ATNP/WG3/WP 15-**31**

**AERONAUTICAL TELECOMMUNICATIONS NETWORK PANEL**

Working Group 3 (Applications and Upper Layers)

Honolulu (U.S.A), 19 – 22 January 1999

# Proposed Modifications to CNS/ATM-1 Applications To Support Package-2 Security Services

Presented by: F. Picard

SUMMARY

This WP provides a summary of the security mechanisms of the ATN Upper Layer Architecture proposed by WG3/SG3 to support security. All security functions are actually provided by a new component of the Dialogue Service Provider, the Security Application Service Object (Security ASO).

This WP proposes a way to modify the current specifications of the ASEs likely to have security requirements (i.e. CM, CPDLC and ADS). CPDLC is taken as an example.

The modification of CM to support the exchange of application-related security information is outside the scope of this WP.

Reference:

[1] Secured ATN Dialogue Service. WG3/SG3 – G. Mittaux-Biron.

# 1. Introduction

This WP provides a summary of the security mechanisms of the ATN Upper Layer Architecture proposed by WG3/SG3 to support the security. All security functions are actually provided by a new component of the Dialogue Service Provider, the Security Application Service Object (Security ASO) except the negotiation of the level of security needed on a dialogue which is left to the dialogue service users.

This WP proposes a way to modify the current specifications of the ASEs likely to have security requirements (i.e. CM, CPDLC and ADS). CPDLC is taken as an example.

The modification of CM to support the exchange of application-related security information is outside the scope of this WP.

This WP has been discussed by WG3/SG2 and WG3/SG3. It is proposed to be used as Guidance Material for the Security SARPs.

# 2. The ATN Security Framework

## 2.1 Overview

The operational requirements for ADS and CPDLC identify the need for security measures to be taken with respect to information flowing between end systems. Risk analysis and studies have shown these ATN applications are vulnerable to several threats, namely message modification, message replay, masquerade and denial of service attacks.

The ATN security architecture provides two main security services providing efficient countermeasures to these threats: peer entities strong authentication and data integrity checking.

The ATN security architecture is a public-key cryptosystem based on the use of a pair of keys (one private and one public) by each of the communicating AEs. Both keys can be used for encipherment, with the private key to be used to decipher if the public key was used, and the public key being used to decipher if the private key was used. By signing the message sent to the peer AE with its private key (known only by itself), the sender AE is identified by any remote AE owning the sender AE's public key. The signature – computed from the contents of the data sent and containing unique identifier and a timestamp – provides the guarantee that the message has not been replayed or modified during the transfer.

Public keys are made available as *certificates*. A certificate authenticated by a trusted party establishes the link between a Public Key and a particular user. The efficiency of the security system relies on the control of the way the keys and certificates are created, authenticated, distributed and updated. The key/certificate management is performed by the Public Key Infrastructure (PKI). The CM Application could be integrated in the PKI for the exchange of keys with the aircraft.

The scenario assumed for the exchange of keys and the establishment of secured dialogues between the air and the ground ASEs is the following:

1. The aircraft is supposed to have the public key of the initial ground CM. The air CM sends a CM-logon request signed with the aircraft private key.

2. The ground retrieves through the ATN directory service the public key of the aircraft and authenticates the aircraft as the originator of the message.

3. The ground-CM sends a CM-logon response to the aircraft signed with the ground CM private key and containing the public keys of the ground ATN applications (ADS, CPDLC and FIS).

4. Using the ground CM's public key, the air CM authenticates the received CM-logon response and stores the public keys of the ground ATN applications.

5. For air-initiated applications, e.g. CPDLC, the air CPDLC establishes a dialogue with a signature created with the aircraft private key. The ground already knowing the aircraft's public key (see 2.) can authenticate the sender. In return, the ground CPDLC sends a signature created with the ground CPDLC's private key. The aircraft uses the CPDLC's public key received in the Logon response to authenticate the ground CPDLC.

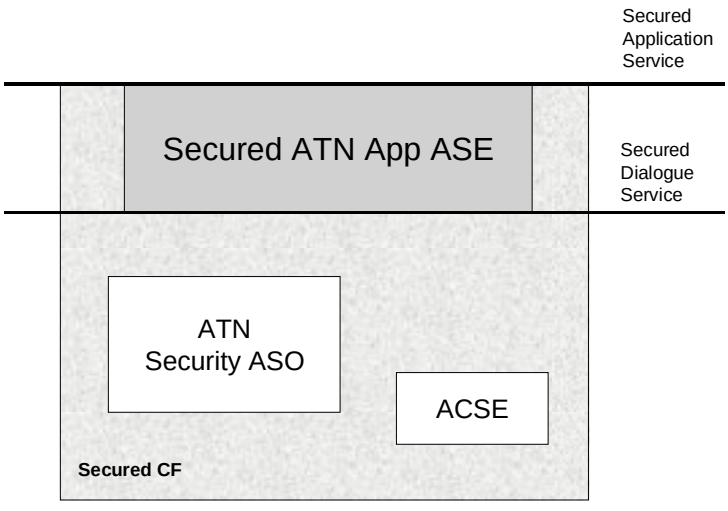
6. For ground-initiated applications, e.g. ADS, the ground ADS establishes a dialogue with a signature created with the ground ADS private key. The aircraft uses the ADS's public key received in the Logon response to authenticate the ground ADS. In return, the air ADS sends a signature created with the aircraft's private key. The ground already knowing the aircraft's public key (see 2.) can authenticate the sender.

The document deals with item 5 and 6, i.e. the way the ATN Application ASE establish a secured dialogue. The way the security information is passed by CM is outside the scope of this document.

## 2.2 Secured ATN Upper Layers

### 2.2.1 Model

The functional model of the ATN applications within the Secured ATN Upper Layer Architecture is showed in Figure 1.



### 2.2.2 Application Entity Security Functions

Security services are provided transparently by the Secure Dialogue Service Provider (SDSP) to the ATN Application Service Elements. The only requirements put on the ASEs when using the secured dialogue service are the following:

- **Supply the desired level of security** on a dialogue basis. 5 levels are defined as follows:
  - No security,
  - Peer Entities Authentication during dialogue establishment and
    - No data integrity check, or
    - Integrity check of data sent by the dialogue initiator only,
    - Integrity check of data sent by the dialogue responder only,
    - Integrity check of all data sent on the dialogue.
- **Negotiate the level of security**. An ATN ASE on behalf of its user must negotiate with the remote entity the level of security services which will be operated on the dialogue being established. In other words, the dialogue responder can downgrade the level of security proposed by the dialogue initiator and agreed by the dialogue service provider.

### 2.2.3 Security ASO

All the other functions are under the responsibility of the Security ASO.

During the dialogue lifetime, the Security ASO:

- takes part to the negotiation of the level of security on the current dialogue. In some circumstances, the Security ASOs involved in the communication can not provide the requested level of security and will therefore downgrade the level of security for the dialogue,
- generates message signatures based on the knowledge of the local private key and hash and signature algorithms,
- checks the received message signatures based on the knowledge of the peer public key and hash and signature algorithms,
- log errors locally when security failure is detected.

As side activities, the Security ASO is responsible for the following:

- Manage and keep secret the private keys in use,
- For each peer user to authenticate, retrieves the needed certificates and authenticates the peer user's public key.

These activities requires specific security services and protocols between the ATN end systems and the security systems as the Certificate Authority (CA).

# 3. The Secured ATN Upper Layers

## 3.1 Secured Dialogue Service

The secured dialogue D-START service is identical to the Package-1 dialogue D-START service with the exception of the *Security Requirements* parameter which should be defined from:

| | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Security Requirements | U | C(=) | U | C(=) |

to:

| | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Security Requirements | U | C(<=) | C(<=) | C(=) |

The DS-Users are expected to use this parameter to negotiate the level of security on the dialogue being established.

This modification of the D-START service does not impact the current specification of the ATN ASEs.

Note that the DS-user is not informed directly of a security check failure. In particular, the dialogue is never aborted by the DSP on detection of a security failure.

If the security check fails during the dialogue establishment on the receiver side, the D-START indication indicates that there is no security mechanism in operation for this dialogue (but the DS-user does not know that a security exception has occurred).

As a consequence, upon receipt of a D-START indication with no security requested, the called DS-user does know what happened:

- The calling DS-user requested no security at all, or
- The calling DS-user requested security but the DS-service provider does not support it (permanently or transiently), or
- The calling DS-user requested security but a security error (authentication or data integrity) has been detected during dialogue establishment.

Likewise, during the data transfer, if a security check fails, the DS-user is not informed. The error is logged locally and the System Management application is supposed to take the appropriate actions.

## 3.2 Secured Dialogue Protocol

During the establishment of a secured dialogue:

– The initiating Security ASO checks that it can comply with the requested level of security. The resulting level of security is passed to the peer in the A-ASSOCIATE request *authenticationMechanismName* parameter.

– The initiating Security ASO creates the following security PDUs (they are passed in the A-ASSOCIATE request *authenticationValue* parameter):

- An **Authentication Security PDU** containing the following fields (the recipient being the dialogue responder):
  - The security token in clear (algorithm identifier, name of the intended recipient, timestamp, token identifier),
  - The certification path needed to validate the public key of the initiator,
  - The security token signature (created with the algorithm identified in the token and the private key of the initiator),
- If initiator's data integrity check is requested, a **Security Signature PDU** containing the data signature (using the default algorithm or the negotiated one).
- If the initiator wants to negotiate the algorithms for future dialogues[1], a **Security Negotiation PDU** identifying the algorithms used for signing the security token and the algorithms used for signing the data,

- On reception of these PDUs, the receiver checks the identify of the initiator using the received security token signature and the validity of the data using the received data signature. Then it checks that it can comply with the requested level of security. The resulting level of security is passed to the local DS-user.

  However, for some reasons, the Security ASO can temporarily be enable to perform security ( e.g. because the Directory is not available, or the keys have been cracked, etc… ). In that case, the security level "no security" is proposed to the user.

  The level of security supplied by the the DS-user is send back in the A-ASSOCIATE response *authenticationMechanismName* parameter.

- The receiving Security ASO creates the following security PDUs (they are passed in the A-ASSOCIATE response *authenticationValue* parameter):
  - An **Authentication Security PDU** containing the same fields as above (the recipient being the dialogue initiator),
  - If responder's data integrity check is requested, a **Security Signature PDU** containing the data signature (using the default algorithm or the negotiated one).
  - If the responder wants to negotiate the algorithms for future dialogues, a **Security Negotiation PDU** identifying the algorithms used for signing the security token and the algorithms used for signing the data,

- On reception of these PDUs, the initiator check the identify of the receiver using the received security token signature and the validity of the data using the received data signature.

During the data transfer phase of a secured dialogue:

- The sender Security ASO creates the following security PDUs (they are passed in the P-DATA request *user data* parameter):

---

[1] *This facility will be used by CM is the default algorithms used for CM are not suitable for other applications. For instance, CM could use a powerful algorithm (producing traffic and time overhead) and the other applications could use a lighter algorithm.*

- If data integrity check is requested, a **Security Signature PDU** containing
  - the data in clear, and
  - the data signature (using the default algorithm or the negotiated one).

For the time being, there is no description in [1] of the actions taken by the Security ASO when a security attack (data modification, invalid signature) is detected.

# 4. Proposed changes to ATN Applications to support security

Security is therefore provided to the ATN applications exactly the same way Quality of Service is provided, i.e. **on a best effort scheme:**

- DS-users can negotiate the **maximum** level of security they can expect on a dialogue. This level of security results from the negotiation between the calling DS-user, the DS-service provider and the called DS-user.
- **The negotiated level of security is never guaranteed**. DS-users are never informed of security failures or the incapacity to support any longer the security functions. When authentication or data integrity checks are negative or when security checks become inoperative, the dialogue is maintained open and no indication is sent to the DS-users. Security failures are handled by the Systems Management application.

The proposal is to modify the ATN ASEs to support security in such a way the processing related to security be completely hidden to the ASE-users. The ASEs are modified to set and check the D-*START Security Requirements* parameter with no involvement of the ASE-users. Basically, the changes are the following:

- **For all applications**, whenever the D-START request primitive is invoked, the ASE sets the *Security Requirement* parameter to the value assigned for the application.
- **Only for the applications requiring strong security mechanisms[2]**, the *Security Requirement* parameter supplied by the DSP in the D-START indication and confirmation primitives is checked against the assigned value for the application. The dialogue is aborted if the expected level of security is not the one requested. For those applications, a new Exception Handling section is created to instruct the ASE to abort the dialogue with a new abort reason ("invalid-security-parameter") when the dialogue still exists.

Example of such modification are proposed in Annex of this document.

# 5. Conclusion

WG3 is invited to note the approach taken by SG2 to include security functionality to the CNS/ATM-1 applications.

---

[2] Strong security does not allow non-secured applications to operate with secured applications. It does not allow applications to operate in a non secured manner (when keys are obsolete for instance). It is likely that no Package-1 application will be defined with such so strong requirements. Therefore it is likely that the second set of changes proposed here will not be accepted.

# Example – CPDLC ASE

- **Chapter 2.3.3 (ASN.1 Description)**

*If strong authentication and data integrity checks are mandatory requirements for operating CPDLC, a new provider abort reason must be created.*

```
CPDLCProviderAbortReason ::= ENUMERATED
        {
        timer-expired                   (0),
        undefined-error                 (1),
        invalid-PDU                     (2),
        protocol-error                  (3),
        communication-service-error     (4),
        communication-service-failure   (5),
        invalid-QOS-parameter           (6),
        expected-PDU-missing            (7),
        ...,
        invalid-security-parameter      (8)
        }
```

- **Chapter 2.3.5 (Protocol Description)**

CPDLC-start service (CPDLC-Air-ASE)

2.3.5.3.8.1 Upon receipt of a CPDLC-start service response, if the CPDLC-air-ASE is in the START-IND state and the CPDLC-start service Result parameter has the abstract value “accepted” and the CPDLC-start service Reject Reason parameter is not provided, and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

a) Invoke D-START response with the following:

1) the abstract value “accepted” as the D-START *Result* parameter value,

2) the *abstract value "XXXX"* as the D-START *Security Requirements* parameter value, and

b) Enter the DIALOGUE state.

2.3.5.3.8.2 Upon receipt of a CPDLC-start service response, if the CPDLC-air-ASE is in the START-IND state, and the CPDLC-start service Result parameter has the abstract value “rejected” and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

a) If the CPDLC-start service *Reject Reason* parameter is provided, create an AircraftPDUs APDU with an ATCDownlinkMessage APDU element based on the *Reject Reason* parameter,

b) Invoke D-START response with the following:

1) the *abstract value "XXXX"* as the D-START *Security Requirements* parameter value,

2) if created, the APDU as the D-START *User Data* parameter, and

3) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and

c) Enter the *IDLE* state.

DSC-start service (CPDLC-Air-ASE)

tbd.

CPDLC-start service (CPDLC-Ground-ASE)

tbd.

DSC-start service (CPDLC-Ground-ASE)

tbd.

*If strong authentication and data integrity checks are mandatory requirements for operating CPDLC, a new provider abort reason must be created.*

2.3.5.3.2.1 Upon receipt of a D-START indication, if the CPDLC-air-ASE is in the *IDLE* state and the D-START *User Data* parameter contains a GroundPDUs [UplinkMessage] APDU, and the D-START *QOS Priority* parameter has the abstract value "high priority flight safety message" and the D-START *QOS Residual Error Rate* parameter has the abstract value "low", and the D-START *QOS Routing Class* parameter has one of the abstract values specified in Table 2.3.6-1, and the D-START *Security Requirements* parameter has the abstract value "XXX", the CPDLC-air-ASE shall:

2.3.5.3.3.1 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the START-REQ state and the D-START Result parameter has the abstract value “accepted” and DSC has the abstract value “false” and D-START User Data parameter is not provided and the D-START *Security Requirements* parameter has the abstract value "XXX", the CPDLC-air-ASE shall:

2.3.5.3.3.2 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the START-REQ state and the D-START Result parameter has the abstract value “rejected (permanent)” and the D-START Reject Source parameter has the abstract value “DS user” and DSC has the abstract value “false” and if the D-START User Data parameter is provided and the D-START *Security Requirements* parameter has the abstract value "XXX", the User Data parameter contains a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

2.3.5.3.3.3 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the START-REQ state and the D-START Result parameter has the abstract value “accepted” and DSC has the abstract value “true” and D-START User Data parameter is not provided and the D-START *Security Requirements* parameter has the abstract value "XXX",, the CPDLC-air-ASE shall:

Exception Handling (CPDLC-Air-ASE)

2.3.5.4.8 D-START *Security Requirements* Not as Expected

2.3.5.4.8.1 If in a D-START indication or confirmation, the *Security Requirement* parameter does not have the abstract value of “XXXX”, the CPDLC-air-ASE shall:

a) Stop any timer,

b) If a dialogue still exists,

1) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [invalid-security-parameter] APDU message element,

2) invoke D-ABORT request with:

i) the abstract value “provider” as the D-ABORT *Originator* parameter value, and

ii) the APDU as D-ABORT *User Data* parameter value,

c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and

d) Enter the *IDLE* state.

Exception Handling (CPDLC-Ground-ASE)

tbd.