Testing NPNT

PATANJALI MAITHANI

Aim

This presentation aims to provide an effecient workflow for testing the implication of NPNT policy as determined by the DGCA.

Some of the features mentioned by the DGCA were not able to be inculcated in the proposed system architecture. Such features are mentioned at the end of the presentation.

Three subsystems

Whole system can be divided into three main subsystems as follows:

1) RFM : Registered Flight Module
2) MC : Management Client and GCS : Ground Control Station
3) MS : Management Server

In the upcoming slides, desired features in a single subsystem and desired communication among them would be discussed.

Management Client

Features

An “init” method should be implemented internally with following features

- read Drone ID during each power-up init() call (Drone ID fetched from RFM)

- flight module is registered or not (by checking if UUID mapping exists for a particular Drone ID)

- connection with MS established or not

- initialize key rotation (triggered from MS or MC UI)

- check for software upgrades (triggered from MS)

- connection with physical flight module established or not.

- communicate flight module public key to MS.

Registration process

The MC fetches the Drone ID from the RFM and checks whether it is mapped to any UUID.

If UUID is not available, that means the system is not registered. If flight module is not registered, it should auto initiate registration. (To avoid the registration of illegal flight modules)

1st step: MC will send the Drone ID to Management Server.

2nd step : MS generates new UUID, sends this back to the MC, initiates other work-flows with Digital sky for the RPAS registration.

NOTE: the Drone ID can be composed of serial numbers, internal identifiers, signatures, etc. Flight Module Providers must ensure that this Drone ID does not change during the life of that physical flight module.

Avoiding invalid flight modules being registered

Flight Module Providers can send the one-time activation codes to people/entities who procure the flight module.

This provides a mechanism for Out of Band Authentication

Case Study example: A consumer buys a RPAS from a manufacturer.

Manufacturer takes the consumer’s phone number, email ID, Aadhar, DL (credentials).

Later when consumer connects his RPAS to MC, a dialogue box will appear saying that no UUID mapping exists, then consumer will have to enter his credentials. These credentials (with Drone ID) will be sent to MS where they will be mapped to credentials in the database, if they both matched an OTP would be send to the consumer’s mobile.

Consumer will then type this OTP in the MC.

This OTP will be send to MS, where it will get verified. If verified, registration can initiate.

Required fields during Registration

1) Drone ID (MC will fetch from RFM , in Backend)
2) Flight module Public key (fetched from RFM, in Backend)
3) Timestamp
4) Aadhar Number (from UI)
5) Driving license (from UI)
6) Mobile Number (from UI)

NOTE: Flight Module Provider should check pre-existence of serial number or other physical unique attributes to ensure same flight module gets same flight module code UUID. In the case of new registration, server generates a new flight module code (UUID) and should send back to client.

Flight Module Providers Management Server should call Digital Sky Register API to ensure flight module is registered with Digital Sky.

After successful registration with Digital Sky, Flight Module Providers Management server should sign the flight module public key and return to client.

Some other Important features in MC

- Sending Plan request to DSP.
- Fetching Permission artefact from DSPs. (rectified Plan)
- Sending Permission artefact to RFM.
- Fetching Signed Bundled Flight logs from the RFM.
- Sending Signed Bundled Flight logs to DSPs.

Key rotation scenario

1) Initiated by Management Server or Management Client UI.
2) Triggered received by Management Client (for the case when initiated by Management Server)
3) Key pair rotation initiated inside RFM.
4) Key-pair created inside RFM.
5) Private key is kept securely inside the RFM.
6) Newly generated public key is signed with the past Private key and sent back to Management Client. (PKI secured)*
7) From Management Client to Management Server. (https secured)
8) Update Management Server registry and update the Digital Sky platform.
9) From MS to MC, signed using Flight Module Provider key (Token).

*NOTE: The Communication between MC and RFM should also be secured. The same can be achieved using PKI infrastructure between RFM and MC.

Management Server

1) Secured Database system.
2) Flight Module public Key should be signed by one of the Flight Module Providers Keys. The provider must sign the flight module public key on the flight module providers server with the smart card/PKI Token.
3) Flight Module provider must ensure each physical flight module has a unique id (drone ID). Maximum length of the code is 50 characters when represented as string. To ensure flight module codes are globally unique it is necessary that Flight Module Provider uses a 128-bit UUID (represented in HEX notation).

Fields in Management Server Database

1) Consumer‘s name
2) Driving License/ PAN Card / Passport number
3) Aadhar Card Number
4) E-mail ID
5) Mobile Number
6) Drone ID
7) UUID
8) Flight Module Public Key
9) Key rotation Logs :

Registered Flight Module (RFM) APIs

RFM core functionality

1) RPAS Identification (Drone ID) : Items to be uniquely identified to avoid alterations.
2) Verifying authenticity of the Permission Artefact :

   The RFM service must use the corresponding public keys to verify that the permission artefact is released by Digital Sky Platform and has not been tampered with during transport.

   If the public key has to be changed (When DigitalSky requests the Flight Module Providers to change), the Flight Module Provider has to release a firmware update to update the key.

   Flight Module Provider should build security measures to ensure that any 3rd party is not able to alter the public keys used to verify the permission artefact in the library.

3) Provide Information of Time and Location bound restrictions to Flight Controller

4) Collect Flight Logs

5) Send Flight Logs to DSP

RFM API Reference

1) Get_rfmInfo
2) Apply permission Artefact
3) Get_Geofence_restriction
4) Get_time_restriction
5) Log_takeoff_location
6) Log_Land_location
7) Log_geofence_breach
8) Log_timelimit_breach
9) Get_individual_flight_logs
10) Bundle_flight_logs

Conducting Test

1) Check for ARM/DISARM in following conditions

(i) Inside permitted Area

ii) Outside permitted Area

iii) In Time limit

iv) Outside Time limit

2) Check for generated flight logs (during flight)

(i) Inside permitted Area

(ii) Outside permitted Area (RTH)

(iii) In time limit

(iv) Outside time limit (RTH)

Testing for Possible Scenarios

1) Registration Scenario
2) Key rotation scenario (of Registered Flight Module)

   -triggered from MS / MC

3) Digital Sky Public Key change Scenario.
4) Flight Log management, generation, signing, transportation

Challenges

1) Secure Firmware upgrade and bootloader upgrade
2) RPAS identifier