**3GPP TSG SA WG3 Security — S3#22** **S3-020102**

**25 - 28 February 2002**

**Bristol, UK**

**Source:** **Alcatel**

**Title:** **Encryption of challenge in CHAP-based OSA authentication**

**Document for:** **Adoption**

**Agenda item:** **T.b.d.**

# 1 Introduction

This contribution discusses a specific functionality in TS 29.198-3 v4.2.0 which makes the challenge used for CHAP-based authentication to be encrypted when passed from the verifier to the claimant.

# 2 Issue

TS 29.198-3 relies on the use of a challenge-based mechanism (CHAP as per IETF RFC 1994) for authentication of the client application by the framework, and vice-versa. CHAP is chosen as the authentication scheme when the authentication type in the initiateAuthenticate() method is set to P_OSA_AUTHENTICATION.

The overall authentication phase works as follows:

- the client first uses the initiateAuthenticate() method to set the P_OSA_AUTHENTICATION scheme (ie CHAP).
- with the selectEncryption() method, the client application and the framework agree on a symmetric encryption function to be used to encrypt the challenge sent from the verifier to the claimant.
- the framework can then use the authenticate() method to pass an encrypted challenge string to the client, using the encryption algorithm (DES, triple DES) negotiated in the previous step. Encryption of the challenge string is done thanks to a secret key which must a priori be shared between the client and the framework (out of scope). The client must then decrypt the received encrypted challenge and generate a response based on the decrypted challenge and a secret shared with the framework. The client can authenticate the framework using the exactly the same mechanism in the other direction.

We hereby discuss various issues related to the above procedure.

## 2.1 Issue 1: no formatting defined for challenge encryption.

Symmetric encryption mechanisms such as DES, 3DES, … to be used for challenge encryption require the use of an Initialisation Vector (IV) as input into the encryption/decryption phases. This IV must be passed from the encryptor to the decryptor (or at least known by the decryptor).

The length of the challenge string is not necessarily a multiple of the encryption algo block length (eg 8 bytes for DES or 3DES). When it is not the case, padding bytes must be appended to the input (ie challenge string) of the encryption algorithm. After decryption, it is obviously necessary to be able to isolate those padding bytes so as not to use them as part of the challenge string. To avoid potential attacks, it is also important to provide the length of the challenge string within the encrypted data.

The description of the authenticate() method does not cover those aspects.

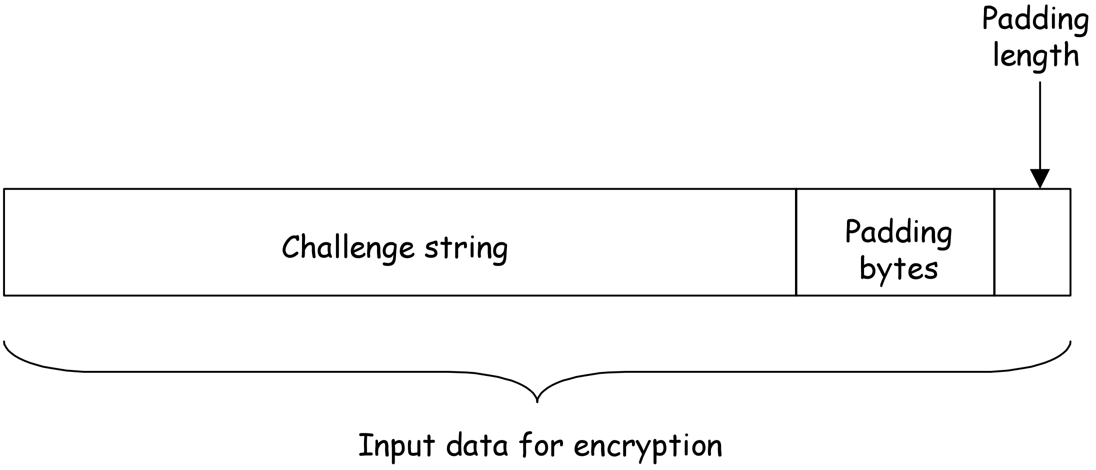
### 2.2 Issue 2: the need for encrypting the challenge

A more fundamental question is whether there is any real security gain in encrypting the challenge string itself. This indeed requires extra management (shared secret key for encryption/decryption between the client and the framework) and processing, while no identified security weakness is solved by this extra encryption process.

## 3 Solution

This contribution suggests to solve issue#1 by adding text to TS 29.198-3 that will clarify the format used to encrypt the challenge.

The challenge value is formatted as follows prior to encryption:



Padding bytes are appended to the challenge string value. A last byte is added that gives the number of padding bytes that were appended. The whole resulting octet stream must be a multiple of the block length of the symmetric encryption algorithm.

The Challenge parameter of the authenticate() method is therefore made as follows.

| | | | |
|---|---|---|---|
| Initialisation Vector | Challenge string | Padding bytes | Pad lgth |

Issue#2 does not necessarily require any solution. Although challenge encryption is not considered to add any security to the authentication phase, it does not bring any real harm either in terms of security.

## 4 Required Modifications to TS 29.198-3

In sections 6.3.1.1 and 6.3.1.5, for the Authenticate() method, the description of the Challenge parameter must be modified in order to clarify the formatting in place, as described in section 3 above.