**3GPP TSG-SA3 Meeting #114e *ad-hoc S3-24xxxx***

Electronic meeting, online, 22 - 26 January 2024

**Source: Federal Office for Information Security (BSI)**

**Title: Discussion of the protection mechanism of the permanent key leaving the UDR environment.**

**Document for: Discussion**

**Agenda Item:**

# 1 Decision/action requested

***This document is intended to serve as a basis for improving and protecting the permanent key leaving the UDR environment and related SCAS test cases.***

# 2 References

[1] 3GPP TS 33.501 Security architecture and procedures for 5G system

[2] 3GPP TS 33.117 Catalogue of general security assurance requirements

[3] OpenAPI Specification v3.1.0: <https://spec.openapis.org/oas/v3.1.0#optional-oauth2-security>.

[4] 3GPP TS 29.505 Usage of the Unified Data Repository services for Subscription Data

# 3 Rationale

The Unified Data Repository (UDR) provides an API that delivers subscriber authentication data to the Unified Data Management (UDM). This data includes the permanent key, encPermanentKey, which serves as the anchor of trust in the entire 5G system. This discussion paper analyzes the current state of protection mechanisms for sensitive data and suggests ways to improve them. Our rationale and proposal is splitted into the transfer and authorization protection mechnisms.

## Transfer Security

According to 5.8.1 in TS33.501 [1], it is the responsibility of the implementation to secure the transfer between the UDR and the UDM, as seen in the screenshot below. This can result in a situation where the transfer is completely unencrypted or proprietary protocols, which have not been security-analyzed, are used to transfer the user's permanent key. This key represents the ultimate trust anchor of each 5G system. Our proposal focuses solely on the transfer, not on the storage.

 **

## Authorization Security

TS 29.505 [4] defines the usage of the API, we will have a look specifications and how they are secured. While the focus in the previous section was on secure transport, this time we focused on authorization. The authorization shall be done by the means of OAuth2.0 according to TS33.501 [1]. This is important as no other network function that the UDM shall have access to the long-term key of the UE.



As discussed in NOTE 1: Only UDR be accessed with the means of OAuth2 authorization.



According to the API specification, the security has three different security definitions which are “or” connected. The first security definition ({}) indicates that security is optional see in the OpenAPI Specification [3]. The next two security definitions (oAuth2ClientCredentials) require the OAuth2 Client Credentials Grant type for authentication. One with the scope nudr-dr the other one with the scope nudr-dr:subscription-data:authentication-subscription:read.

With this current specification, there is the risk that any network function can access the permanent key with a nonsecurity mechanism, which contradicts the initial requirement of having a limited access to the permanent key stored in the UDR. We can use two routes to improve the security.

# 4 Detailed proposal

## Transfer Security

While 13.1.0 TS33.501 [1] points out that all network functions shall support mutually authenticated TLS and HTTPS and it is also recommended for usage “TLS shall be used for transport protection within a PLMN unless network security is provided by other means.” We see the uttermost importance of enforcing the use of TLS for the usage of the UDR. Therefore we suggest the following changes to 5.8.1 Generic requirements in TS33.501 [1] .

*NOTE 2: Security mechanisms for storage of subscription credentials in the UDR ~~and for the transfer of authentication subscription data (as specified in 3GPP TS 29.505 [70]) between UDR and ARPF~~ are left to implementation.*

Further, we add a note to point out the secure transport for this specific API endpoint.

*NOTE 3: The UDR should only support the access and transport of authentication subscription data only through the security mechanisms defined in section 13.1.0.*

## Authorization Security

To limited the access to the permanent key stored in the UDR, we can have two directions.

1. **Deleting nonsecurity mechsim**: We therefore suggest deleting the nonsecurity mechsim {}. From a security perspective, we also suggest further limiting access to only those who have access to the end-point nudr-dr:subscription-data:authentication-subscription:read. Also having the nudr-dr scope can have a wide audience gaining access to the security-sensitive key material. Therefore we would only see the following security definition for the endpoint /subscription-data/{ueId}/authentication-data/authentication-subscription: (get, patch, put). The final API specifcation could look like the following:



1. **Extra visibility document.** The second approach involves defining the visibility of data when using the non security parameter. The data returned depends on the authorization level provided to the API endpoint. For example, curl https://api.github.com/users/3gpp will only return public information. If you provide an OAuth2.0 token, you can also access private information. However, the current OpenAPI specification does not allow you to define which data is returned for each authorization level. Therefore, the suggestion is to map the response parameter to the authorization level **outside** of the OpenAPI specification, in a supplementary document, such as the TS29.505 [4].

## Implications for SCAS Test Cases for the UDR

In relation to the SCAS\_5G\_UDR work item, the decisions will have some impact on the SCAS-specific test cases. For example, if we add specific requirements to the UDR, these will need to be tested in the UDR-specific document. For example, an extra scope definition is proposed in the extra visibility document. If those are more of the general nature, those can be tested via the TS33.117 [2[. Nevertheless, it is important to discuss the requirements before deciding on the required SCAS actions.