**3GPP TSG-SA3 Meeting #116 draft\_S3-241864-r1
May 20 – 24, 2024 Jeju, KR**

**Source: Interdigital**

**Title: Solution for** **unauthenticated DOS protection**

**Document for: Approval**

**Agenda Item: 5.7**

# 1 Decision/action requested

***Approve the pCR to TR 33.713***

# 2 References

[1] TS 33.501 Security architecture and procedures for 5G System

[2] TR 33.700-29 Study on Security Aspects of 5G Satellite Access

# 3 Rationale

This contribution proposes a solution for remediation of unauthenticated DOS in S&F.

# 4 Detailed proposal

SA3 is asked to approve the following pCR.

\*\*\* Start of 1st Change \*\*\*

6.X Solution #X: Remediation of unauthenticated (D)DOS in S&F

### 6.X.1 Introduction

This solution addresses requirement #3

*The 3GPP system shall support means to mitigate the potential denial of service attack in the Store and Forward Satellite Operation.*

in Key Issue #1: Security protection in Store and Forward Satellite Operation, of the present document

This solution assumes that a satellite in S&F a Service Link (SL) is available, and the Feeder Link (FL) is temporarily unavailable.

When SL and FL are not available at the same time, the satellite in S&F mode has to store NAS or AS data on board of satellite. This condition can lead to the exhaustion of storage and processing resources on board the satellite in S&F mode.

When an unauthenticated UE attempts to register, a satellite in S&F mode will need to store the registration request until the FL becomes available. This condition can be exploited by fraudulent UEs mounting a (D)DOS attack. Backoff timer and throttling are effective means to remediate (D)DOS attacks post-authentication.

This contribution proposes a solution for the remediation of unauthenticated (D)DOS in S&F.

Registration Requests can be easily spoofed by attacker(s) generating a massive number of such requests. The proposed solution is based on the idea of slowing down the rate of Registration Requests from unauthenticated UEs and it is somewhat similar to the use of CAPTCHA in WEB access. The entity that is issuing the Registration Request (i.e., a legitimate UE or an attacker) is asked to perform a kind of action that is intended to slow down the rate of the Registration Request issuance.

Such action is more expensive to perform for the entities whose goal is to issue massive amounts of such requests (i.e., an attacker) than for a legitimate UE. As a result, this solution is disproportionally affecting the attacker.

### 6.X.2 Details

The simplified call flow associated with the proposed solution is presented below.



Figure 6.X.2-1: Call flow for Solution #X: Remediation of unauthenticated (D)DOS in S&F

Steps associated with the call flow in Figure 6.X.2-1:

1. UE issues Registration Request as per Registration Procedure as per TS 23.501 Figure 4.2.2.2.2-1: Registration procedure

2. The equipment on board of satellite, (either eNB/MME or gNB/AMF) senses that there is no available Feeder Link (FL) and decides to provide a puzzle-based (D)DOS remediation

3. The equipment on board of satellite, (either eNB/MME or gNB/AMF) composes a puzzle to be solved by the UE before propagating the Registration Request further in the operator’s network

4. The equipment on board of satellite, (either eNB/MME or gNB/AMF) forwards the puzzle to the UE

5. The UE solves the puzzle and produces the evidence

6. The UE re-issues the Registration Request with the evidence of solved puzzle

7. The equipment on board of satellite, (either eNB/MME or gNB/AMF) verifies the evidence and checks for the optional freshness.

Editor’s Note: It is FFS whether the additional workload that this solution (i.e., steps 2 – 4, and 7) proposes for the Network Nodes and NFs on board of satellite can be an acceptable tradeoff for remediation of (D)DOS attack.

Editor’s Note: It is FFS how to reduce latency due to the additional round trip.

8. The satellite detects the FL availability

9. The equipment on board of satellite, (either eNB/MME or gNB/AMF) forwards the Registration Request to the land-based CN authentication functionality (e.g., AUSF/UDM)

10. The S&F Registration Procedure continues with steps (4 or 5) as per TS 23.501 Figure 4.2.2.2.2-1: Registration procedure

### 6.X.2.1 Puzzles

A puzzle can be any cryptographic primitive (encryption, hash function, etc.) that would require brute brute-force attack to reverse.

#### Solving an Encryption Reversing puzzle example

Solving an encryption reversal puzzle is based on the “brute-force” method and it comprises finding plaintext or partial plaintext with either no encryption key knowledge, partial key knowledge, or reduced key size.

Note that processor productivity has an outsized effect on the time/effort needed to reverse encryption.

The procedure of solving this puzzle type is built around going through all existing permutations of the whole encryption key while knowing the partial encryption key. The process diagram is produced in Figure 3.



Figure 6.X.2.1-1: Solving an Encryption Reversing Puzzle

The steps in Figure 6.X.2.1-1 are reflected below:

1. Start of the procedure

2. The UnEn receives the puzzle and corresponding parameters from the ENN. The puzzle and the parameters correspond to what was generated in the procedure of Figure 2 step 10. This step corresponds to step 4 of Figure 1.

3. UnEn selects the initial (e.g., the starting value of the unknown part of the encryption key and uses it together with the known part of the key

4. UnEn executes the encryption function

5. UnEn checks if the encryption is brute-forced (e.g., if the brute-forced cleartext contains the optional known clear text corresponding to the input in Figure 2, step 3).

6. If No, the UnEn increments the unknown part of the key, uses that part together with the known part, and tries to brute-force the encryption again in step 4. If Yes, the process proceeds to step 7.

7. The evidence is comprised of the solved (i.e., brute-forced) encrypted text. The UnEn sends the evidence to the ENN. Sending the evidence corresponds to step 6 of Figure 1.

8. End of the procedure

#### Solving a one-way cryptographic hash function reversing puzzle example

Solving a one-way cryptographic hash function (e.g., SHA-256) reversing puzzle is based on the “brute-force” method and it comprises finding the complete hash function input text with only partial input hash function argument knowledge.

Increasing/decreasing the proportion between known and unknown portions of the hash function input can modulate the strength of the puzzle and the amount of work/effort that an entity (e.g., a UE) has to spend to solve it

RAM productivity has an outsized effect on the time needed to reverse the hash function.

The partially known argument to the cryptographic hash function will be the input parameter. For example, when using SHA-256 cryptographic hash, the input string to the hash has a total length of N and a known input length of N-m. The hash output is provided as one of the input parameters (stated length of 256 for SHA-256). It is the m-bits of the input to the hash function that are not known and comprise the puzzle. The effort will be needed to use a brute-force attack and discover the unknown m-bits of input, so that output = HASH-256 (known input || unknown input)

Evidence produced in the process of solving the puzzle comprises the full length of hash input.



Figure 6.X.2.1-2: Solving a Hash Function Reversing Puzzle

The steps in Figure 6.X.2.1-2 are reflected below:

1. Start of the procedure

2. The UnEn receives the puzzle and corresponding parameters from the ENN. The puzzle and the parameters correspond to what was generated in the procedure of Figure 2 step 10. This step corresponds to step 4 of Figure 1.

3. UnEn selects the initial (e.g., the starting value of the unknown part of the hash input and uses it together with the known part of the hash input

4. UnEn executes the hash function

5. UnEn checks if the hash is brute-forced (e.g., if the hash output corresponds to the whole hash input)

6. If No, the UnEn increments the unknown part of the hash input, uses that part together with the known part, and tries to brute-force the hash again in step 4. If Yes, the process proceeds to step 7.

7. The evidence is comprised of the solved (i.e., brute-forced) complete hash input text. The UnEn sends the evidence to the ENN. Sending the evidence corresponds to step 6 of Figure 1.

8. End of the procedure

### 6.X.3 Evaluation

This solution proposes a method for remediation of unauthenticated (D)DOS in S&F.

Further evaluation is FFS.

\*\*\* End of 1st Change \*\*\*