the aircraft entity CMA wish to exchange a data item $Data_3$, such as a request for additional application entity public keys, the sender authenticates the data item by MACing it using the CMA session key $MacKey_{U,V}$ established during CMA login as follows:

$$MAC\ (\ MacKey_{U,V};\ Source\ ||\ Count\ ||\ Data_3\ )$$

Source contains the sender's identity. Inclusion of *Source* enables detection of attacks in which an adversary bounces a data item sent by the aircraft or ground back to its sender. *Count* is a 16-bit counter indicating the number of data items sent over the RF channel by its sender secured under $MacKey_{U,V}$ during this CMA session. Inclusion of *Count* enables detection of attacks in which an adversary replays a previously sent data item.

The data item $Data_3$ is then sent over the RF channel along with the MAC value calculated on it. This ensures the authenticity of $Data_3$.

[[Note that as before *Count* may either be sent along with secured messages, or may be inferred from previous communications. Inference of *Count* is preferred in order to save bandwidth.]]

This CMA communication procedure is based on the HMAC scheme which is specified in the international standard IETF RFC 2104 [6], and the US standard ANSI X9.71 [3].

3.1.10 Solution proposal:

3.1.10.1 Remaining data transfer: emitter side

a) The Dialogue user issues a D-DATA Request at the dialogue interface, with the User Data containing the user data provided by the user of the dialogue service.

b) The Dialogue control function maps the D-DATA Request on a SA-SEND Request at the Security ASO interface, with the following parameters:
- the identification of the Calling and Called Peer IDs,
- The D-DATA Request user data,

c) The Security ASO control function triggers the retrieval of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

d) The security ASO control function activates the SSO function SSO-ProtectSign to append a signature to the user data provided in the SA-SEND Request, with the atnProtectSignST parameter containing the following:
- HMAC as the algorithm identifier,
- The Calling and Called Peer IDs,
- the User Data provided in the SA-SEND Request.

e) The security ASO control function activates the SESE SE-TRANSFER request, using the following parameters:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

f) Upon reception of the request to send the SESE PDU, the Dialogue control function

encodes the SESE PDU using the definition of the presentation-user-data with the presentation-context-identifier value corresponding to “sa-apdu” and invokes the P-DATA Request primitive with the resulting encoding as User Data.

3.1.10.2 Remaining data transfer: receiver side

a) The Presentation Protocol Machine (PPM) invokes the P-DATA Indication primitive,

b) The Security ASO decodes the presentation user data and extracts the presentation data value,

c) The Security ASO provides the SESE PDU to the Security ASO, together with the Calling and Called Peer Ids,

d) The SEPM activates the SE-TRANSFER indication, using the following parameters, as set by the peer SEPM:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

e) The Security ASO control function triggers the retrieval of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

f) The Security ASO control function activates the SSO function SSO-SignCheck to check the validity of the signature, using the following parameters:
- the Calling Peer ID,
- the Called Peer ID,
- the User Data and the signature,
- the algorithm identifier contained in the atnProtectSign security exchange item.

The SSO-SignCheck function is in charge of the following:
- verify that the counter corresponds to a valid number,
- check the validity of the signature contained in the atnProtectSign security exchange item, using the following parameters:
  - the algorithm identifier included in the received signature,
  - the public key of the Calling Peer Id,
  - the Calling Peer ID,
  - the Called Peer ID,
  - the User Data provided in the SA-START request.
  - the signature contained in the atnProtectSign security exchange item.

g) the Security ASO control function issues a SA-SEND Indication with the following parameters:
- the identification of the Calling Peer ID,
- the identification of the Called Peer ID,
- the Dialogue user data.

h) Upon reception of the SA-SEND Indication, the Dialogue control function issues a D-DATA Indication to the DS-User.

3.2 Other Application dialogues

3.2.1 Requirement:

The session key $MacKey_{U,W_i}$ is computed by the aircraft entity application as follows.

1. The aircraft entity application retrieves from the aircraft entity CMA the aircraft entity's private key agreement key $d_{s,U}$, the ground application entity's public key Q_{s,W_i} which the aircraft entity CMA obtained from the ground CMA entity during CMA login, and the shared public value $X_{U,V}$ agreed by the aircraft entity CMA and the ground CMA entity during CMA login.

2. The aircraft entity application calculates the application session key $MacKey_{U,W_i}$. To do this it first computes the shared secret value Z_{U,W_i} from the *x*-coordinate of the point $d_{s,U}Q_{s,W_i}$ using the aircraft entity's private key agreement key $d_{s,U}$ as specified in ANSI X9.63 [2]. It then computes the 80-bit $MacKey_{U,W_i}$ using the SHA-1 based ANSI X9.63 key derivation function from the shared secret value Z_{U,W_i}, the single octet 01_{16}, the shared public value $X_{U,V}$, an indication *APP* of the application the session key is for, the aircraft entity identity *U*, and the ground application entity's identity W_i as:

$$MacKey_{U,W_i} = KDF\ (Z_{U,W_i};\ 80;\ 01_{16} \parallel X_{U,V} \parallel APP \parallel U \parallel W_i)$$

Including $X_{U,V}$ in the key derivation process ensures the session key $MacKey_{U,W_i}$ is unique to this CMA session. Including *APP*, *U*, and W_i ensures that the session key is specific to an *APP* session between *U* and W_i.

This calculation employs the static unified model key agreement scheme specified in Section 6.3 of ANSI X9.63 [2].

[[Note that inclusion of the ground application entity's identity W_i in the key derivation procedure ensures that the session key is unique to a particular ground application location within the CM domain. This is desirable because it removes the need to synchronize a counter across ground application locations to prevent replays.]]

When the ground application entity and the aircraft entity application wish to exchange a data item $Data_4$, such as a D-Start or D-Data item, the sender authenticates the data item by MACing it using the application session key $MacKey_{U,W_i}$ as follows:

$$MAC\ (MacKey_{U,W_i};\ Source \parallel Count \parallel Data_4\ )$$

Source contains the sender's identity. Inclusion of *Source* enables detection of attacks in which an adversary bounces a data item sent by the aircraft or ground back to its sender. *Count* is a 16-bit counter indicating the number of data items sent over the RF channel by its sender secured under $MacKey_{U,W_i}$ during this CMA session. Inclusion of *Count* enables detection of attacks in which an adversary replays a previously sent data item.

The data item $Data_4$ is then sent over the RF channel along with the MAC value calculated on it. This ensures the authenticity of $Data_4$.

[[Note that as before *Count* may either be sent along with secured messages, or may be inferred from previous communications. Inference of *Count* is preferred in order to save bandwidth.]]

This application communication procedure is based on the HMAC scheme which is specified in the international standard IETF RFC 2104 [6], and the US standard ANSI X9.71 [3].

3.2.2 Solution proposal:

3.2.2.1 Initiator side: Request phase on aircraft

a) The Dialogue user issues a D-START request at the dialogue interface, with the Security

Requirements parameter set to the abstract value “Secured exchanges” and the User Data containing the user data provided by the user of the dialogue service.

- The identification of the Calling Peer ID,
- The identification of the Called Peer ID,
- The D-START request user data

b) The Security ASO control function triggers the retrieval of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

c) The security ASO control function activates the SSO function SSO-ProtectSign to append a signature to the user data provided in the SA-SEND Request, with the atnProtectSignST parameter containing the following:
- HMAC as the algorithm identifier,
- the Calling and Called Peer IDs,
- the User Data provided in the SA-SEND Request.

d) The security ASO control function activates the SESE SE-TRANSFER request, using the following parameters:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

e) Upon reception of the request to send the SETR PDU, the Dialogue control function:
   a) Retrieves the AE-qualifier as defined for the ATN-App AE,
   b) Constructs the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and zero otherwise,
   c) Looks up the calling presentation address,
   d) Looks up the called presentation address from the Called Peer Id parameter,
   e) Retrieves the Calling AP Title and Calling AE-Qualifier from the Calling Peer Id parameter.
   f) Retrieves the Called AP Title and Called AE-Qualifier from the Called Peer Id parameter.
   g) Sets the ACSE Requirements parameter to the symbolic value “Authentication functional unit ”,
   h) Maps the Security Requirements value to the A-ASSOCIATE Authentication mechanism name parameter.
   i) Maps the SETR PDU to the A-ASSOCIATE Authentication value parameter,
   j) Construct an A-ASSOCIATE Request primitive with the following parameters:

A-ASSOCIATE Request parameter	**ISO Status**	**ATN Status**
Mode	U	Not used (default value)
Application Context Name	M	As derived in b) above
Application Context Name List	C	Not used
Calling AP Title	U	As derived in e) above
Calling AE Qualifier	U	As derived in e) above
Calling AP Invocation Identifier	U	Not used
Calling AE Invocation Identifier	U	Not used
Called AP Title	U	As derived in f) above
Called AE Qualifier	U	As derived in f) above
Called AP Invocation Identifier	U	Not used
Called AE Invocation Identifier	U	Not used

ACSE Requirements	U	As derived in g) above
Authentication mechanism name	U	As derived in h) above
Authentication value	U	As derived in i) above
User Information	U	D-START User Data Parameter
Calling Presentation Address	M	As derived in c) above
Called Presentation Address	M	As derived in d) above
Presentation Context Definition List	U	Not used
Default Presentation Context Name	U	Not used
Quality of Service	M	See SV4
Presentation Requirements	U	Not Used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronisation Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not Used
Session-connection identifier	U	Not used

k) Invoke the A-ASSOCIATE Request primitive.

3.2.2.2 Receiver side: Indication phase on aircraft

a) The ACSE Protocol Machine (ACPM) invokes the A-ASSOCIATE Indication primitive with:
- the Calling and Called AP Titles set,
- the ACSE Requirements parameter set to the symbolic value “Authentication functional unit ”,
- the Authentication mechanism name parameter set to the abstract value “Secured Exchange”,
- the Authentication value parameter set to a SETR PDU,

b) The Dialogue control function:
- extracts the Calling Peer Id from the Calling AP Title,
- extracts the Called Peer Id from the Called AP Title,
- extracts the Dialogue user data, from the A-ASSOCIATE Indication user information field,
- extract the SESE PDU from the A-ASSOCIATE Indication Authentication value field,
- provides the SESE PDU to the Security ASO, together with the Calling and Called Peer Id and the Dialogue user data,

c) Upon reception of the SESE PDU, the Security ASO control function provides it to the SESE Protocol Machine (SEPM),

d) The SEPM activates the SE-TRANSFER indication, using the following parameters, as set by the peer SEPM:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

e) The Security ASO control function triggers the computation of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

The SSO-SessionKey function is in charge of the following:
- retrieve its private key agreement key
- retrieve the ground application entity public key,
- retrieve the shared public value computed during the CM login exchange,

- compute the shared secret value,
- compute the session key which will be used during the subsequent exchanges, using key derivation mechanisms with the following parameters:
  - The abstract value associated to SHA-1 as the algorithm identifier,
  - The shared secret value,
  - The shared public value,
  - The Calling Peer ID,
  - The Called Peer ID

f) The Security ASO control function activates the SSO function SSO-SignCheck to check the validity of the signature, using the following parameters:
- the Calling Peer ID,
- the Called Peer ID,
- the User Data and the signature,
- the algorithm identifier contained in the atnProtectSign security exchange item.

The SSO-SignCheck function is in charge of the following:
- verify that the counter corresponds to a valid number,
- check the validity of the signature contained in the atnProtectSign security exchange item, using the following parameters:
  - the algorithm identifier included in the received signature,
  - the session key,
  - the Calling Peer ID,
  - the Called Peer ID,
  - the User Data and the signature contained in the atnProtectSign security exchange item.

g) the Security ASO control function issues a SA-START indication with the following parameters:
- The identification of the Calling Peer ID,
- The identification of the Called Peer ID,

h) Upon reception of the SA-START indication, the Dialogue control function issues a D-START indication with:
- the Security requirements parameter set to the abstract value received in the Authentication mechanism name of the A-ASSOCIATE indication,
- the user data set to the user information of the A-ASSOCIATE indication,

3.2.3 Requirement:

The session key $MacKey_{U,W_i}$ is computed by the ground application entity as follows.

1. The ground application entity retrieves from the certificate directory the key agreement key certificate of aircraft entity *U*, the signature key certificate of the ground CMA entity *V*, and the current CRL. It checks neither of the certificates have been revoked using the CRL, and it verifies the certificates are valid using its copy of the CA's public key $Q_{sig,CA}$. It retrieves the aircraft entity's key agreement public key $Q_{s,U}$ and the ground CMA entity's signing public key $Q_{sig,V}$ from the certificates.

   These calculations employ the ECDSA verification transformation specified in Section 5.9.2 of ANSI X9.63 [2].

2. The ground application entity retrieves from the ground CMA entity the shared public value $X_{U,V}$ agreed by the aircraft entity CMA and the ground CMA entity during CMA login. The ground CMA entity should send the ground application entity $X_{U,V}$, a time field $Time_V$ indicating the current time, and its signature s_V on $Time_V \| X_{U,V}$:

   $$s_V = Sign\ (\ d_{sig,V};\ Time_V \| X_{U,V}\ )$$

   Signing the $X_{U,V}$ value is necessary to ensure the ground application entity that it really was generated by the ground CMA entity and the aircraft entity CMA (otherwise an active adversary on the ground network could substitute a fake value $X_{U,V}$' for $X_{U,V}$). Including a time field in the signed message is necessary to prevent an adversary replaying an old shared public value. When the ground application entity receives the $X_{U,V}$ value from the ground CMA entity, it should check the time field and the signature before accepting the validity of the $X_{U,V}$ value.

   These calculations employ the ECDSA signing transformation and the ECDSA verifying transformation specified in Section 5.9 of ANSI X9.63 [2].

3. The ground application entity calculates the application session key $MacKey_{U,W_i}$. To do this it first computes the shared secret value Z_{U,W_i} from the *x*-coordinate of the point $d_{s,W_i}Q_{s,U}$ using the ground application entity's private key agreement key d_{s,W_i} as specified in ANSI X9.63 [2]. It then computes the 80-bit $MacKey_{U,W_i}$ using the SHA-1 based ANSI X9.63 key derivation function from the shared secret value Z_{U,W_i}, the single octet 01_{16}, the shared public value $X_{U,V}$, an indication *APP* of the application the session key is for, the aircraft entity identity *U*, and the ground application entity's identity W_i as:

   $$MacKey_{U,W_i} = KDF\ (Z_{U,W_i};\ 80;\ 01_{16} \| X_{U,V} \| APP \| U \| W_i)$$

   The mathematical properties of elliptic curves ensure that the aircraft and the ground application compute the same session key $MacKey_{U,W_i}$.

   This calculation employs the static unified model key agreement scheme specified in Section 6.3 of ANSI X9.63 [2].

3.2.4 Solution proposal:

3.2.4.1 Initiator side: Request phase on ground

a) The Dialogue user issues a D-START request at the dialogue interface, with the Security Requirements parameter set to the abstract value "Secured exchanges" and the User Data containing the user data provided by the user of the dialogue service.

b) The Dialogue control function maps the D-START request on a SA-SEND request at the Security ASO interface, with the following parameters:

- The identification of the Calling Peer ID,
- The identification of the Called Peer ID,
- The D-START request user data

c) The Security ASO control function triggers the computation of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

The SSO-SessionKey function is in charge of the following:
- read, from the ATN Directory, the Certificate Revocation List,
- read, from the ATN Directory, the information associated to the Calling Peer ID, by selecting its signed key agreement certificate attribute, under the following form:
  - version of the certificate,
  - certificate serial number,
  - the identification of the algorithm used for signing the certificate,
  - the name of the certification authority which issued the certificate,
  - the period of validity of the certificate,
  - the identity of the Calling Peer Id,
  - the public key of the Calling Peer Id,
- check that the Certificate Revocation List does not contain, in its list of revoked certificates, the serial number of the certificate associated with the Calling Peer ID,
- check the validity of the Calling Peer ID signed key agreement certificate, using the following parameters:
  - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
  - The Certificate Authority public key, as signing key,
  - The public key of the Calling Peer Id as clear data,
  - The signature of the certificate.
- read, from the ATN Directory, the information associated to the Called Peer ID, by selecting its signature key certificate attribute, under the following form:
  - version of the certificate,
  - certificate serial number,
  - the identification of the algorithm used for signing the certificate,
  - the name of the certification authority which issued the certificate,
  - the period of validity of the certificate,
  - the identity of the Called Peer Id,
  - the public key of the Called Peer Id,
- check that the Certificate Revocation List does not contain, in its list of revoked certificates, the serial number of the certificate associated with the Called Peer ID,
- check the validity of the Called Peer ID signature key certificate, using the following parameters:
  - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
  - The Certificate Authority public key, as signing key,
  - The public key of the Called Peer Id as clear data,
  - The signature of the certificate.
- compute a time stamp,
- retrieve the shared public value computed during the CM login exchange,
- compute the shared secret value,
- compute the session key which will be used during the exchange, using key derivation mechanisms with the following parameters:
  - The abstract value associated to SHA-1 as the algorithm identifier,
  - The shared secret value,
  - The shared public value,
  - The Calling Peer ID,
  - The Called Peer ID

d) The security ASO control function activates the SSO function SSO-ProtectSign to append

a signature to the user data provided in the SA-SEND Request, with the atnProtectSignST parameter containing the following:
- HMAC as the algorithm identifier,
- the Calling and Called Peer IDs,
- the User Data provided in the SA-SEND Request.

e) The security ASO control function activates the SESE SE-TRANSFER request, using the following parameters:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

f) Upon reception of the request to send the SETR PDU, the Dialogue control function:
   a) Retrieves the AE-qualifier as defined for the ATN-App AE,
   b) Constructs the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and zero otherwise,
   c) Looks up the calling presentation address,
   d) Looks up the called presentation address from the Called Peer Id parameter,
   e) Retrieves the Calling AP Title and Calling AE-Qualifier from the Calling Peer Id parameter.
   f) Retrieves the Called AP Title and Called AE-Qualifier from the Called Peer Id parameter.
   g) Sets the ACSE Requirements parameter to the symbolic value “Authentication functional unit ”,
   h) Maps the Security Requirements value to the A-ASSOCIATE Authentication mechanism name parameter.
   i) Maps the SETR PDU to the A-ASSOCIATE Authentication value parameter,
   j) Construct an A-ASSOCIATE Request primitive with the following parameters:

A-ASSOCIATE Request parameter	**ISO Status**	**ATN Status**
Mode	U	Not used (default value)
Application Context Name	M	As derived in b) above
Application Context Name List	C	Not used
Calling AP Title	U	As derived in e) above
Calling AE Qualifier	U	As derived in e) above
Calling AP Invocation Identifier	U	Not used
Calling AE Invocation Identifier	U	Not used
Called AP Title	U	As derived in f) above
Called AE Qualifier	U	As derived in f) above
Called AP Invocation Identifier	U	Not used
Called AE Invocation Identifier	U	Not used
ACSE Requirements	U	As derived in g) above
Authentication mechanism name	U	As derived in h) above
Authentication value	U	As derived in i) above
User Information	U	D-START User Data Parameter
Calling Presentation Address	M	As derived in c) above
Called Presentation Address	M	As derived in d) above
Presentation Context Definition List	U	Not used
Default Presentation Context Name	U	Not used
Quality of Service	M	See SV4
Presentation Requirements	U	Not Used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronisation Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not Used

Session-connection identifier	U	Not used

k) Invoke the A-ASSOCIATE Request primitive.

3.2.4.2 Receiver side: Indication phase on ground

a) The ACSE Protocol Machine (ACPM) invokes the A-ASSOCIATE Indication primitive with:
- the Calling and Called AP Titles set,
- the ACSE Requirements parameter set to the symbolic value “Authentication functional unit ”,
- the Authentication mechanism name parameter set to the abstract value “Exchange supporting key management”,
- the Authentication value parameter set to a SETR PDU,

b) The Dialogue control function:
- extracts the Calling Peer Id from the Calling AP Title,
- extracts the Called Peer Id from the Called AP Title,
- extract the SESE PDU from the A-ASSOCIATE Indication Authentication value field,
- provides the SESE PDU to the Security ASO, together with the Calling and Called Peer Id and the Dialogue user data,

c) Upon reception of the SESE PDU, the Security ASO control function function provides it to the SESE Protocol Machine (SEPM),

d) The SEPM activates the SE-TRANSFER indication, using the following parameters, as set by the peer SEPM:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

l) The Security ASO control function triggers the computation of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

The SSO-SessionKey function is in charge of the following:
- read, from the ATN Directory, the Certificate Revocation List,
- read, from the ATN Directory, the information associated to the Called Peer ID, by selecting its signed key agreement certificate attribute, under the following form:
  - version of the certificate,
  - certificate serial number,
  - the identification of the algorithm used for signing the certificate,
  - the name of the certification authority which issued the certificate,
  - the period of validity of the certificate,
  - the identity of the Called Peer Id,
  - the public key of the Called Peer Id,
- check that the Certificate Revocation List does not contain, in its list of revoked certificates, the serial number of the certificate associated with the Called Peer ID,
- check the validity of the Called Peer ID signed key agreement certificate, using the following parameters:
  - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
  - The Certificate Authority public key, as signing key,
  - The public key of the Called Peer Id as clear data,

    - The signature of the certificate.
- read, from the ATN Directory, the information associated to the Calling Peer ID, by selecting its signature key certificate attribute, under the following form:
    - version of the certificate,
    - certificate serial number,
    - the identification of the algorithm used for signing the certificate,
    - the name of the certification authority which issued the certificate,
    - the period of validity of the certificate,
    - the identity of the Calling Peer Id,
    - the public key of the Calling Peer Id,
- check that the Certificate Revocation List does not contain, in its list of revoked certificates, the serial number of the certificate associated with the Calling Peer ID,
- check the validity of the Calling Peer ID signature key certificate, using the following parameters:
    - The algorithm used for signing, as specified in the certificate, as the algorithm identifier,
    - The Certificate Authority public key, as signing key,
    - The public key of the Calling Peer Id as clear data,
    - The signature of the certificate.
- compute a time stamp,
- retrieve the shared public value computed during the CM login exchange,
- compute the shared secret value,
- compute the session key which will be used during the exchange, using key derivation mechanisms with the following parameters:
    - The abstract value associated to SHA-1 as the algorithm identifier,
    - The shared secret value,
    - The shared public value,
    - The Calling Peer ID,
    - The Called Peer ID

e) the Security ASO control function issues a SA-SEND Indication with the following parameters:
- the identification of the Calling Peer ID,
- the identification of the Called Peer ID,
- the Dialogue user data.

f) Upon reception of the SA-SEND indication, the Dialogue control function issues a D-START indication with:
    a) the Security requirements parameter set to the abstract value received in the Authentication mechanism name of the A-ASSOCIATE indication,
    b) the user data set to the user information of the A-ASSOCIATE indication.

3.2.5 Requirement:

When the ground application entity and the aircraft entity application wish to exchange a data item $Data_4$, such as a D-Start or D-Data item, the sender authenticates the data item by MACing it using the application session key $MacKey_{U,W_i}$ as follows:

$$MAC\ (MacKey_{U,W_i};\ Source \parallel Count \parallel Data_4\ )$$

Source contains the sender's identity. Inclusion of *Source* enables detection of attacks in which an adversary bounces a data item sent by the aircraft or ground back to its sender. *Count* is a 16-bit counter indicating the number of data items sent over the RF channel by its sender secured under $MacKey_{U,W_i}$ during this CMA session. Inclusion of *Count* enables detection of attacks in which an adversary replays a previously sent data item.

The data item $Data_4$ is then sent over the RF channel along with the MAC value calculated on it. This ensures the authenticity of $Data_4$.

[[Note that as before *Count* may either be sent along with secured messages, or may be inferred from previous communications. Inference of *Count* is preferred in order to save bandwidth.]]

This application communication procedure is based on the HMAC scheme which is specified in the international standard IETF RFC 2104 [6], and the US standard ANSI X9.71 [3].

3.2.6 Solution proposal:

3.2.6.1 Remaining data transfer: emitter side

a) The Dialogue user issues a D-DATA Request at the dialogue interface, with the User Data containing the user data provided by the user of the dialogue service.

b) The Dialogue control function maps the D-DATA Request on a SA-SEND Request at the Security ASO interface, with the following parameters:
- the identification of the Calling and Called Peer IDs,
- The D-DATA Request user data,

c) The Security ASO control function triggers the retrieval of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

d) The security ASO control function activates the SSO function SSO-ProtectSign to append a signature to the user data provided in the SA-SEND Request, with the atnProtectSignST parameter containing the following:
- HMAC as the algorithm identifier,
- The Calling and Called Peer IDs,
- the User Data provided in the SA-SEND Request.

e) The security ASO control function activates the SESE SE-TRANSFER request, using the following parameters:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

f) Upon reception of the request to send the SESE PDU, the Dialogue control function

encodes the SESE PDU using the definition of the presentation-user-data with the presentation-context-identifier value corresponding to “sa-apdu” and invokes the P-DATA Request primitive with the resulting encoding as User Data.

3.2.6.2 Remaining data transfer: receiver side

a) The Presentation Protocol Machine (PPM) invokes the P-DATA Indication primitive,

b) The Security ASO decodes the presentation user data and extracts the presentation data value,

c) The Security ASO provides the SESE PDU to the Security ASO, together with the Calling and Called Peer Ids,

d) The SEPM activates the SE-TRANSFER indication, using the following parameters, as set by the peer SEPM:
- Security exchange identifier set to the identifier of the atnProtectSignSE security exchange.
- An unambiguous invocation identifier for the exchange.
- Security exchange item containing the signature computed using the SSO-Signature function, together with the clear user data,
- Item identifier set to the value of the atnProtectSign security exchange item.
- Start flag set to true in order to specify that this is the first request of the exchange.
- End flag set to true.

e) The Security ASO control function triggers the retrieval of the session key by activating the SSO-SessionKey function with the atnSessionKeyST parameter containing the following:
- the Calling Peer ID,
- the Called Peer ID,

f) The Security ASO control function activates the SSO function SSO-SignCheck to check the validity of the signature, using the following parameters:
- the Calling Peer ID,
- the Called Peer ID,
- the User Data and the signature,
- the algorithm identifier contained in the atnProtectSign security exchange item.

The SSO-SignCheck function is in charge of the following:
- verify that the counter corresponds to a valid number,
- check the validity of the signature contained in the atnProtectSign security exchange item, using the following parameters:
  - the algorithm identifier included in the received signature,
  - the public key of the Calling Peer Id,
  - the Calling Peer ID,
  - the Called Peer ID,
  - the User Data provided in the SA-START request.
  - the signature contained in the atnProtectSign security exchange item.

g) the Security ASO control function issues a SA-SEND Indication with the following parameters:
- the identification of the Calling Peer ID,
- the identification of the Called Peer ID,
- the Dialogue user data.

h) Upon reception of the SA-SEND Indication, the Dialogue control function issues a D-DATA Indication to the DS-User.