**3GPP TSG-SA3 Meeting #108e-AdHoc *draft S3-222854-r2***

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

**Source: Samsung**

**Title: New Solution on User Authorization in API Invocation**

**Document for: Approval**

**Agenda Item: 5.11**

# 1 Decision/action requested

***The contribution proposes a new solution for key issue for obtaining user authorization in API invocation.***

# 2 References

[1] 3GPP TR 23.700-95: “Study on application enablement aspects for subscriber-aware northbound API access”

[2] 3GPP TS 33.122: “Security aspects of Common API Framework (CAPIF) for 3GPP northbound APIs”

[3] 3GPP TS 33.535: “Authentication and Key Management for Applications (AKMA) based on 3GPP credentials in the 5G System (5GS)”

# 3 Rationale

This contribution proposes a new solution for a resource owner to provide/revoke the resource owner’s authorization to API Invoker.

# 4 Detailed proposal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.X Solution #X: Obtain User Authorization in API Invocation

### 6.X.1 Introduction

This solution addresses the requirement *“allow the UE to provide/revoke consent for information (e.g., location, presence) to be shared with the third-party”* in KI#Y.

This solution proposes to use a resourse owner’s 3GPP credentials for UE to generate a token, which is used to validate an API Invoker accessing the resource owner’s resources.

### 6.X.2 Solution details

### 6.X.2.1 Architecture



Figure 6.X.2.1-1 architecture for CAPIF with SNA enhancement

This solution uses an architecture proposed in solution #2 of TR 23.700-95 [xx]. As defined in TR 23.700-95 [xx], the Resource owner client, which is a client-side entity, is an application client used by end-user or subscriber of the API provider domain’s service provider. In addition, the resource owner communicates with the Authorization Function to revoke resource owner authorization via the Resource owner client, which interacts with the Authorization Function via CAPIF-8 interface.

### 6.X.2.2 Procedure

It is assumed that Authorization Function information (e.g., address), which is used by the resource owner to access the Authorization Function via resource owner client, is provisioned to the UE (e.g., during registration procedure).

It is assumed that API Invoker knows which APIs require a resource owner’s authorization.



Figure 6.X.2.2-1 Procedure for resource owner authorization based API invocation

1. If API Invoker does not have TokenAuz for service API invocation which requires the resource owner’s authorization, API Invoker needs to request resource owner’s authorization for the API invocation even if the API invocation is authorized from API exposing function as defined in TS 33.122 [yy].
2. API Invoker requests resource owner’s authorization for the API invocation. The request message includes Service API name (e.g. QoS API, location API, etc.) and API Invoker Information (e.g. API Invoker identity which is provided from CAPIF Core Function).
3. When the resource owner decides to give an authorization on the Service API name to the API Invoker (e.g. using GUI), UE generates a token (TokenAuz). The claims of TokenAuz include service API name, A-KID (AKMA Key Identifier), API Invoker Information, “Authorized”, generated time, and validity time. TokenAuz contains the claims (TokenAuz, claim) and the verification information (TokenAuz, verify). TokenAuz, verify is generated as detailed in 6.X.2.3 by using the claims and the key KAuz. Derivation of KAuz is detailed in 6.X.2.4 of this document.

Editor's Note: The need for online user interaction while the API is being invoked is FFS.

Editor's Note: When a UE is accessing the resources of another UE, how to get an authorization from the user of another UE is FFS.

Editor's Note: Consideration of subscription user and subscription owner is FFS.

1. If the resource owner gives the authorization for the API invocation, UE responds withTokenAuz. Upon receving the response, the API Invoker stores TokenAuz with UE ID (e.g. application layer ID or GPSI or SUPI).
2. API Invoker performs service API invocation withTokenAuz and UE ID (e.g. application layer ID or GPSI or SUPI).
3. API exposing function requests token verification to Authorization Function, via CAPIF-9 interface.
4. If Authorization Function does not have KAuz matched to A-KID which is included in TokenAuz claims, API exposing function requests KAuz from AAnF by sending a Naanf\_AKMA\_AuthorizationKey\_Get request as defined in 6.X.2.5 of this document. A-KID is defined in TS 33.535 [zz], and API exposing function discovers the AAnF using the A-KID as defined in TS 33.535 [zz]. If the API exposing function is outside the PLMN trust domain, Authorization Function shall request GPSI.
5. AAnF generates KAuz from KAKMA as specified in clause 6.X.2.4 in this document, and responds to Authorization Function with KAuz and UE ID (SUPI and optionally GPSI).
6. Authorization Function stores KAuz with UE ID (SUPI and optionally GPSI) and A-KID.
7. Authorization Function verifies the TokenAuz using KAuz.
8. Authorization Function responds with the verification result and UE ID (SUPI or GPSI).
9. If the verification result of the TokenAuz is successful, API exposing function stores the TokenAuz with UE ID (SUPI or GPSI). Untill API exposing function receives a revocation notification for the service API invocation, API exposing function uses the stored TokenAuz for authorizing the API Invoker without performing Token verification request to Authorization Function.
10. API Invoker receives the service API invocation response with UE ID (e.g. application layer ID or GPSI or SUPI) as a result of the service API invocation.
11. If the resource owner does not want for the API Invoker to invoke the service API, the resource owner can revoke the token anytime before the validity time of TokenAuz by using resource owner client. When the resource owner decides to revoke the token for the service API, UE generates a token (TokenRev). The claims of TokenRev include service API name, A-KID, API Invoker information, “Not authorized”, generated time. TokenRev contains the claims (TokenRev, claim) and the verification information (TokenRev, verify). TokenRev, verify is generated as detailed in 6.X.2.3 by using the claims and the key KAuz.
12. TokenRev is transmitted to Authorization Function via CAPIF-8 interface with revocation request message.
13. Authorization Function finds KAuz by using A-KID which is included in the TokenRev claims. Authorization Function verifies the TokenRev using KAuz.
14. Authorization Function notifies the revocation on the API Invoker’s service API invocation. The revocation notification includes TokenRev and UE ID.
15. For the same UE ID stored at step 11 and received at step 16, API exposing function finds TokenAuz which has same service API name and API Invoker information as TokenRev. If the generated time of TokenAuz is prior to that of TokenRev, API exposing function revokes TokenAuz and stores TokenRev. After this, if the API Invoker performs service API invocation using the revoked TokenAuz, API exposing function shall reject the API invocation request by noticing that the generation time in TokenAuz is prior to the generation time in TokenRev.
16. API exposing function notifies the revocation on the token with UE ID (e.g. application layer ID or GPSI or SUPI) to API Invoker.

### 6.X.2.3 Verification informationderivation

When deriving the verificiation information (TokenAuz, verify or TokenRev, verify) from KAuz, the following parameters shall be used to form the input S to the KDF.

* FC = 0xYY
* P0 = TokenAuz, claim or TokenRev, claim;
* L0 = length of TokenAuz, claim or TokenRev, claim;

The input key KEY shall be KAuz.

The verification information is identified with the 128 least significant bits of the output of the KDF.

NOTE: FC value to be determined during normative phase.

### 6.X.2.4 KAuz derivation function

When deriving a KAuz from KAKMA, the following parameters shall be used to form the input S to the KDF:

* FC = 0xZZ;
* P0 = “Authorization”;
* L0 = length of “Authorization”;

The input key KEY shall be the KAKMA.

NOTE: FC value to be determined during normative phase.

### 6.X.2.5 Naanf\_AKMA\_AuthorizationKey\_Get service operation

Service operation name: Naanf\_AKMA\_AuthorizationKey\_Get.

Description: The NF consumer (Authorization Function) requests Authorization Key and UE ID from the AAnF. If API exposing function is outside the PLMN trust domain, Authorization Function shall request GPSI.

Input, Required: A-KID.

Input, Optional: GPSI request indication.

Output, Required: KAuz, KAuz expiration time, and SUPI.

Output, Optional: GPSI.

### 6.X.3 Evaluation

TBD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*