**3GPP TSG-SA3 Meeting #108-AdHoc *S3-222749-r1***

**e-meeting, 10th October – 14th October 2022**

**Source: Intel**

**Title: Updates to solution 2: remove EN key management**

**Document for: Approval**

**Agenda Item: 5.8**

# 1 Decision/action requested

***This pCR removes an EN in Solution 2 to TR 33.738. It is requested to approve the pCR for TR 33.738***

# 2 References

# 3 Rationale

# 4 Detailed proposal

SA3 is kindly requested to agree on the pCR below to TR 33.738

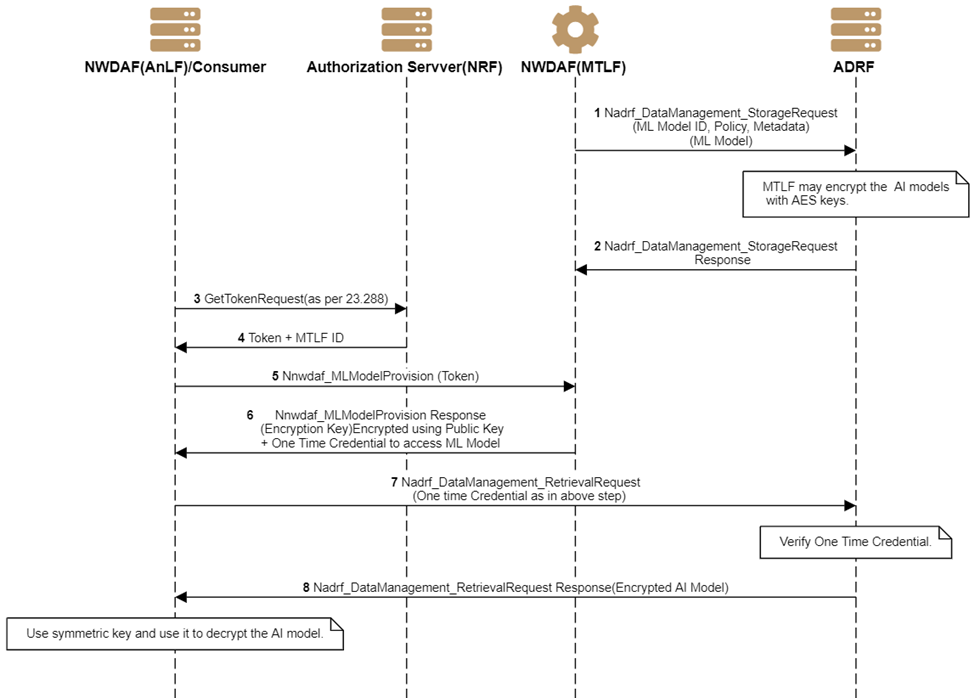
\*\*\*\*\*Start of Changes\*\*\*\*\*

6.2 Solution #2: Authorization and Authentication of ML model transfer

6.2.1 Introduction

The solution proposed below protects AI/ML models between the entity which produces the ML model or stores the ML model in ADRF and the entity which consumes the model (NFc). In this solution, an authorization token is used by ADRF to verify that the NFc is allowed to access the ML model.

6.2.2 Solution Details



**Figure 6.Y.2.1-1 Secure ML model transfer**

1. The MTLF trains the ML model and sends ML Model to the ADRF by invoking the Nadrf\_DataManagement\_StorageRequest (ML Model) service operation. Along with Model. Metadata of the model is also sent for each model, e.g., ML model ID, analytics ID, Vendor ID, MAC or SHA256 Signature of the Binary of the application, environment required for ML model execution, URL/link to retrieve configuration, and secrets, and/or a signing key, certificate to generate authentication credentials. The NWDAF containing MTLF generates a security context for protecting the ML model information using a logical function or named network function NKGC. MTLF may send an ML model encrypted using a symmetric key (e.g., AES key) before the storage. The security context consists of an encryption key Kenc, an integrity key Kint, and the corresponding security algorithm(s) for encryption and integrity protection. The NWDAF containing MTLF uses the encryption key Kenc and integrity key Kint. to protect the ML model and related information. The NKGC stores the security context.

2. ADRF stores the ML model and response as per TS 23.288[5], except that the ADRF stores the ML model.

Editor’s Note: The procedure to store the ML model in the ADRF needs to be updated per the SA2 conclusion.

Editor’s Note: Clarification on necessity on end-to-end protection of ML model is FFS

Editor’s Note: Key refresh and revocation is FFS.

3. Consumer, e.g., NWDAF/ANLF, contacts the NRF and requests an access token using existing procedures in 33.501[2]

4. NRF sends an access token along with MTLF ID using existing procedures per TS 23.288[5].

Editor’s Note: How NRF authorizes the request of NWDAF (AnLF) is ffs.

5. The consumer uses Nnwdaf\_MLModelProvision service operation for ANLF receives ML model ID based on analytics ID and ADRF id to retrieve ML model.

6. MTLF verifies the access token received in step 3. MTLF may send the encryption key used in step 1 to encrypt the ML model, which is stored in ADRF. MTLF also sends one-time credentials to access the ML model from ADRF. One-time credentials may include

a. Nonce, which is shared in step 1 as part of the metadata OR

b. MAC or Hash of a binary or random number shared in step 1 as part of the data OR

c. A signing key as a private key of the. The public part is passed in step 1 OR

d. MTLF uses it's signing key to generate the credentials, e.g., a JWT token or a certificate.

NOTE: One-time credentials can be used to limit the number of accesses from the consumer NF. The one-time credential may be used as a regular authorization token for accessing the ML model multiple times, i.e., not only once, as the name suggests otherwise.

7. Consumer of the ML model, e.g., ANLF, uses the ADRF service procedure to request the ML model. It also sends a one-time credential received in step 6.

8. ADRF verifies the one-time credentials (as specified in step 6). If the access token verification is successful, the ADRF provides the stored model to the consumer NF.

6.2.3 Evaluation

TBD

\*\*\*\*\*End of Changes\*\*\*\*\*