Implementation Details of Protocol 2 (Serverless authentication of Remote ID)

Serverless authentication method for Drone Remote ID (RID) that operates by broadcasting two distinct authentication message packs: one carrying the drone's compressed certificate and another conveying the RID telemetry data along with its corresponding digital signature. This method assumes no live server dependency—AKA all verification happens offline using cryptographic validation.

During flight, the drone periodically transmits its identity and telemetry data in compliance with ASTM F3411 broadcast specifications. The authentication process is split into two independent authentication message packs:

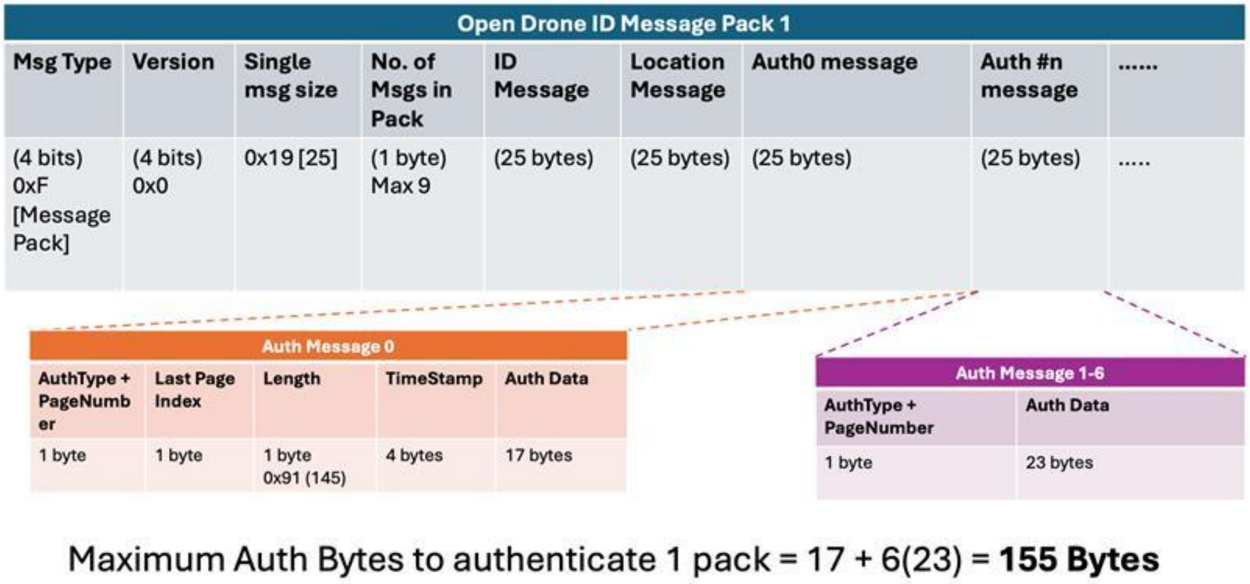
- **Message Pack 1** (AuthType = MessageSetSignature) encapsulates the real-time location data and the drone's unique identifier. Crucially, it also includes a digital signature computed over this data using the drone's private key. This co-location of data and signature ensures that each message is synchronized with its integrity check.
- **Message Pack 2** (AuthType = Reserced, for experimental deployment) carries the drone's compressed certificate. This message is broadcast less frequently and serves to convey the public key and associated metadata required to validate Message Pack 1.

Field	**Description**	**Size**
String subject	A short, encoded identifier of the drone.	8 bytes
String issuer	Identifies the Certificate Authority (CA) that issued the certificate.	8 bytes
Long begin	Unix timestamp representing the certificate's start of validity.	8 bytes
Long end	Unix timestamp representing the certificate's expiration.	8 bytes
BigInteger serialNumber	A unique serial number for this certificate	6 bytes
byte[] publicKey	The drone's public key (Ed25519)	32 bytes
byte[] signature	A digital signature by the CA over the certificate's content using Curve Ed25519	72 bytes
Total	Combined size of all fields.	142 bytes

The certificate is compressed because the maximum number of authentication data bytes that can be transmitted within a message pack 2 is limited to 201 bytes.

Message pack details:

Maximum auth data bytes for message pack 1:



Open Drone ID Message Pack 1

Msg Type	Version	Single msg size	No. of Msgs in Pack	ID Message	Location Message	Auth0 message	Auth #n message	......
(4 bits) 0xF [Message Pack]	(4 bits) 0x0	0x19 [25]	(1 byte) Max 9	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)	.....

Auth Message 0

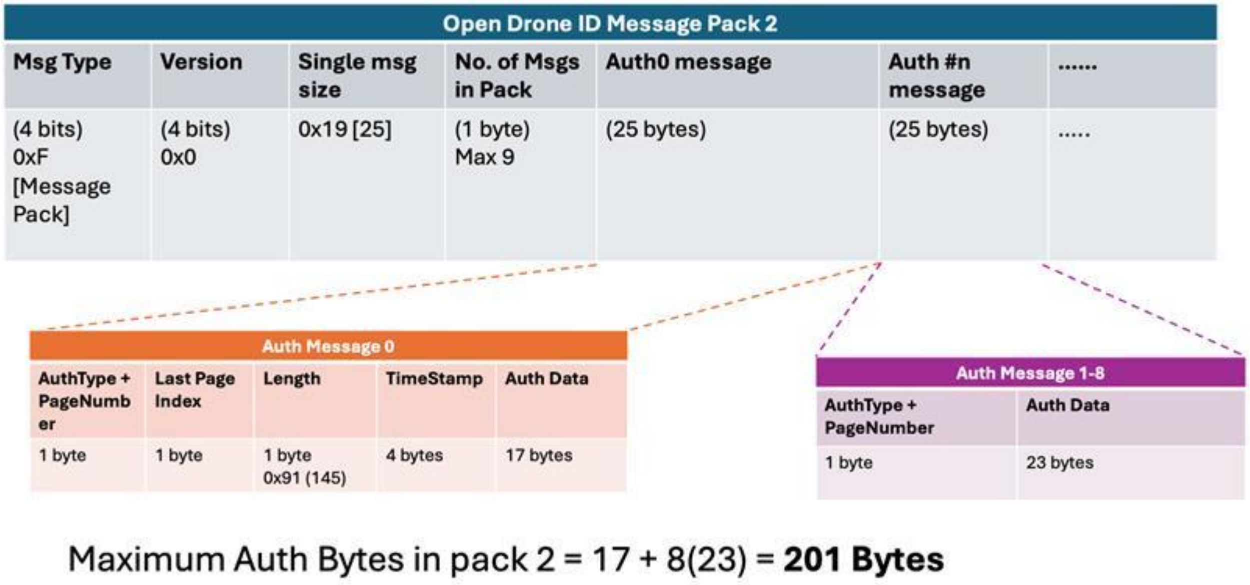
AuthType + PageNumber	Last Page Index	Length	TimeStamp	Auth Data
1 byte	1 byte	1 byte 0x91 (145)	4 bytes	17 bytes

Auth Message 1-6

AuthType + PageNumber	Auth Data
1 byte	23 bytes

Maximum Auth Bytes to authenticate 1 pack = 17 + 6(23) = **155 Bytes**

Maximum auth data bytes for message pack 2:



Open Drone ID Message Pack 2

Msg Type	Version	Single msg size	No. of Msgs in Pack	Auth0 message	Auth #n message	......
(4 bits) 0xF [Message Pack]	(4 bits) 0x0	0x19 [25]	(1 byte) Max 9	(25 bytes)	(25 bytes)	.....

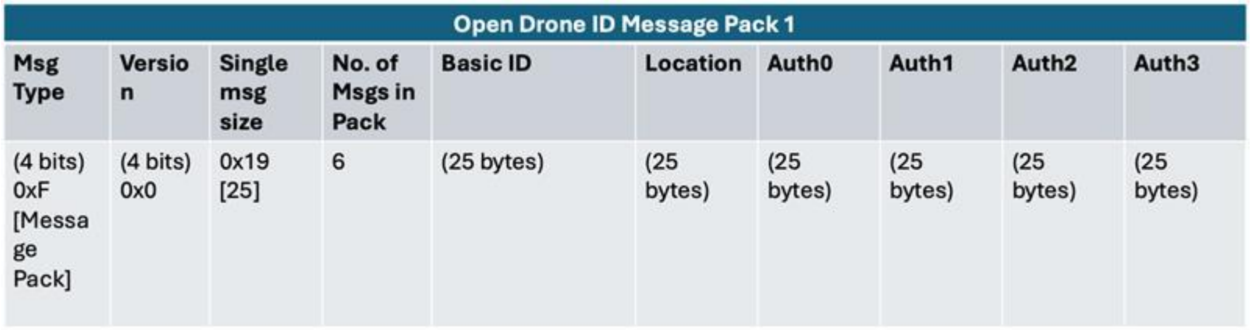
Auth Message 0

AuthType + PageNumber	Last Page Index	Length	TimeStamp	Auth Data
1 byte	1 byte	1 byte 0x91 (145)	4 bytes	17 bytes

Auth Message 1-8

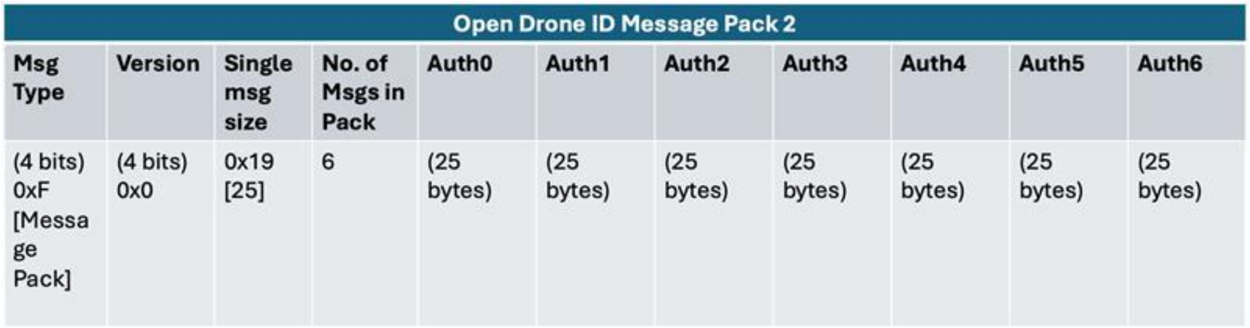
AuthType + PageNumber	Auth Data
1 byte	23 bytes

Maximum Auth Bytes in pack 2 = 17 + 8(23) = **201 Bytes**



Open Drone ID Message Pack 1

Msg Type	Version	Single msg size	No. of Msgs in Pack	Basic ID	Location	Auth0	Auth1	Auth2	Auth3
(4 bits) 0xF [Message Pack]	(4 bits) 0x0	0x19 [25]	6	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)



Open Drone ID Message Pack 2

Msg Type	Version	Single msg size	No. of Msgs in Pack	Auth0	Auth1	Auth2	Auth3	Auth4	Auth5	Auth6
(4 bits) 0xF [Message Pack]	(4 bits) 0x0	0x19 [25]	6	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)	(25 bytes)

Observers, upon receiving Message Pack 2, extract and decompress the certificate. The authenticity of this certificate is verified using the CA's public key, which is assumed to be pre-installed in the observer's system. Upon successful validation, the certificate and its hash are cached for subsequent use. When Message Pack 1 is received, the observer retrieves the cached certificate using the reference hash, extracts the drone's public key, and verifies the signature attached to the telemetry data. If the signature is valid and the certificate remains within its validity period, the message is deemed authentic and trustworthy.

However, if the observer receives Message Pack 1 before Message Pack 2, the signature contained in the RID data cannot be immediately verified. In such cases, the message is flagged as untrusted. The authenticity of this message remains in doubt until the corresponding certificate is received and validated via Message Pack 2

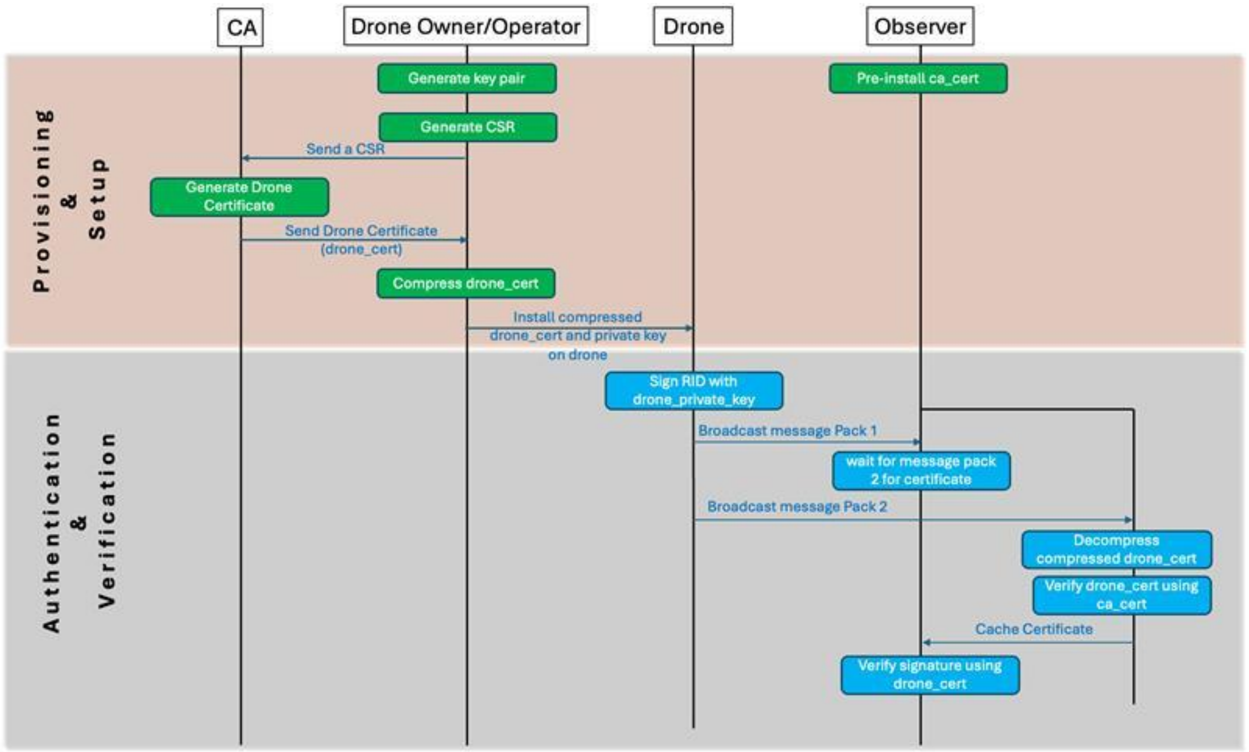

CA
Drone Owner/Operator
Drone
Observer
Provisioning & Setup
Generate key pair
Pre-install ca_cert
Generate CSR
Send a CSR
Generate Drone Certificate
Send Drone Certificate (drone_cert)
Compress drone_cert
Install compressed drone_cert and private key on drone
Authentication & Verification
Sign RID with drone_private_key
Broadcast message Pack 1
wait for message pack 2 for certificate
Broadcast message Pack 2
Decompress compressed drone_cert
Verify drone_cert using ca_cert
Cache Certificate
Verify signature using drone_cert

Fig.: Workflow of Protocol in the implementation perspective: