**3GPP TSG-SA3 Meeting #109*AdHoc-e draft\_S3-230237-r1***

**16 - 20 January 2023**

**Source: Intel**

**Title: EN related to Key Refresh and revocation**

**Document for: Approval**

**Agenda Item: 5.8**

# 1 Decision/action requested

***This contribution proposes delete EN in solution 2 in TR 33.738.***

# 2 References

[1] 3GPP TR 33.738 V0.4.0 "Study on security aspects of enablers for Network Automation for 5G - phase 3"

[2] 3GPP TR 23.700-81 V2.0.0 " Study of Enablers for Network Automation for 5G System (5GS); Phase 3"

# 3 Rationale

This pCR proposes remove the deletion of key refresh and revocation.

# 4 Detailed Proposal

It is proposed to approve the following changes for inclusion in TR 33.738.

\*\*\*BEGIN OF First CHANGE\*\*\*

# 7 Conclusions

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 6.2.2.1-1 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.

.

Note: Key handling such as key refresh and revocation is outside of the scope of this solution and not addressed in this solution.

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]. According to TS 29.510[8], An NWDAF can add Nnwdaf\_MLModelProvision\_Subscribe service operation in "allowedOperationsPerNfType" and/or "allowedOperationsPerNfInstance" for specific NF type and/or specific instance ID of the Consumer, and register its NF profile to NRF. When an NF Service Consumer requests an access token for the Nnwdaf\_MLModelProvision\_Subscribe to retrieve AI/ML models, the NRF grants the access token if Nnwdaf\_MLModelProvision\_Subscribe is present in either "allowedOperationsPerNfType", for the NF type of the NF Service Consumer, or in "allowedOperationsPerNfInstance", for the instance ID of the NF Service Consumer. MTLF knows the NF instance IDs of the AnLF as per existing procedures in 23.288[5] (e.g. through OAM).

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

As per key issue 3, ”the ADRF itself cannot be considered as a fully trusted entity storing the sensitive AI/ML data models ML models may be stored in the public cloud for storage”. The solution proposes to address this issue that ADRF can’t be a fully trusted entity,The solution presents the E2E encryption approach where the consumer and ML model owner can access ML models in clear text. 3rd party public cloud will not have access to the ML model, whose sole purpose is to store the model/files. E2E encryption prevents leakage of IP, i.e., cloning, copying, or internal ML models details by heuristics. in single vendor and multi-vendor environments; furthermore, when ML models are not encrypted, they are cloned ML models may lead to revenue loss or advantage to a competitor. E2E encryption provides security guarantees not provided by just transit security or access level controls to the database.

This solution assumes protection of ML model is vendor specific.

Editor’s Note: Further evaluation is FFS

.\*\*\* END OF First CHANGE \*\*\*