PROGRAMME OF THE EUROPEAN UNION

NAVIGATION MADE IN EUROPE

GALILEO OPEN SERVICE NAVIGATION MESSAGE AUTHENTICATION (OSNMA) RECEIVER GUIDELINES

Issue 1.3 | January 2024

#EUSpace

Terms of Use and Disclaimers

Authorised Use and Scope of Use

The European GNSS (Galileo) Open Service Navigation Message Authentication (OSNMA) Receiver Guidelines Issue 1.3 (hereinafter referred to as OSNMA Receiver Guidelines) and the information contained herein is made available to the public by the European Union (hereinafter referred to as Publishing Authority) for information, standardisation, research and development and commercial purposes for the benefit and the promotion of the European Global Navigation Satellite Systems programmes (European GNSS Programmes) and according to terms and conditions specified thereafter. The disclaimers contained in this document apply to the extent permitted by applicable law.

General Disclaimer of Liability

With respect to the OSNMA Receiver Guidelines and any information contained in the OSNMA Receiver Guidelines, neither the EU as the Publishing Authority nor the generator of such information make any warranty, express or implied, including the warranty of fitness for a particular purpose, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information hereby disclosed or for any product developed based on this information, or represents that the use of this information would not cause damages or would not infringe any intellectual property rights. No liability is hereby assumed for any direct, indirect, incidental, special or consequential damages, including but not limited to, damages for interruption of business, loss of profits, goodwill or other intangible losses, resulting from the use of the OSNMA Receiver Guidelines or of the information contained herein. Liability is excluded as well for consequences of the use and/or abuse of the OSNMA Receiver Guidelines or the information contained herein.

Copyright

The OSNMA Receiver Guidelines is protected by copyright. Any alteration or translation in any language of the OSNMA Receiver Guidelines as a whole or parts of it is prohibited unless the Publishing Authority provides a specific written prior permission. The OSNMA Receiver Guidelines may only be partly or wholly reproduced and/or transmitted to a third party in accordance with the herein described permitted use and under the following conditions:

- the present “Terms of Use and Disclaimers” are accepted, reproduced and transmitted entirely and unmodified together with the reproduced and/or transmitted information;
- the copyright notice “© European Union 2024” is not removed from any page.

Miscellaneous

No failure or delay in exercising any right in relation to the OSNMA Receiver Guidelines or the information contained therein shall operate as a waiver thereof, nor shall any single or partial exercise preclude any other or further exercise of such rights.

Updates

The OSNMA Receiver Guidelines could be subject to modification, update and variations.

The publication of updates will be subject to the same terms as stated herein unless otherwise evidenced.

Although the Publishing Authority will deploy its efforts to give notice to the public for further updates of OSNMA Receiver Guidelines, it does not assume any obligation to advise on further developments and updates of the OSNMA Receiver Guidelines, nor to take into account any inputs, comments proposed by interested persons or entities, involved in the updating process.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

ISBN: 978-92-9206-066-4
doi: 10.2878/256023

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Document Change Record

Reason for Change	Issue	Revision	Date
First Issue	1	0	December 2022
The document implements: The inclusion of a reference to the OSNMA IDD ICD (section 1.1, 1.5 and 2.4). The inclusion of an example of the OSNMA data verification processes in Annex A. The inclusion of two test vectors in Annex B.	1	1	November 2023
The new revision of the document provides new text vectors in Annex B.	1	2	January 2024
Correction of the cryptographic material provided in *Annex B – OSNMA Test Vectors cryptographic_material/Merkle_tree_3/PublicKey* folder. Text in section B.2 is aligned to the new content of the folder above.	1	3	January 2024

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Table of Contents

1 Introduction 6

1.1 Scope of the Document 6

1.2 What is OSNMA 6

1.3 Structure of the Document 7

1.4 Bit and Byte Ordering Criteria 7

1.5 Applicable Documents 8

2 Receiver Requirements 9

2.1 Time Synchronisation Requirement 9

2.2 Required Cryptographic Functions 9

2.3 Integrity of the Cryptographic Material and Functions 9

2.4 Interfaces 10

2.5 Memory Requirement 10

3 OSNMA Data Retrieval 11

3.1 Merkle Tree Root Retrieval 11

3.2 Public Key Retrieval 11

3.3 TESLA Root Key Retrieval 12

3.4 TESLA Chain Keys Retrieval 12

3.5 Navigation Data Retrieval 12

3.6 Tags Retrieval 12

4 OSNMA Workflow and Status Monitoring 13

4.1 Workflow 13

4.1.1 Initialisation 13

4.1.2 Processing of the Authentication Material 14

4.2 NMA Status 15

4.2.1 Monitoring of the NMA Status 15

4.2.2 “Test” and “Operational” Status 16

4.2.3 “Don’t Use” Status 17

5 Overview of Cryptographic Operations 19

5.1 Verification of the Public Key Retrieved from the SIS 19

5.2 Verification of the TESLA Root Key 21

5.3 Determination of the GST Sub-frame 22

5.3.1 GST Retrieval and Verification from the SIS 22

5.3.2 GST Sub-frame Propagation 22

5.4 Verification of the TESLA Chain Key 23

5.4.1 TESLA Chain Properties 23

5.4.2 Number of Recursive Operations 23

5.4.3 Verification Process 23

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

5.5 Verification of the Tag and Authentication of Navigation Data 25
5.5.1 Identification of the Applicable Tag 25
5.5.2 Tags Sequence Verification 25
5.5.3 Flexible Tags Sequence Verification 25
5.5.4 Identification of the Applicable TESLA Chain Key 26
5.5.5 Tag Verification 26
5.5.6 Navigation Data Authentication 27
References 28
List of acronyms 29
ANNEX A Examples of OSNMA Verifications 31
A.1 Binary Data Representation Conventions 31
A.2 OSNMA Configuration 31
A.3 NMA Header 31
A.4 DSM-KROOT 32
A.4.1 DSM-KROOT interpretation 32
A.4.2 DSM-KROOT verification 32
A.4.3 PDK verification 33
A.5 TESLA Chain Key 34
A.5.1 TESLA Chain Key Interpretation 34
A.5.2 TESLA Chain Key Verification 34
A.6 Tags 35
A.6.1 MACK Message Interpretation 35
A.6.2 Tag Sequence Verification 36
A.6.3 Associated Navigation Data 36
A.6.4 MACSEQ Verification 37
A.6.5 Tags Verification 37
A.6.5.1 Tag0 Verification 37
A.6.5.2 ADKD4 Verification 38
A.6.5.3 ADKD12 Verification 38
A.7 DSM-PKR 39
A.7.1 DSM-PKR Interpretation 39
A.7.2 DSM-PKR Verification 40
A.7.3 Verification of the P_{DP} 40
ANNEX B OSNMA Test Vectors 42
B.1 Time Synchronisation considerations 42
B.2 Format of the Test Vectors 42
B.3 Test Vectors for Different OSNMA Configurations 47
B.3.1 Configuration 1 47

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

B.3.2 Configuration 2 48
B.4 Test Vectors for the Renewal and Revocation Processes 48
B.4.1 Chain Renewal 48
B.4.1.1 Step 1 49
B.4.1.2 Step 2 49
B4.2 Chain Revocation 49
B.4.2.1 Step 1 49
B.4.2.2 Step 2 50
B.4.2.3 Step 3 50
B.4.3 Public Key Renewal 51
B.4.3.1 Step 1 51
B.4.3.2 Step 2 51
B.4.3.3 Step 3 52
B.4.4 Public Key Revocation 52
B.4.1.1 Step 1 52
B.4.1.2 Step 2 53
B.4.1.3 Step 3 53
B.4.5 Merkle Tree Renewal 54
B.4.5.1 Step 1 54
B.4.5.2 Step 2 54
B.4.5.3 Step 3 55
B.4.6 OSNMA Alert Message 55
B.4.6.1 Step 1 55
B.4.6.2 Step 2 56
ANNEX C Receiver Initial Conditions and Fulfilment of the Time Synchronisation Requirement 57
C.1 Known Time Synchronisation Uncertainty 57
C2. Unknown Time Synchronisation Uncertainty 58
ANNEX D Increasing Resilience of Receivers to Spoofing Attacks and the role of OSNMA 59
ANNEX E Changes between the OSNMA Receiver Guidelines and the OSNMA Receiver Guidelines for the Test Phase 60

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

List of Figures

Figure 1. OSNMA processing logic 7
Figure 2. Verification of the Public Key and TESLA root key 14
Figure 3. Verification of the TESLA chain key and tags 15
Figure 4. Verification of the Public Key through the use of Merkle tree 20
Figure 5. TESLA chain key 23
Figure 6. EOC Step 1 test vector 49
Figure 7. EOC Step 2 test vector 49
Figure 8. CREV Step 1 test vector 50
Figure 9. CREV Step 2 test vector 50
Figure 10. CREV Step 3 test vector 51
Figure 11. NPK Step 1 test vector 51
Figure 12. NPK Step 2 test vector 52
Figure 13. NPK Step 3 test vector 52
Figure 14. PKREV Step 1 test vector 53
Figure 15. PKREV Step 2 test vector 53
Figure 16. PKREV Step 3 test vector 54
Figure 17. NMT Step 1 test vector 54
Figure 18. NMT Step 2 test vector 55
Figure 19. NMT Step 3 test vector 55
Figure 20. OAM Step 1 test vector 56
Figure 21. OAM Step 2 test vector 56
Figure 22. ADKD processing as a function of the time synchronisation uncertainty 57

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

List of Tables

Table 1. Required cryptographic functions 9
Table 2. Operations associated with the NMA status "Operational/Test" 16
Table 3. Operations associated with the NMA status "Don't Use" 18
Table 4. Description of the Public Key verification process 19
Table 5. Description of the TESLA root key verification process 21
Table 6. Description of the TESLA chain key verification process 24
Table 7. Description of the MACSEQ verification process 26
Table 8. Description of the tag verification process 26
Table 9. Cryptographic material 47
Table 10. PK curves 47
Table 11. OSNMA Configuration 1 48
Table 12. OSNMA Configuration 2 48

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

1 Introduction

1.1 Scope of the Document

The scope of this document is to provide guidelines for the user segment implementation of the OSNMA functionality, as defined in the Galileo OSNMA SIS ICD [AD.2] and should be considered strictly as a complement to it. The current document, together with the Galileo OSNMA SIS ICD [AD.2] and the Galileo OSNMA IDD ICD [AD.4], provides the information needed to verify the authenticity of the Galileo navigation message. These guidelines are drafted in a generic way and are not tailored for any specific platform or application.

The guidelines provided in this document enable an implementation of the OSNMA protocol, providing navigation data authentication. For the interested reader, examples of techniques that can be additionally exploited to obtain a more robust PVT solution are provided in Annex D.

The document also provides test vectors and sample data supporting the verification of OSNMA functionality implementation.

The test vectors and sample data are available to download in a dedicated link at the GSC web portal.

1.2 What is OSNMA

The Galileo Open Service (OS) is providing a Navigation Message Authentication (NMA) capability, allowing the users to confirm that Galileo OS Navigation Data originated from the Galileo system and has not been modified.

The authentication concept is based on two main principles:

- The use of different keys from a single one-way chain shared by the Galileo satellites through a Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol [1][2][3].
- The possibility to authenticate satellites which do not transmit OSNMA with the data retrieved from satellites transmitting OSNMA, referred to as cross-authentication.

Both principles reduce the computation and communication overhead, and increase the service availability and robustness to data loss.

From a receiver perspective, the process of the OSNMA data can be described at a high level by the following steps, illustrated in Figure 1:

- The receiver retrieves the **navigation data** and the corresponding OSNMA data (**tag**, **TESLA chain key** and **TESLA root key**). The **tag** authenticates the **navigation data** and is received before its associated **TESLA chain key**.
- The **TESLA root key** is authenticated by means of its digital signature using a **Public Key** that shall be available at the receiver.
- The receiver authenticates the **TESLA chain key** with the **TESLA root key** or with a previously authenticated key from the TESLA chain.
- The receiver re-generates locally the **tag** with the verified **TESLA chain key** and the **data**, and checks whether it coincides with the received **tag**.

If the result of all these steps is successful the user shall consider the **navigation data** as authentic.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

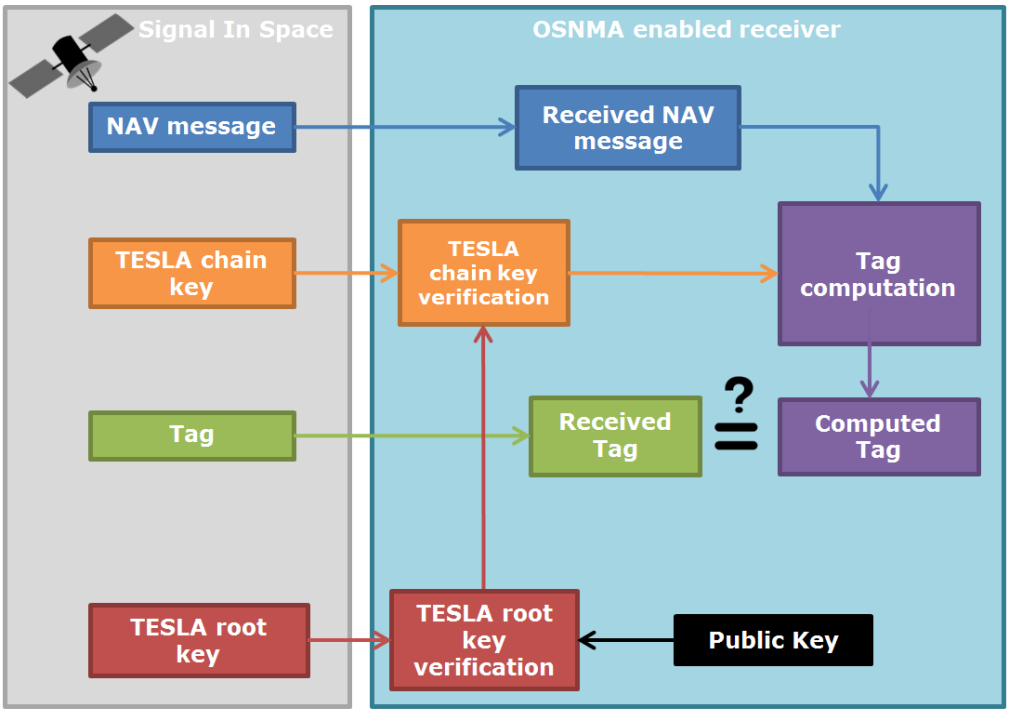


Figure 1. OSNMA processing logic

The retrieval of the data and operations required to perform these verification steps are further detailed in these guidelines. Specific strategies regarding the use of the information originated from the OSNMA verification steps, in the positioning process, are out of the scope of this document.

1.3 Structure of the Document

The document is structured as follows. After a short introduction in section 1, the requirements the receiver shall comply with to exploit OSNMA capabilities are presented in section 2. The data that need to be retrieved to exploit the scheme are presented in section 3. Section 4 shows how this data fits in the OSNMA verification workflow. The cryptographic operations that have to be applied to this data are presented in section 5.

Additional aspects are covered in the Annexes:

- Annex A provides examples of OSNMA step-by-step verifications,
- Annex B describes the set of OSNMA test vectors,
- Annex C discusses ways for the receiver to fulfil the time synchronisation requirement,
- Annex D presents additional steps that can be taken by the receiver to obtain a more robust PVT solution,
- Annex E presents the changes between the current document and the previous version of the document used for the OSNMA Test Phase.

1.4 Bit and Byte Ordering Criteria

All data values are encoded using the following bit and byte ordering criteria:

- For numbering, the most significant bit/byte is numbered as bit/byte 0.
- For bit/byte ordering, the most significant bit/byte is transmitted first.
- Except when noted, all fields are represented as unsigned integers as per Galileo OS SIS ICD [AD.1].

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

1.5 Applicable Documents

AD.1 European GNSS (Galileo) Open Service, Signal-In-Space Interface Control Document, Issue 2.1, 2023

AD.2 European GNSS (Galileo) Open Service, Galileo OSNMA SIS ICD, Issue 1.1, 2023

AD.3 European GNSS (Galileo) Open Service, Service Definition Document, Issue 1.2, 2021

AD.4 European GNSS (Galileo) Open Service, Galileo OSNMA Internet Data Distribution Interface Control Document (OSNMA IDD ICD), Issue 1.1, 2024

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

2 Receiver Requirements

This section presents the requirements the receiver shall fulfil in order to exploit Galileo OSNMA capabilities.

2.1 Time Synchronisation Requirement

To ensure the security of the TESLA protocol and guarantee the authenticity of the data, the receiver must ensure it has received the navigation data and associated tag before the corresponding TESLA chain key is disclosed by the system. This implies that the receiver must be synchronised with a given accuracy to the Galileo System Time (GST) before receiving and processing OSNMA information. The time synchronisation requirement T_L is set to 30 sec. If the receiver verifies this condition, all tags for all authentication types can be used.

If the condition is not verified, slow MACs, i.e. messages whose associated TESLA chain key is transmitted with an extra delay, may be exploited. A receiver synchronised to GST with an accuracy better than $T_L + 300\ sec$, can process slow MAC with a 10 sub-frame delay (ADKD12 from [AD.2]).

If none of the above conditions on the receiver time synchronisation to GST are verified, the OSNMA protocol shall not be used.

Additional considerations on the receiver time synchronisation uncertainty with respect to the GST can be found in Annex C.

2.2 Required Cryptographic Functions

The receiver shall be able to perform all the variants of the cryptographic functions listed in Table 1, according to their current standards. Future revisions of this document may consider additional algorithms. These functions are used for the verification of the OSNMA data, as further detailed in their indicated associated sections.

Table 1. Required cryptographic functions

Function	Ref.	Variants	Applicable section
Elliptic Curve Digital Signature Algorithm (ECDSA)	[4]	ECDSA P-256 ECDSA P-521	Verification of the TESLA root key (5.2)
Secure Hash Algorithm 2 (SHA-2)	[5]	SHA-256 SHA-512	Verification of the TESLA root key, TESLA chain key and Public Key (5.1, 5.2, 5.4)
Secure Hash Algorithm 3 (SHA-3)	[6]	SHA3-256	Verification of the Public Key and TESLA chain key (5.1, 5.4)
Hash-based Message Authentication Code (HMAC)	[7]	HMAC-SHA-256	Verification of the tag (5.5)
Cipher-based Message Authentication Code (CMAC)	[8], [9]	CMAC-AES	Verification of the tag (5.5)

2.3 Integrity of the Cryptographic Material and Functions

The receiver or the system containing the receiver is responsible for ensuring the integrity of the cryptographic material stored in its memory and of the processing of OSNMA data at a security level corresponding to its needs. This security level shall be defined in accordance with the protection profile of the receiver or of the system containing the receiver, taking into account the application it is supporting

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

and its environment. Integrity breaches shall be notified to the systems or applications which use the output of the receiver.

It shall also be considered that all cryptographic material may be subject to renewal or revocation.

2.4 Interfaces

To exploit the OSNMA capabilities, the receiver has to interface with the Galileo OS SIS, as described in [AD.1]. In addition, the receiver may interface with the OSNMA Server of the European GNSS Service Centre (GSC) to retrieve the cryptographic material, as described in [AD.4]. In case an OSNMA Alert Message (as defined in [AD.2]) is received, the receiver is required to connect to the GSC OSNMA server.

2.5 Memory Requirement

The different elements required by the receiver to exploit the OSNMA protocol can be retrieved from the SIS or from the OSNMA Server. In addition to the Merkle tree root (see section 3.1), the receiver may store some other elements of the protocol, in particular the Public Key and the TESLA root key (or an intermediate TESLA chain key previously verified). These elements shall be stored according to the requirement stated in section 2.3.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

3 OSNMA Data Retrieval

In order to exploit the OSNMA capabilities of Galileo Open Service, a receiver shall retrieve different information:

- A Merkle tree root;
- A Public Key;
- A TESLA root key;
- TESLA chain keys;
- Tags and associated navigation data.

As specified in the following section and detailed in [AD.2] and [AD.4], this information can be retrieved from the GSC OSNMA server or from the Signal In Space (SIS). The OSNMA data are transmitted within the Galileo SIS 40-bit OSNMA field, within the E1-B I/NAV navigation message. The OSNMA data is transmitted only by a subset of satellites; the remaining satellites have an OSNMA field filled with zeroes and their navigation data are cross-authenticated by the satellites transmitting OSNMA data. Therefore, the user shall discard any OSNMA field which is set to zero (i.e. 40 bits set to zero).

It shall be noted that in case of non-continuous data reception, the user can retrieve the different elements of the protocol provided in the SIS individually (e.g. tags, keys, parts of the DSM block), exploiting the way they are transmitted within the OSNMA fields, as defined in [AD.2]. For example, it is possible to retrieve the key by combining pages received from multiple satellites.

To be noted that, as specified in [AD.3], an authentication verification shall be performed only on navigation and OSNMA data for which a CRC checksum was successfully passed. The Galileo I/NAV CRC is described within [AD.1].

Additionally, several fields in the OSNMA data are reserved, as described in [AD.2]. The receiver shall be robust to any value being transmitted within the reserved fields.

3.1 Merkle Tree Root Retrieval

In order to validate new Public Keys retrieved from the SIS, the user must have the root of the Merkle tree and know which cryptographic function shall be used for the verification. This information can be loaded in the receiver after being retrieved from the GSC OSNMA server and verified using the PKI certificates, as described in [AD.4].

The receiver shall be able to download new Merkle roots from the OSNMA Server during the OSNMA lifetime, following an eventual Merkle tree renewal. Note that a renewal of the Merkle tree is expected to take place very rarely, typically after more than 10 years, as stated in [AD.2]. Note also that the future Merkle tree root will be available on the GSC OSNMA server at least two years before the planned renewal. In the event of a Merkle tree renewal, during this two-year period, a user may decide to store both the applicable and future Merkle tree roots in memory, and shall use the applicable one until the transition takes place. A user loading the Merkle tree root before the renewal will handle the transition as explained in section 4.2.2. Note that in the exceptional case that an OSNMA alert message is transmitted, the user will have to connect to the GSC OSNMA server for the recovery process, which includes the retrieval of the new Merkle tree root.

3.2 Public Key Retrieval

The receiver shall have a Public Key, with its associated ID and signature algorithm. This information can be retrieved from:

- The GSC OSNMA server and loaded in the receiver after verification using the PKI certificates, as described in [AD.4].

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

- The SIS, within the DSM-PKR transmitted at defined time intervals, as described in [AD.2]. The Public Key retrieved through the SIS shall then be authenticated, as per section 5.1. New Public Keys are broadcast during renewal and revocation processes, as described in [AD.2]. To be noted that Merkle tree renewal will induce a Public Key ID rollover. While retrieving the DSM-PKR, the sequencing of the DSM (described in [AD.2]) as well as their redundancy (i.e. same DSM ID block transmitted by several satellites) can be exploited. In addition, DSM blocks can be built using pages retrieved from different sub-frames. If the DSM-PKR is not completed after 13 hours, the retrieved DSM blocks shall be discarded.
- A previously stored Public Key, provided that its applicability is verified as per section 4.1.1.2.

3.3 TESLA Root Key Retrieval

The TESLA root key shall be retrieved from the DSM-KROOT transmitted by the SIS, as defined in [AD.2], and shall be verified as per section 5.2. Similar to the DSM-PKR, the defined transmission sequence of the DSM-KROOT can be exploited to optimise its reception. If a DSM-KROOT is not completed after 1 hour, the retrieved DSM blocks shall be discarded. Additionally, the MACK section can be retrieved in parallel to the DSM-KROOT, with the aim of verifying the retrieved TESLA chain keys and tags as soon as the TESLA root key has been verified. Note that also in that case, the conditions stated in section 5.4 and 5.5 shall be respected.

Instead of retrieving the TESLA root key from the SIS, the receiver can also use a previously stored TESLA root key, provided that its applicability is verified as per section 4.1.1.2.

Note that according to [AD.2], a DSM-PKR can be alternated with a DSM-KROOT, and two DSM-KROOT can also be transmitted alternatively, for example when a new chain is about to enter into force, as explained in section 4.2. Therefore, the receiver must be able to retrieve and store more than one DSM in parallel.

3.4 TESLA Chain Keys Retrieval

TESLA chain keys are provided in the MACK message, within the OSNMA message transmitted by the SIS, as defined in [AD.2]. The size of the TESLA chain key (l_K) is defined in the KS field of the DSM-KROOT.

3.5 Navigation Data Retrieval

The navigation data shall be retrieved on a sub-frame basis, as a function of the ADKD type as per the descriptions provided in [AD.2].

In addition, the PRN of the satellite transmitting the authentication information, $\mathrm{PRN_A}$, to be used for the verifications in sections 5.5.3 and 5.5.5, shall be the one of the PRN code used for tracking, and shall not be retrieved from the navigation message.

3.6 Tags Retrieval

The tags and their associated Tag-Info are also provided in the MACK message transmitted in the SIS, as described in [AD.2]. Several tags can be sent within a MACK message. The number of tags per MACK message n_t can be computed as a function of the TESLA chain parameter, as defined in [AD.2]. Each tag is a truncation of a MAC generated with a given TESLA chain key, as further detailed in section 5.5.5.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

4 OSNMA Workflow and Status Monitoring

4.1 Workflow

This section provides a description of the different operations that need to be performed in order to authenticate navigation data using the Galileo OSNMA protocol. The OSNMA workflow is divided in two steps:

- First, the initialisation, which consists in retrieving and verifying the Public Key and TESLA root key;
- Second, the processing of the authentication material to achieve navigation data authentication.

4.1.1 Initialisation

The OSNMA initialisation consists in the retrieval and verification of the Public Key and of the TESLA root key. As previously mentioned, this data can be either retrieved from the SIS and/or the GSC OSNMA server, referred as a factory start in the following section, or stored material can be used, referred as a warm start or hot start depending on the stored material.

4.1.1.1 Cold Start

In case the Public Key and TESLA root key in force are not available, the receiver shall first retrieve them, as illustrated in Figure 2. The Public Key can be retrieved directly from the GSC OSNMA server or from a DSM-PKR. In this latter case, the Public Key must be verified making use of the Merkle tree root, as further detailed in section 5.1.

The verified Public Key can then be used to verify the TESLA root key and associated chain parameters, sent in the DSM-KROOT, as per section 5.2. The verified TESLA root key can then be stored and used for the verification of the TESLA chain keys broadcast within the MACK message, as per section 5.4.
Note that if the receiver retrieved the Public Key from the GSC OSNMA server, it should verify that the key belongs to the Merkle tree the root of which is stored in memory. If the Merkle tree has been renewed, the new Merkle tree root shall be downloaded and the previous tree root and associated Public Keys shall be discarded.

If instead, the receiver retrieves the Public Key from the SIS and its verification fails, the Merkle tree root may have be renewed:

- If the receiver has the Merkle tree root in memory, it should verify the Public Key against it. If the verification is successful, the old Merkle tree root and associated Public Keys shall be discarded.
- If the receiver does not have an additional Merkle tree root in memory, the event shall be logged and reported at application level for a recovery action[1].

[1] Recovery action might require retrieval of new Merkle tree from the GSC OSNMA server.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

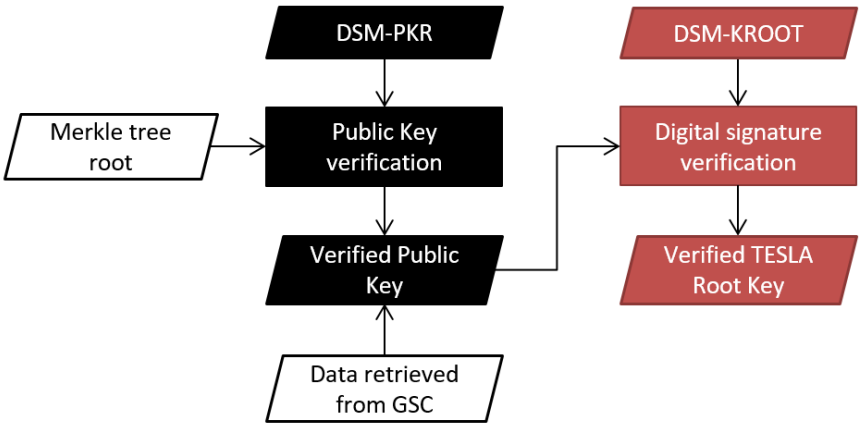


Figure 2. Verification of the Public Key and TESLA root key

4.1.1.2 Warm Start

If the receiver has stored a previously verified Public Key, and as long as it has been stored according to the requirements stated in section 2.3, the following steps shall be followed.

The receiver shall retrieve the DSM-KROOT and verify that the Public Key ID (PKID) contained in the DSM-KROOT corresponds to the one of the stored Public Key:

- If the PKID values are equal, the receiver can use the stored Public Key;
- If the values are different, the receiver shall discard the stored Public Key and retrieve a new one, as per section 4.1.1.1.
-

If the PKID of the DSM-KROOT matches the one of the stored key, the receiver shall verify the DSM-KROOT with the stored key:

- If the verification fails, the receiver shall retrieve a new Public Key, as per section 4.1.1.1.
- If the verification is successful, the receiver can use DSM-KROOT data (TESLA root key and chain parameters).

4.1.1.3 Hot Start

If, in addition of a stored Public Key, the receiver also possesses previously verified TESLA key and chain parameters, it does not have to retrieve the DSM-KROOT. The receiver shall attempt to verify the received TESLA chain key with the stored key, being this either the TESLA root or a previously verified TESLA chain key, as per section 5.4.

- If the verification is successful, the receiver shall use the stored material for the data authentication.
- If the verification fails, the receiver shall retrieve the DSM-KROOT and verify it as per section 4.1.1.2.

4.1.2 Processing of the Authentication Material

The processing of the authentication material is represented in **Figure 3**. To authenticate a navigation data set, the user identifies the required tag and retrieves it, as explained in section 5.5.1. The ADKD type of the retrieved tag is verified against the ADKD sequence defined in the MAC look-up table (MACLT), as described in section 5.5.2. If some of these ADKDs are defined as flexible (i.e. they are not fixed for the given chain), their Tag-Info sections can be verified through the MACSEQ verification, as detailed in section 5.5.3. The TESLA key applicable for the tag verification is retrieved as per section 5.5.4 and verified as per section 5.4, with the verified TESLA root key or with a TESLA chain key previously verified. The verified TESLA chain key is then used for the verification of the tag, as per section 5.5.5. It can also be used for the verification of successive TESLA chain keys, acting recursively

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

as new verified elements within the TESLA chain. The navigation data is authenticated as a result of successful tag verifications as per section 5.5.6.

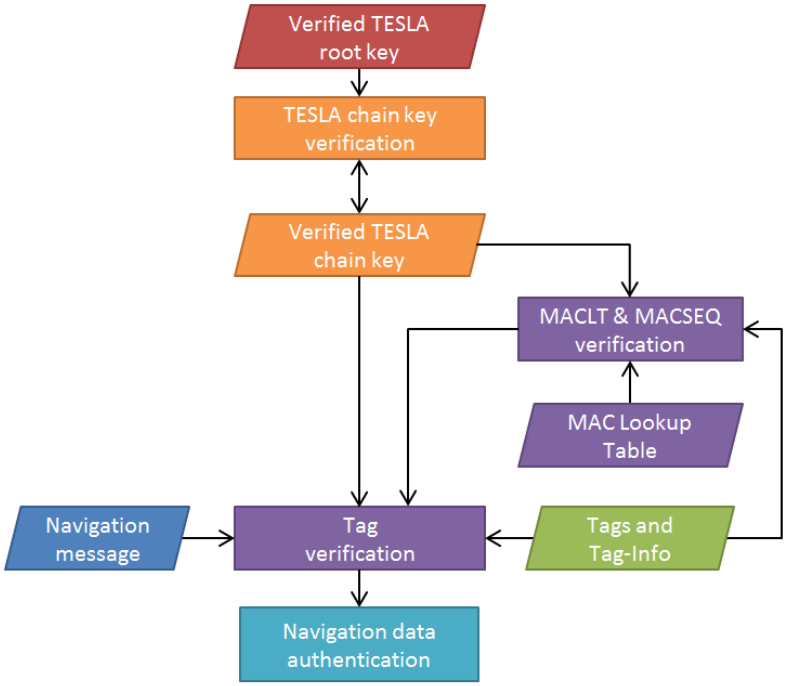


Figure 3. Verification of the TESLA chain key and tags

4.2 NMA Status

The result of the OSNMA authentication process is conditioned by the status of the NMA, which is reported in the NMAS field of the NMA Header. This value, which is the same for all satellites, shall be monitored by the receiver, and may require specific operations, as described below.

4.2.1 Monitoring of the NMA Status

The NMA status field can be set to three different values:

- "Test", to indicate that OSNMA is provided without any operational guarantees.
- "Operational", to indicate that OSNMA is provided according to the specifications.
- "Don't Use", to indicate that the receiver shall not perform navigation data authentication.

Before being interpreted, the NMA status shall be verified. The NMA status can be verified in two ways:

- In the tag verification, described in section 5.5. Thus, every time a tag is verified, the NMA status is verified as well.
- In the TESLA root key signature verification, described in section 5.2. As part of this verification, the Chain ID and CPKS flags, which may indicate that the cryptographic material is being renewed and revoked as defined in [AD.2], are also authenticated.

The user processing OSNMA data is required to verify the NMA status through the retrieval and verification of the TESLA root key signature at least once every hour. In the case the NMA status "Don't Use" is verified by means of the tag, the retrieval and verification of the TESLA root key signature, authenticating as well the Chain ID and CPKS flags, shall be immediately performed.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

These flags may indicate that the cryptographic material relative to the TESLA chain and to the Public Key are being revoked or renewed, as detailed in [AD.2].

4.2.2 "Test" and "Operational" Status

The CPKS field can be set to different values that shall be interpreted in combination with the NMA status. The possible combinations of CPKS and NMA status set to "Operational" or "Test", together with their interpretations and the operations the receiver shall perform, are presented in Table 2. This table shall be read together with the description of the renewal and revocation processes provided in [AD.2].

The cryptographic material defined in the table as 'current' corresponds to that applicable for the tag verification, as defined by the CID field for the TESLA chain in force (indicated in the NMA header) and by the PKID field associated to the Public Key in force (indicated in the DSM-KROOT), reported in the third and fourth columns. More detailed information of the renewal and revocation of the OSNMA cryptographic material can be found in [AD.2].

Table 2. Operations associated with the NMA status "Operational/Test"

NMA status	CPKS	PKID	CID	Required operations
Operational/ Test	Nominal	p	i	The navigation data shall be authenticated using the current TESLA chain material (CID = *i*). The TESLA chain material consists of the TESLA root key and the TESLA chain parameters.
Operational/ Test	End of Chain (EOC)	p	i	The current TESLA chain (CID = *i*) is coming to an end. The receiver shall retrieve the new TESLA chain material (CID = *i'*) sent in the DSM-KROOT and verify it with the Public Key (PKID=*p*) as per section 5.2. This material (CID = *i'*) will be used to authenticate the navigation data from its time of applicability, as defined in [AD.2]. Before that, while the CPKS flag is set to 'EOC', the navigation data shall still be authenticated using the current TESLA chain material (CID= *i*).
Operational/ Test	Chain Revoked (CREV)	p	i'	A previous TESLA chain (CID = *i*) has been revoked and shall be discarded. If the receiver does not possess the current TESLA chain material (CID = *i'*), it shall retrieve it in the current DSM-KROOT and verify it with the Public Key (PKID = *p*) as per section 5.2. The navigation data shall be authenticated using the current TESLA chain material (CID = *i'*).
Operational/ Test	New Public Key (NPK)	p	i	The Public Key (PKID = *p*) is being renewed. The receiver shall retrieve the new Public Key (PKID = *p'*), either from the DSM-PKR and verify it as per section 5.1, or from the GSC OSNMA server. When a DSM-KROOT is retrieved, it shall be verified with the current Public Key (PKID = *p*) and the navigation data shall be authenticated using the current TESLA chain material (CID = *i*).
		p'	i	The Public Key (PKID = *p*) has been renewed. The receiver shall retrieve the new Public Key (PKID = *p'*), either from the DSM-PKR or from the GSC OSNMA server. In the first case, the receiver shall verify DSM-PKR message as per section 5.1. The receiver shall discard the previous Public Key (PKID = *p*) and shall verify the retrieved DSM-KROOT with the current Public Key (PKID = *p'*) as per section 5.2. The receiver can then authenticate the navigation data using the current TESLA chain material (CID = *i*).

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

NMA status	CPKS	PKID	CID	Required operations
Operational/ Test	Public Key Revoked (PKREV)	p'	i'	The previous Public Key *p* has been revoked and shall be discarded. The TESLA chain material associated to this Public Key (CID = *i*) shall also be discarded. If the receiver does not possess the current Public Key (PKID = *p'*), it shall retrieve it in the current DSM-PKR and verify it as per section 5.1, or retrieve it from the GSC OSNMA server. If the receiver does not possess the current TESLA chain material (CID = *i'*) associated with (PKID = *p'*), it shall retrieve it in the current DSM-KROOT and verify it as per section 5.2. The navigation data shall be authenticated using the current TESLA chain material (CID = *i'*).
Operational/ Test	New Merkle Tree (NMT)	p	i	The Merkle tree is being renewed. The receiver shall retrieve the new Merkle tree root from the GSC OSNMA server[2]. When a DSM-KROOT is retrieved, it shall be verified with the current Public Key (PKID = *p*), while the navigation data shall be authenticated using the current TESLA chain material (CID = *i*).
		p'	i	The Merkle tree has been renewed. The receiver shall retrieve the new Public Key (PKID = *p'*), either from the DSM-PKR or from the GSC OSNMA server. In the first case, the receiver shall verify the DSM-PKR with the new Merkle tree root, as per section 5.1. The receiver shall discard the previous Merkle tree root and associated Public Keys. It shall verify the retrieved DSM-KROOT with the current Public Key (PKID = *p'*) as per section 5.2. The receiver can then authenticate the navigation data using the current TESLA chain material (CID = *i*).

Note that while the NMA status is set to “Operational” or “Test”, the continuity of the navigation data authentication process is maintained. In addition, once a key with a certain PKID is in force, any key with a lower PKID is considered not in use by the system (as per [AD.2]). Therefore, DSM with a PKID lower than that in force shall be rejected.

Note that if a DSM-KROOT is to be verified with a Public Key *p'* that the receiver does not possess, and such that PKID(*p'*) > PKID(*p*), *p* being the trusted Public Key stored in the receiver:

- If the CPKS flag is set to NPK or PKREV, the receiver should get the new DSM-PKR for *p'*, authenticate *p'* and replace the Public Key.
- If the CPKS flag is set to “Nominal”, the receiver may have been switched off for a long time and thus may have missed the PK update. The receiver may either get *p'* from the GSC OSNMA server, or get a DSM-PKR transmitted by the SIS and verify it.

4.2.3 “Don’t Use” Status

When the NMA Status is verified to be set to “Don’t use”, the receiver shall not perform navigation data authentication. The retrieval of new OSNMA material can be carried out, as described in Table 3. This table shall be read together with the description of the renewal and revocation processes provided in [AD.2].

[2] The new Merkle tree root may be loaded in the receiver before the CPKS flag is set to ‘NMT’, as per section 3.1. In that case, once the ‘NMT’ flag is verified, the receiver should wait for the new Public Key *p'* to enter into force in order to transition to the new Merkle tree root. If the applicability time of the new Merkle tree is found on the GSC OSNMA Server (Merkle tree renewal on-going), it can also be used to handle the transition.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Table 3. Operations associated with the NMA status "Don't Use"

NMA status	CPKS	PKID	CID	Required operations
Don't use	Nominal	p	i	The receiver shall not perform navigation data authentication.
Don't use	Chain Revoked (CREV)	p	i	The current TESLA chain (CID = *i*) is revoked and the material associated to this chain shall be discarded. The receiver shall retrieve the new TESLA chain material (CID = *i'*) sent in a new DSM-KROOT and verify it with the current Public Key (PKID = *p*), as per section 5.2. The receiver shall not perform navigation data authentication.
Don't use	Public Key Revoked (PKREV)	p'	i	The previous Public Key (PKID = *p*) is revoked and shall be discarded, along with its associated TESLA chain material (CID = *i*). The receiver shall retrieve the current Public Key (PKID = *p'*), either from the GSC OSNMA server or in the DSM-PKR and, in the latter case, verify it as per section 5.1.The receiver shall also retrieve the new TESLA chain material (CID = *i'*) and verify it as per section 5.2. The receiver shall not perform navigation data authentication.
Don't use	Alert Message (AM)	p	i	An OSNMA Alert Message is being transmitted. The receiver shall either retrieve the DSM-PKR and verify the presence of the alert message as per section 5.1 or retrieve the DSM-KROOT and verify the CPKS as per section 5.2. The receiver shall stop processing OSNMA data and connect to the GSC OSNMA server for further updates. The receiver shall discard any previously stored cryptographic material, including the Merkle tree root, upon reception and verification of the OAM.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

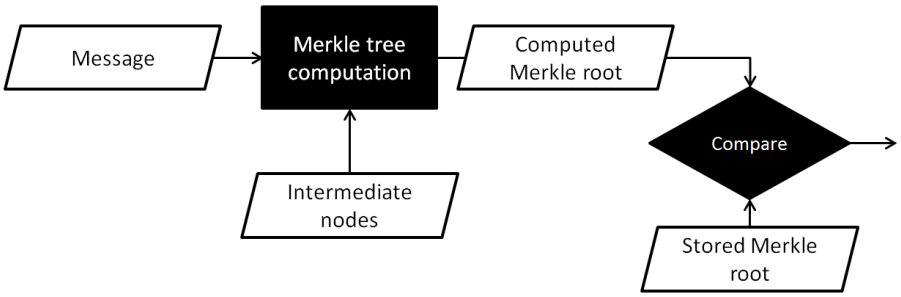
5 Overview of Cryptographic Operations

This section presents an overview of the cryptographic operations that shall be performed in order to authenticate navigation data using Galileo OSNMA data. Further details on these operations are provided in [AD.2].

5.1 Verification of the Public Key Retrieved from the SIS

The Public Key retrieved from the SIS within the DSM-PKR (as explained in section 3.2) shall be authenticated against the stored Merkle three root using the Secure Hash Algorithm (SHA). This procedure is summarised in Table 4.

Table 4. Description of the Public Key verification process

Public Key verification	
Inputs	• Message obtained by concatenating the Public Key type (NPKT), the Public Key ID (NPKID) and the Public Key (NPK) itself, as described in [AD.2] • Intermediate nodes received in the DSM-PKR [AD.2] • Merkle tree root, stored in the receiver.
Cryptographic functions	Hash functions used to build the Merkle tree: • SHA-256 [6],[10] • SHA3-256 [6][3]
Schematic	
Output	• If the computed tree root matches the stored root, the Public Key is verified. The Public Key can be used to verify the TESLA root key, as per section 5.2. • Otherwise, the verification fails. The Public Key shall be discarded. The navigation data cannot be authenticated. The event shall be logged and reported at application level.

As explained in [AD.2], based on the message ID (MID) provided in the DSM-PKR, the message is associated to a leaf m_i of the Merkle tree, and the associated intermediate nodes (ITN) $x_{j,i}$ can then be used to compute the Merkle tree root $x_{4,0}$, using the relationship between nodes described in [AD.2].
An example of the tree root computation is provided in Figure 4, assuming that the MID = 0, meaning that the message is associated to the first leaf $m_0$and that the provided intermediate nodes are ($x_{0,1}$, $x_{1,1}$, $x_{2,1}$, $x_{3,1}$), as defined in [AD.2].

[3] The system might use in future SHA3-256 for the tree generation.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

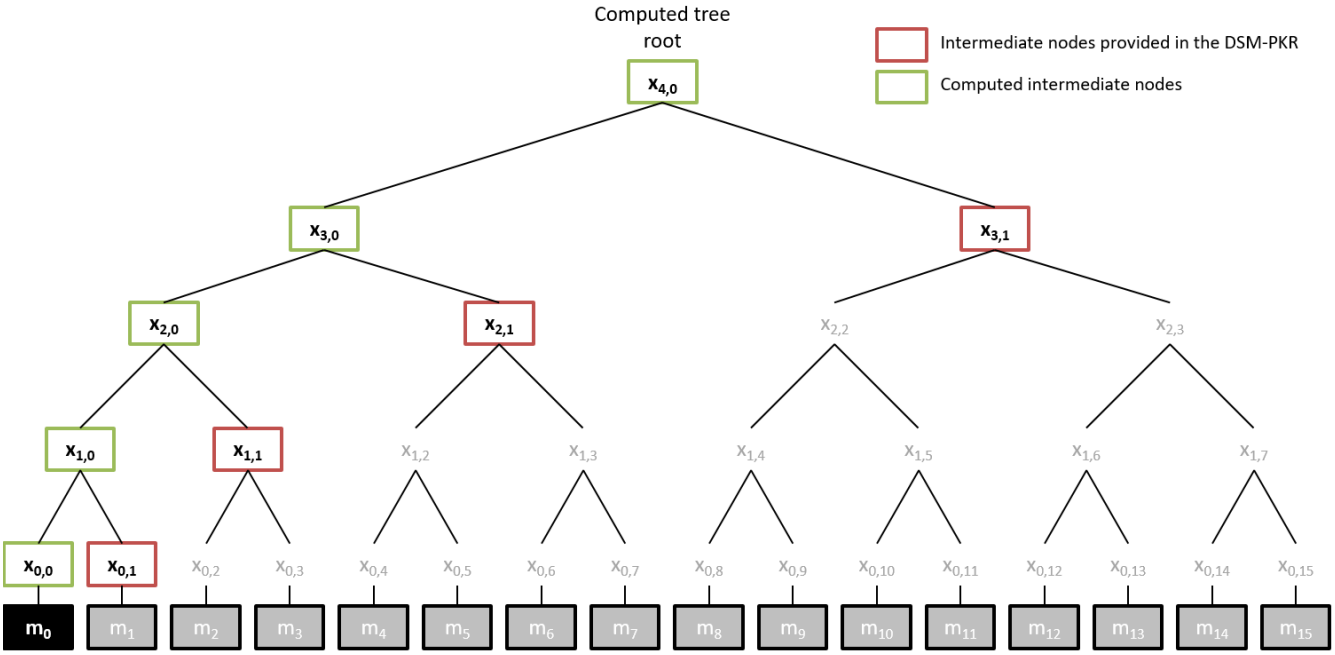


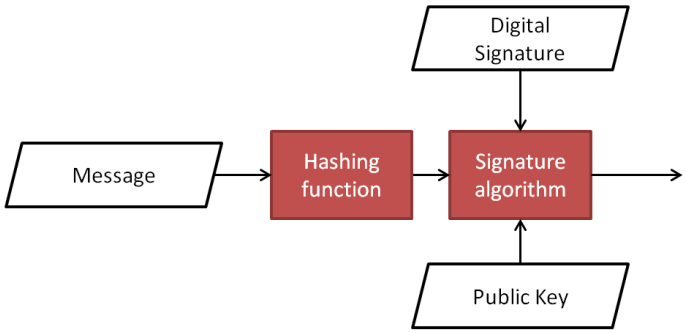
Figure 4. Verification of the Public Key through the use of Merkle tree

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

5.2 Verification of the TESLA Root Key

The TESLA root key, chain cryptographic functions, TESLA chain key length, tag length, and other TESLA chain parameters are received in the DSM-KROOT and verified using the Elliptic Curve Digital Signature Algorithm (ECDSA), as summarised in Table 5.

Table 5. Description of the TESLA root key verification process

Root Key verification	
Inputs	• Message produced by concatenating some of the fields from the DSM-KROOT, as described in [AD.2]. • Digital Signature (DS) obtained within the DSM-KROOT, as described in [AD.2] • Public Key (PK), retrieved as per section 3.2.
Cryptographic functions	Signature algorithm / Hash function combinations: • ECDSA P-256 / SHA-256 [4] • ECDSA P-521 / SHA-512 [4] The applicable function is specified in the New Public Key Type (NPKT) field of the DSM-PKR [AD.2] and is provided on the GSC OSNMA server in case the Public Key is retrieved there.
Schematic	
Output	• If the signature algorithm confirms that the received digital signature is valid, the verification is successful. The NMA Status, chain ID and CPKS fields are verified, as are the TESLA root key and the TESLA chain parameters. These parameters can be used to verify the TESLA chain keys (as per section 5.4). • Otherwise, the verification fails. The TESLA root key, TESLA chain parameters and flags are not verified and shall be discarded. The navigation data cannot be authenticated. The event shall be logged and reported at application level.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

5.3 Determination of the GST Sub-frame

As further detailed in sections 5.4 and 5.5, the GST of the sub-frame GST_{SF} in which the TESLA chain key and the tags are retrieved is needed for their verification. GST_{SF} can be computed in one of two ways, as per the next sub-sections.

5.3.1 GST Retrieval and Verification from the SIS

The GST of the sub-frame, GST_{SF}, can be derived from the GST^{SIS} retrieved from the SIS, either from the I/NAV word type 5 or from the word type 0 (Spare Word). For E1, this is the E1 sub-frame start minus 1 second. The Secondary Synchronisation Pattern (SSP) may also be exploited [AD.1].

The retrieved GST^{SIS} value shall be verified against the receiver local realisation GST^{Rx}, to ensure that:

$$|\mathrm{GST}^{SIS} - \mathrm{GST}^{Rx}| < \frac{T_L}{2} \qquad \textbf{Eq. 1}$$

Where T_L is the receiver time synchronisation requirement defined within the section 2.1. If this condition is not verified, the user shall not process OSNMA data.
In the case slow MACs only are exploited, the following condition shall be verified instead:

$$|\mathrm{GST}^{SIS} - \mathrm{GST}^{Rx}| < \frac{T_L + 300}{2} \qquad \textbf{Eq. 2}$$

The GST of the sub-frame, GST_{SF}, used in the TESLA chain key and tag verifications, can then be computed as:

$$\mathrm{GST}_{SF} = \mathrm{GST}_0 + 30 \cdot \left\lfloor \frac{\mathrm{GST}^{SIS} - \mathrm{GST}_0}{30} \right\rfloor \qquad \textbf{Eq. 3}$$

Where:

- $\lfloor n \rfloor$ is the floor operator, indicating the greatest integer less than or equal to *n*;
- GST_0 is the time of applicability of the TESLA chain in force, as per [AD.2], expressed in seconds.

5.3.2 GST Sub-frame Propagation

The GST of the sub-frame can also be propagated internally by the receiver, based on the fixed structure of the OSNMA data. In that case the GST_{SF} will be incremented by 30 seconds every time a new sub-frame is being retrieved. The initial value of GST_{SF} to be propagated shall be computed from the GST^{SIS} retrieved from the SIS and verified, as per section 5.3.1.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

5.4 Verification of the TESLA Chain Key

5.4.1 TESLA Chain Properties

As explained in [AD.2], the keys used for the verification of the tags are part of a one-way chain. Keys from the same chain are related in such a way that each key can be re-constructed by applying a function F to the next transmitted key, as in the following equation:

$$K_I = F(K_{I+1}) = trunc\left(l_k, \text{hash}_{\text{chain}}\left(K_{I+1} \| \text{GST}_{SF,I} \| \alpha\right)\right) \quad \textbf{Eq. 4}$$

Where:

- I is the index of the key in the TESLA chain, computed as per [AD.2];
- l_k is the TESLA chain key size defined [AD.2];
- $\text{hash}_{\text{chain}}$ is the specific hash function used for the TESLA chain as indicated in the HF field defined in [AD.2][4];
- $\text{GST}_{SF,I}$ is the Galileo System Time at the start of the 30-seconds sub-frame in which the TESLA chain key K_I is transmitted, computed as per section 5.3 and in the format WN (12 bits) and TOW (20 bits);
- α is the unpredictable chain pattern defined in [AD.2];
- $trunc(L, I)$ is the truncation function retaining the L most significant bits (MSB) of the input I;
- $(X||Y)$ indicates the concatenation of X and Y.

5.4.2 Number of Recursive Operations

Given the previously mentioned properties of the TESLA chain, a TESLA chain key K_I can be verified against a previously authenticated key belonging to the same TESLA chain, K_J, where $J < I$, by recursively applying the function F. The number of times the function shall be applied to perform the verification is equal to the difference between the positions of the two keys in the chain, such that:

$$K_J = F^{I-J}(K_I) \quad \textbf{Eq. 5}$$

Where:

- I is the index of the key to be verified in the TESLA chain, computed as per [AD.2]. The GST_{SF} used in the calculation is derived as per section 5.3
- J is the index of the previously verified key in the TESLA chain, computed as per [AD.2]. In the case the TESLA root key K_0 is used, $J = 0$.

This concept is further illustrated in Figure 5.

Figure 5. TESLA chain key

The function F shall be applied recursively exactly $(I - J)$ times to consider the verification successful.

5.4.3 Verification Process

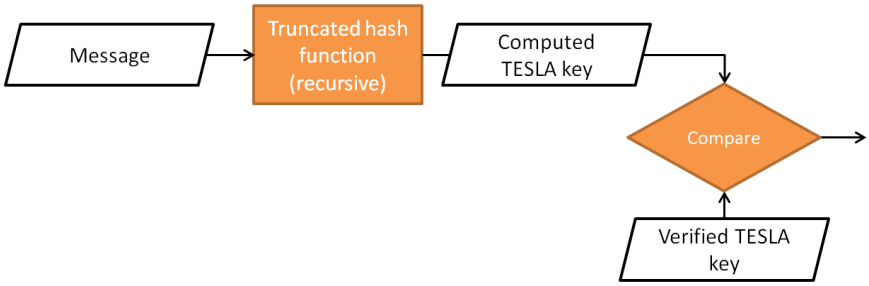
The TESLA chain key verification is performed by means of a SHA cryptographic function, as summarised in Table 6.

[4] Note that $\text{hash}_{\text{chain}}$ takes as input a message that fits an integer number of bytes, and if that is not the case, it needs to be zero-padded, as explained in [AD.2].

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

.

Table 6. Description of the TESLA chain key verification process

TESLA key verification	
Inputs	• Message produced by concatenating the TESLA chain key provided in the MACK message, the Galileo System time GST_{SF} computed as per section 5.3 and expressed in the format WN (12 bits) and TOW (20 bits), and the unpredictable chain pattern α, as defined in [AD.2] • A verified TESLA key: the TESLA root key (KROOT) verified as per section 5.2 or a previously authenticated TESLA key from the same chain.
Cryptographic functions	Hash functions: • SHA-256 [5] • SHA3-256 [6] The applicable function is specified in the Hash Function (HF) field of the DSM-KROOT [AD.2].
Schematic	
Output	• If the computed TESLA key matches the verified TESLA key, the verification is successful. The verified TESLA chain key can be used for the verification of the tag as per section 5.5 and for the verification of successive TESLA chain keys. • Otherwise, the verification fails. The TESLA chain key is not verified and shall be discarded. The event shall be logged and reported at application level.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

5.5 Verification of the Tag and Authentication of Navigation Data

As mentioned in section 4.1.2, the authentication of the navigation data is carried out in several steps:

- The user identifies the tag required to authenticate the navigation data and retrieves it, as per section 5.5.1;
- The user verifies the ADKD of the retrieved tag against the fixed sequence defined by the MACLT, as per sections 5.5.2 and 5.5.3;
- The TESLA chain key, applicable for the verification, is retrieved and verified as per section 5.5.4;
- The verification of the tags and the authentication of the navigation data are then performed as per sections 5.5.5 and 5.5.6.

Based on the successful verification of the tags, the data navigation can be authenticated as per section 5.5.6.

5.5.1 Identification of the Applicable Tag

The navigation data retrieved in a certain time interval is linked to a tag which is transmitted in an ADKD specific sub-frame, defined in [AD.2]. For example, clock and ephemeris data (ADKD 0) retrieved in the I/NAV sub-frame transmitted at time GST_{SF} is authenticated by the tags transmitted in the next sub-frame. Different parts of the navigation data are authenticated by different tags, as specified by the tag ADKD type, for different satellites, as specified by the tag PRN_A field.

Provided that all the prescriptions described here and in [AD.2] are respected, and in particular the fixed link between a tag and the data it authenticates (as discussed above), users can exploit all the properties of the Galileo I/NAV message as they are described in [AD.1] in order to optimise the performance. An example is the repetition of clock and ephemeris data (unambiguously identified by an IOD_{nav}) and including Reed-Solomon information provided within word types 17-20. In any case, a tag received in a certain sub-frame shall never be used to authenticate navigation data retrieved after the transmission of the last bit of the last Tag-Info field received in that sub-frame.

To be noted that, depending on its needs, a receiver may process all transmitted tags or only a subset of them. As an example, a user can decide to process only tags with an ADKD = 0 in order to authenticate only I/NAV ephemeris, clock and satellite health data.

5.5.2 Tags Sequence Verification

In order to verify a tag, the receiver shall first verify that its associated ADKD corresponds to the one predefined in the MAC Look-up Table [AD.2]. The entry value to the MAC Look-up Table is provided by the MACLT field of the DSM-KROOT and is fixed for a given TESLA chain.

The MAC Look-up Table can define some slots in the tag sequence as flexible (‘FLX’), which means that the ADKD values of these tags are not fixed and can be allocated dynamically. The Tag-Info field of these tags is then authenticated using the MACSEQ, as described in section 5.5.3.

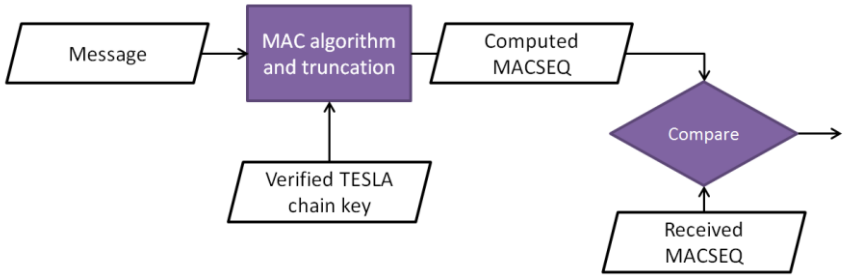
If the ADKD type of a received tag differs from the one defined in the MAC Look-up Table sequence, the tag shall be discarded.

5.5.3 Flexible Tags Sequence Verification

To exploit the tags defined as flexible in the MAC Look-up Table, the MACSEQ verification shall be performed to verify the authenticity of the associated Tag-Info sections. The MACSEQ is verified with the same key and MAC function as the Tag_0 in the same MACK message [AD.2], as summarised in Table 7. The MAQSEC verification is not mandatory in case no FLEX tags are broadcast.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Table 7. Description of the MACSEQ verification process

MACSEQ verification	
Inputs	• Message produced by concatenating fields as described in [AD.2] • MACSEQ received, as described in [AD.2] • Applicable TESLA chain key, as specified in 5.5.4, verified as described in section 5.4
Cryptographic functions	MACSEQ verification algorithm: • HMAC-SHA-256 [7] • CMAC-AES [8] [9] The applicable function is specified in the MAC Function (MF) field of the DSM-KROOT [AD.2].
Schematic	
Output	• If the locally computed MACSEQ matches the received MACSEQ, the verification is successful. The FLEX tags sequence is verified. • Otherwise, the verification fails. The related flexible tags shall be discarded. The event shall be logged and reported at application level.

5.5.4 Identification of the Applicable TESLA Chain Key

A tag retrieved in a certain MACK message has to be verified with the TESLA chain key broadcast in the next MACK message, as specified in [AD.2]. It is important to note that, given the properties of the TESLA chain keys mentioned in section 5.4.1, even if the receiver did not retrieve the key required for the verification from the SIS, it can compute this key from any retrieved key which has a subsequent position in the chain. A TESLA chain key shall be verified as per section 5.4 before being used for the tag verification.

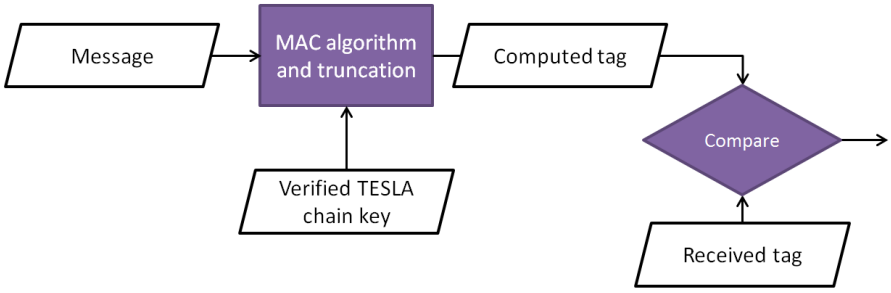
5.5.5 Tag Verification

The tag verification is performed to verify the authenticity of the navigation data it is associated to. A summary of the tag verification scheme is provided in Table 8.

Table 8. Description of the tag verification process

Tag verification	
Inputs	• Message produced by concatenating fields, as described in [AD.2]. • Tag received in the MACK message, as described in [AD.2] • Applicable TESLA chain key, as specified in 5.5.4, verified as in section 5.4
Cryptographic functions	MAC verification algorithm: • HMAC-SHA-256 [7] • CMAC-AES [8] [9]

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Tag verification	
	The applicable function is specified in the MAC Function (MF) field of the DSM-KROOT [AD.2].
Schematic	
Output	• If the locally computed tag matches the received tag, the verification is successful. • Otherwise, the verification fails. The tag shall be discarded. The event shall be logged and reported at application level.

Note that the system may transmit dummy tags, indicated by a COP field set to zero, as per [AD.2]. These tags have to be verified following the above process, with the difference that the message is to be replaced with a sequence of zeros having the same length as the corresponding message associated to the tag ADKD type, as defined in [AD.2].

5.5.6 Navigation Data Authentication

Navigation data authentication is achieved after verifying a minimum number of tag bits, L_t^{min}, associated to the same navigation data set. The value of the minimum equivalent tag length L_t^{min} will be provided in the future Galileo Authentication Service Definition Document. The minimum equivalent tag length L_t^{min} may evolve during the service provision and therefore this parameter should be configurable in the receiver.

If the tag being used for the authentication has a length l_t such that $l_t \geq L_t^{min}$, then the navigation data is authenticated if the tag is verified as per section 5.5.5.

If the tag length l_t is such that $l_t < L_t^{min}$, tag accumulation shall be performed. The steps below shall be followed:

- The receiver shall accumulate N_t tags, such that $l_t \cdot N_t \geq L_t^{min}$. All tags to be accumulated must correspond to the same data to be authenticated. In particular, a tag corresponding to a specific PRN_D and ADKD type can be accumulated with any tag corresponding to the same PRN_D and ADKD which was retrieved within the time interval indicated by the COP field, T_{COP}, as described in [AD.2]. The receiver shall also register the sub-frame time GST_{SF} (computed as per section 5.3) in which the tags are retrieved.
- The receiver shall then ensure that the registered GST_{SF} values are coherent with the sub-frames in which the tags are retrieved, e.g. two tags retrieved in two consecutive sub-frames will have an offset of 30 seconds between their GST_{SF}.
- Once N_t tags have been retrieved and the coherency between the registered GST_{SF} is verified, the receiver shall verify the tags using their associated and previously authenticated TESLA chain keys, as described in section 5.5.5.
- If one of the verifications fails, all the tags shall be discarded, the navigation data is not authenticated. If all the tags are verified, the navigation data is authenticated.

If a tag authentication fails, new navigation data shall be retrieved and tag accumulation restarted.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

References

[1] International Organization for Standardization, “ISO/IEC 29192-7:2019 Information security — Lightweight cryptography — Part 7: Broadcast authentication protocols.” 2019.

[2] Method and system to optimise the authentication of radio navigation signals, Patent application PCT/EP2015/056120, 23/03/2015, European Union represented by European Commission

[3] Digitally-signed satellite radio-navigation signals, Patent application PCT/EP2014/064285, 04/07/2014, European Union represented by European Commission

[4] National Institute of Standards and Technology, “FIPS PUB 186-4: Digital Signature Standard (DSS).” 2013.

[5] National Institute of Standards and Technology, “FIPS PUB 180-4: Secure Hash Standard (SHS).” 2012.

[6] National Institute of Standards and Technology, “FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions.” 2015.

[7] National Institute of Standards and Technology, “FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (HMAC).” 2008.

[8] National Institute of Standards and Technology, “NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication.” 2004.

[9] International Organization for Standardization, “ISO/IEC 9797-1:2011: Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher.” 2011.

[10] National Institute of Standards and Technology, “NIST Special Publication 800-208: Recommendation for Stateful Hash-Based Signature Schemes.” 2020.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

List of acronyms

Acronym	Definition
ADKD	Authentication Data & Key Delay
AES	Advanced Encryption Standard
BID	Block ID
CID	Chain ID
CMAC	Cipher-based Message Authentication Code
CPKS	Chain and Public Key Status
CRC	Cyclic Redundancy Check
CREV	Chain Revoked
DSM	Digital Signature Message
DSM-KROOT	DSM for a KROOT
DSM-PKR	DSM for a PKR
ECDSA	Elliptic Curve Digital Signature Algorithm
EOC	End Of Chain
GSC	European GNSS Service Centre
GST	Galileo System Time
HF	Hash Function
HKROOT	Header and KROOT
HMAC	Hash-based Message Authentication Code
ICD	Interface Control Document
IDD	Internet Data Distribution
IOD	Issue of Data
ITN	Intermediate Tree Node
KROOT	Root Key
MAC	Message Authentication Code
MACK	MAC and Key
MACLT	MAC Look-up Table
MACSEQ	MAC Sequence
MF	MAC Function
MID	Message ID
MSB	Most Significant Bit
NB	Number of Blocks
NMA	Navigation Message Authentication
NPK	New Public Key
NPKID	New Public Key ID
NPKT	New Public Key Type
OAM	OSNMA Alert Message
OS	Open Service
OSNMA	Open Service Navigation Message Authentication
PK	Public Key
PKI	Public Key Infrastructure
PKID	Public Key ID

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Acronym	Definition
PKR	Public Key Renewal
PKREV	Public Key Revocation
PRN	Pseudo Random Noise
PVT	Position Velocity Time
SHA	Secure Hash Algorithm
SIS	Signal In Space
TESLA	Timed Efficient Stream Loss-Tolerant Authentication
TOW	Time of Week
WN	Week Number

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

ANNEX A Examples of OSNMA Verifications

This Annex provides a set of OSNMA data with the purpose of supporting developers in their implementation of the OSNMA protocol with respect to data parsing and verifications using cryptographic operations.

A.1 Binary Data Representation Conventions

The hexadecimal string values in the following sections are right padded (LSB) with zeroes whenever the number of bits of the value to be represented is not a multiple of 8. Strings in hexadecimal and binary values are always in big-endian bytes order with the bits transmitted first in the MSB position (which appears to the left of the string). Binary values are prepended by the '0b' prefix, hexadecimal values are prepended by the '0x' prefix, decimal numbers have no prefix.

A.2 OSNMA Configuration

The test vectors correspond to the OSNMA configuration summarised in the following table.

Parameters	Settings
Tag size	40 bits
Key size	128 bits
Digital Signature Algorithm	ECDSA P-256
Hash Function	SHA-256
MAC Function	HMAC-SHA-256
MACLT	34

A.3 NMA Header

This section reports the NMA Header to be used for the verifications presented in the following sections.

NMA Header = 0b10000010 (8 bits)

As per [AD.2], the header can be interpreted as follows.

Field	Field value	Interpretation
NMAS	0b10	Operational
CID	0b00	0
CPKS	0b001	Nominal
Reserved	0b0	N/A

The header indicates that:

- the OSNMA is in Operational mode;
- the TESLA chain in force is the one with ID 0;
- the public key and chain in force are nominal (i.e. no renewal or revocation processes are taking place).

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

A.4 DSM-KROOT

The DSM-KROOT message is reported below.

DSM-KROOT = 0x2210492204E060610BDF26D77B5BF8C9CBFCF70422081475FD445DF0FFF8CD88299FA4605800207BFEBEAC55024053F30F7C69B35C15E60800AC3B6FE3ED0639952F7B028D86867445961FFE94FB226BFF7006E0C451EE3F8728C177FB5E130DA4B44BBE7EC29522 (832 bits)

A.4.1 DSM-KROOT interpretation

The following chain parameters are extracted from the first 104 bits of the DSM-KROOT message.

Field	Field value	Interpretation
NB_{DK}	0x2	8 blocks
PKID	0x2	2
CIDKR	0b00	0
Rsvd1	0b01	N/A
HF	0b00	SHA-256
MF	0b00	HMAC-SHA-256
KS	0x4	128 bits
TS	0x9	40 bits
MACLT	0x22	34
Rsvd2	0x0	N/A
WN_K	0x4E0	1248
$TOWH_K$	0x60	96 hours
α	0x610BDF26D77B	0x610BDF26D77B

The remaining bits of DSM-KROOT contain the KROOT itself, its digital signature (DS) and a padding sequence (PDK).

KROOT = 0x5BF8C9CBFCF70422081475FD445DF0FF (128 bits)

DS = 0xF8CD88299FA4605800207BFEBEAC55024053F30F7C69B35C15E60800AC3B6FE3ED0639952F7B028D86867445961FFE94FB226BFF7006E0C451EE3F8728C177FB (512 bits)

P_{DK} = 0x5E130DA4B44BBE7EC29522 (88 bits)

A.4.2 DSM-KROOT verification

In order to verify the DSM-KROOT content, the message M is produced by concatenating some of the DSM-KROOT fields, as described in [AD.2]. The message to be signed is:

M = 0x8210492204E060610BDF26D77B5BF8C9CBFCF70422081475FD445DF0FF (232 bits)

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

The digital signature can be verified with the following public key, provided here in certificate format:

```
-----BEGIN CERTIFICATE-----
MIICbDCCAhKgAwIBAgIQR8TxQ8P6YaUpTmPVVysBYjAKBggqhkjOPQQDAjA3MQsw
CQYDVQQGEwJFUzEOMAwGA1UECgwFRVVTUEExGDAWBgNVBAMMD0VVU1BBIE9TTk1B
IElDQTAeFw0yMzA3MjAxMTIyMzBaFw0yNTA4MDgxMTMzMDBaMDoxCzAJBgNVBAYT
AkVTMQ4wDAYDVQQKDAVFVVNQQTEbMBkGA1UEAwwSRVVTUEEgT1NOTUEgRUUgUEtS
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEA7LOZLwge92LxN+FkYf8toYyDWP/
oJFBD8FY+7d5gOqIaE2RjPAnKI68s/OK/HPgoLkO2ijQ8xAZyDdPB1dHSaOB/DCB
+TAdBgNVHQ4EFgQUaiIWWJsjyUNBPLb4nZMP4P5qPFQwHwYDVR0jBBgwFoAUIMBU
ha+CrpY8vN/BuQXe10ZyMqMwYwYDVR0gBFwwWjBOBgsrBgEEAYPVEQEBATA/MD0G
CCsGAQUFBwIBFjFodHRwczovL3d3dy5nc2MtZXVyb3BhLmV1L2dzYy1wcm9kdWN0
cy9PU05NQS9QS0kvMAgGBgQAj3oBAjBCBgNVHR8EOzA5MDegNaAzhjFodHRwczov
L3d3dy5nc2MtZXVyb3BhLmV1L2dzYy1wcm9kdWN0cy9PU05NQS9QS0kvMA4GA1Ud
DwEB/wQEAwIHgDAKBggqhkjOPQQDAgNIADBFAiEA6buQjuUM89pXcePQ0uqsGwDz
UenYuwqyTIplUnmfQ/YCIBBlL2r4JiBC/wlr0I0LdRUkv+T+YMNuLTEy7WVsXIsU
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIICMjCCAdigAwIBAgIUVmo7J+pBUUDP2fWjLPvDvY2HSIcwCgYIKoZIzj0EAwIw
NzELMAkGA1UEBhMCRVMxDjAMBgNVBAoMBUVVU1BBMRgwFgYDVQQDDA9FVVNQQSBP
U05NQSBJQ0EwHhcNMjMwNzIwMTAzMTM4WhcNMjUwNzIwMTAzMTM4WjA3MQswCQYD
VQQGEwJFUzEOMAwGA1UECgwFRVVTUEExGDAWBgNVBAMMD0VVU1BBIE9TTk1BIElD
QTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABE+Z3W0RHPYZfUolnhAUp4tRBSmv
1xdSZRhdcmYowNpc15N+x7xmDiYGPQwSBufoH2zU6HWp2bEpQhLEn1qeRhyjgcEw
gb4wHQYDVR0OBBYEFCDAVIWvgq6WPLzfwbkF3tdGcjKjMB8GA1UdIwQYMBaAFCDA
VIWvgq6WPLzfwbkF3tdGcjKjMBIGA1UdEwEB/wQIMAYBAf8CAQAwWAYDVR0gBFEw
TzBNBgorBgEEAYPVEQEBMD8wPQYIKwYBBQUHAgEWMWh0dHBzOi8vd3d3LmdzYy1l
dXJvcGEuZXUvZ3NjLXByb2R1Y3RzL09TTk1BL1BLSS8wDgYDVR0PAQH/BAQDAgEG
MAoGCCqGSM49BAMCA0gAMEUCIQDIwR/ODYlDt5WMFjuJIxtgDePgOWdXda2Mx/7/
ZorpKgIgQ/vrSy9PUxkO116V8HWJN7l05FSpVkqW9fwZNVYnoLU=
-----END CERTIFICATE-----
```

The corresponding hexadecimal value of the public key is:

```
0x0403B2CE64BC207BDD8BC4DF859187FCB686320D63FFA091410FC158FBB77980EA8
8684D918CF027288EBCB3F38AFC73E0A0B90EDA28D0F31019C8374F07574749
```
(520 bits)

Note that the public key is broadcast via SIS in compressed ECDSA format, as provided in section A.7.1

A.4.3 PDK verification

This section describes the verification of the DSM-KROOT padding bits, PDK. As this verification is not mandatory, it is not covered in the OSNMA Receiver Guidelines. The description below is provided for the sake of completeness and in the interest of the developer.

As per [AD.2], the message hashed for the verification is composed of the concatenation of the message M used for the DSM-KROOT verification and of the digital signature DS:

```
0x8210492204E060610BDF26D77B5BF8C9CBFCF70422081475FD445DF0FFF8CD88299
FA4605800207BFEBEAC55024053F30F7C69B35C15E60800AC3B6FE3ED0639952F7B02
8D86867445961FFE94FB226BFF7006E0C451EE3F8728C177FB
```
(744 bits)

The message is then hashed with the SHA-256 function, leading to:

```
0x5E130DA4B44BBE7EC29522C957244699A362EA030F32A852FE98776F9B23E5FA
```
(256 bits)

This result is then truncated to the length of P_{DK} and compared bit-by-bit to the received field.

```
0x5E130DA4B44BBE7EC29522
```
(88 bits)

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

A.5 TESLA Chain Key

A.5.1 TESLA Chain Key Interpretation

This section reports some of the TESLA chain keys and their associated index in the chain. It shall be noted that the key with index 0 corresponds to the root key transmitted over the DSM-KROOT section, as explained in section A.4 . The WN and TOW corresponding to the GSTSF of the sub-frame in which the TESLA chain key is *transmitted* is also reported.

Key Index	WN	TOW	Key (128 bits)
120	1248	349170	0x01D3E3E2667A0A1894D04BDD98CABA17
119	1248	349140	0x6C9292CEF363E38C563CA756FACE5B89
…			…
14	1248	345990	0xED2B77AAED7B633B58A69551FC388421
13	1248	345960	0xCD5653B3E47D3EA12862523BF943BA0B
12	1248	345930	0x339533D046155631AF348D371EDA501A
…			…
4	1248	345690	0x69C00AA7364237A65EBF006AD8DDBC73
3	1248	345660	0x2E9C651C410CFF23D370497D28AB4B14
2	1248	34630	0x2DC3A3CDB117FAADB83B5F0B6FEA88EB
1	1248	345600	0xEFF999040E19B570835060BEBD23ED92
0 (KROOT)	1248	345570	0x5BF8C9CBFCF70422081475FD445DF0FF

Note that as per [AD.2], the GSTSF associated to the KROOT is GST0 – 30 seconds, where GST0 is the time provided in the DSM-KROOT, in the form of WNK and TOWHK (see section A.4).

A.5.2 TESLA Chain Key Verification

This section describes the verification of the second key of the TESLA chain, K_2, against the root key, K_0.

As a first step, K_2 is used to reconstruct K_1, the previous key in the chain. The message to hash for this step is the concatenation of K_2, GST_{SF}, and α. Note that, as per [AD.2], GST_{SF} corresponds to the time of the sub-frame in which the key being computed, K_1, was transmitted (here WN = 1248 and TOW = 345600 sec).

0x2DC3A3CDB117FAADB83B5F0B6FEA88EB4E054600610BDF26D77B (208 bits)

Note that as the length of the message fits an integer number of bytes, no padding is needed. The message is then hashed with the HF function (SHA-256) and the following hash is obtained:

0xEFF999040E19B570835060BEBD23ED92EEBA1E6AE8EE5A80CBBD62E6C711DA73
(256 bits)

The first key of the chain, K_1, is obtained by truncation, retaining the first KS = 128 bits of the hash:

0xEFF999040E19B570835060BEBD23ED92 (128 bits)

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Another step has to be performed to generate the local replica of the KROOT, K_0. This time, the message to hash is the concatenation of K_1, GST_{SF} and α. GST_{SF} is the time associated to K_0, corresponding to WN = 1248 and TOW = 345570 sec.

0xEFF999040E19B570835060BEBD23ED924E0545E2610BDF26D77B (208 bits)

The message is hashed and the following hash is obtained:

0x5BF8C9CBFCF70422081475FD445DF0FF4F5BB0B311287DEAFEDB4FD7A171DEA3 (256 bits)

K_0 is obtained after truncation:

0x5BF8C9CBFCF70422081475FD445DF0FF (128 bits)

The computed K_0 local replica can now be verified against the root key obtained from the DSM-KROOT, performing a bit-by-bit comparison. Note that following these steps, K_2 is verified. This implies that the following key, K_3, can be verified against K_2, without having to hash back to K_0.

A.6 Tags

A.6.1 MACK Message Interpretation

The tags are transmitted within the MACK message. The MACK messages transmitted by the satellite E02 at WN = 1248, TOW = 345630 seconds and at WN = 1248, TOW = 345660 sec are reported below.

0x4DA9D1D763DE4FDE4439ED5A180F765C4FFE3D1B0FE7025D4D0A02CF4947DAFF180C0F97FF3ABD2312C42DC3A3CDB117FAADB83B5F0B6FEA88EB0000 (480 bits)

0xE37BC4F858B08F0726A9000C12057BB238C883024FC8FB247323050F91F230FE4D02CF2F1C4D28C1220F2E9C651C410CFF23D370497D28AB4B140000 (480 bits)

Six tags can be extracted from each of this MACK message, they are reported in the following table along with their position within the MACK message (CTR field), their associated PRND and ADKD values.

TOW	CTR	Tag	PRND	ADKD	COP
345630	1 (Tag_0)	0x4DA9D1D763	2	0	15 (0xF)
345630	2	0xDE4439ED5A	24 (0x18)	0 (0x0)	15 (0xF)
345630	3	0x765C4FFE3D	27 (0x1B)	0 (0x0)	15 (0xF)
345630	4	0xE7025D4D0A	2 (0x02)	12 (0xC)	15 (0xF)
345630	5	0x4947DAFF18	12 (0x0C)	0 (0x0)	15 (0xF)
345630	6	0x97FF3ABD23	18 (0x12)	12 (0xC)	4 (0x4)
345660	1 (Tag_0)	0xE37BC4F858	2	0	15 (0xF)
345660	2	0x0726A9000C	18 (0x12)	0 (0x0)	5 (0x5)
345660	3	0x7BB238C883	2 (0x02)	4 (0x4)	15 (0xF)
345660	4	0xC8FB247323	5 (0x05)	0 (0x0)	15 (0xF)
345660	5	0x91F230FE4D	2 (0x02)	12 (0xC)	15 (0xF)
345660	6	0x2F1C4D28C1	34 (0x22)	0 (0x0)	15 (0xF)

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

In addition, one MACSEQ is transmitted in each MACK block, corresponding to `0xDE4` at TOW = 345630 sec and `0xB08` at TOW = 345660 sec.

A.6.2 Tag Sequence Verification

As part of the tag verification, its ADKD shall be verified against the one predefined in the MAC look-up table. The sequence corresponding to the MACLT 34 is recalled here.

ID	Msg	n_t	Sequence	
34	2	6	00S, FLX, 04S, FLX, 12S, 00E	00S, FLX, 00E, 12S, 00E, 12E

It can be seen that the ADKD sequence matches the one of the MACLT, where the self-authenticating tags (indicated with 'S' in the MACLT) are associated with PRN 2 and where the FLX are transmitted as cross-authenticating ADKD0 (see section A.6.4 for the MACSEQ verification). Also, it can be seen that the order of the two MACK messages is respected, as the sequence transmitted in the first 30 seconds of the GST minute corresponds to the first part of the MACLT sequence.

A.6.3 Associated Navigation Data

As an example, some of the tags retrieved in the sub-frame WN = 1248, TOW = 345660 sec are verified. These tags are authenticating the navigation data transmitted in the previous sub-frame, at WN = 1248 and TOW = 345630 sec. The following table reports the E1-B I/NAV pages transmitted by E02, at the reported times of transmission. They are used in the following section for the tag verification.

WN	Time of transmission	Page part index	E1-B I/NAV page (120 bits)
1248	345631	1	`0x020EB72D6AB6C9DEBDBF3C87EE63C0`
1248	345632	2	`0xBDFA60936A7475EAAAAA7085944100`
1248	345633	3	`0x040E82001A000E5910006D31F00000`
1248	345634	4	`0xA68070D8F793F7AAAAAA68A91DCAC0`
1248	345635	5	`0x06FFFFFFFF0000011248E089E24A80`
1248	345636	6	`0x8C46B2510E7B56AAAAAA7A56FF0BC0`
1248	345637	7	`0x09A23A5555555555552A0110156B40`
1248	345638	8	`0x85F48E4603DD972AAAAA7B8D5E4100`
1248	345639	9	`0x0AADC96FE3DEC7FDC7FFF0FFBDFF00`
1248	345640	10	`0xA6083B13FF8F46EAAAAA723127CAC0`
1248	345641	11	`0x125A95F82358E0521768C7435B86C0`
1248	345642	12	`0x871C9883F9C0976AAAAA5A59088BC0`
1248	345643	13	`0x148BB93DF4BB237D30163105D10840`
1248	345644	14	`0x851E42134280B3EAAAAA6CFB898100`
1248	345645	15	`0x1010D97DB7D45ADCB5A831D98C00C0`

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

WN	Time of transmission	Page part index	E1-B I/NAV page (120 bits)
1248	345646	16	0x9990055251F6BFEAAAAA6F7FCF0AC0
1248	345647	17	0x00955555555555555555555538140
1248	345648	18	0x918BC4060303E5EAAAAA7BD6FCCBC0
1248	345649	19	0x00955555555555555555555538140
1248	345650	20	0x918C503FCEAF48EAAAAA63697AC100
1248	345651	21	0x010E96441475A496001DACFBAA0500
1248	345652	22	0x86FCF304B10B70EAAAAA44E2740AC0
1248	345653	23	0x030EBFF0DC080EC363843AC5344440
1248	345654	24	0xA39AFFE8F36C45EAAAAA7BD4640BC0
1248	345655	25	0x0554A02C22607F7FC009C0A8C6EA80
1248	345656	26	0xAAAAB3FEAB6E0EEAAAAA754D4B0100
1248	345657	27	0x00955555555555555555555538140
1248	345658	28	0x918E47D7C2DBFAAAAAAA4A4B4DCAC0
1248	345659	29	0x1010D97DB7D45ADCB5A831D98C00C0
1248	345660	30	0x99902DE23AC0002AAAAA5EC4050BC0

A.6.4 MACSEQ Verification

In the case the flexible tags are processed, the MACSEQ verification is mandatory. As per [AD.2], the message m used for the verification is obtained by concatenating the PRN of the satellite transmitting the OSNMA data, PRN_A, the GST at the start of the sub-frame, GST_{SF}, corresponding to WN = 1248 and TOW = 345660 sec, and the Tag-Info fields on the flexible tags (2nd and 4th tags in the sub-frame).

m = 0x024E05463C1205050F (72 bits)

The MACSEQ is verified with the key of the next sub-frame, K_4, applying the MAC function indicated by the MF field, HMAC-SHA-256. The result of the MAC function is:

0xB08E8714188441EBEBF9260BFC8E4772AAEE92945A39EC86A60219BBDE7FECEE (256 bits)

This result is then truncated to the 12 more significant bits, leading to 0xB08. The value is compared bit-by-bit with the retrieved MACSEQ. As the values are the same, the verification is successful.

A.6.5 Tags Verification

A.6.5.1 Tag0 Verification

This section describes the verification of the tag received in first position (Tag0) in the second of the two MACK messages provided in section A.6.1 (WN = 1248, TOW = 345660 sec). As per [AD.2], Tag0 is associated to ADKD0. The navigation data to be used for the verification shall be retrieved in the previous sub-frame, reported in section A.6.3 and is the following:

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

ADKD0 navigation data =
0x0E96441475A496001DACFBAA0506FC0EB72D6AB6C9DEBDBF3C87EE63FDFA0EBFF0D
C080EC363843AC53444639AC3A0800680039644001B4C7C0009A015280B08981FDFF0
00 (552 bits, 549 bits without padding)

Note that the ADKD0 navigation data has been right zero padded in order to be expressed in hexadecimal format.

The message m0 used for the verification is obtained by concatenating the different fields described in [AD.2]. It is reported below.

m_0 =
0x024E05463C0183A591051D692580076B3EEA8141BF03ADCB5AADB277AF6FCF21FB9
8FF7E83AFFC370203B0D8E10EB14D1118E6B0E82001A000E5910006D31F000268054A
02C22607F7FC00 (600 bits)

The message is hashed with the key of the next sub-frame, K_4, applying the MAC function indicated by the MF field, HMAC-SHA-256. The result of the MAC function is:

0xE37BC4F858AE1E5CFDC46F054B1F47B9D2EA61E1EF09115CFE706852BFF23A83
(256 bits)

This result is finally truncated to the tag length indicated by the TS field, 40 bits.

0xE37BC4F858 (40 bits)

The value computed locally is then compared bit-by-bit with the retrieved Tag0. As the values are the same, the verification is successful.

A.6.5.2 ADKD4 Verification

This section describes the verification of the ADKD4 tag received in 3rd position in the first of the two MACK messages provided in section A.6.1 (WN = 1248, TOW = 345660 sec). The navigation data used for ADKD 4 verification shall also be retrieved from the previous sub-frame reported in section A.6.3 and is the following:

ADKD 4 navigation data = 0xFFFFFFFF0000011248E089E25FF7BFE4C100 (144 bits, 141 without padding)

The message *m* used for the verification of the tag associated with ADKD4, as described in [AD.2], is reported below. Also note that as the message does not fit an integer number of bytes, a 1-bit zero padding is added.

m = 0x02024E05463C03BFFFFFFFC00000449238227897FDEFF93040 (200 bits)

The message is hashed with the chain key of the next sub-frame, K_4, applying the HMAC-SHA-256 function. The result of the MAC function is:

0x7BB238C883C06A2B508FE63FB7F4F54D44ABEE4DCEB93DCF65CB3A5B814A34E9
(256 bits)

This result is finally truncated to the tag length indicated by the TS field, 40 bits.

0x7BB238C883 (40 bits)

The value computed locally is then compared bit-by-bit with the retrieved tag. As the values are the same, the verification is successful.

A.6.5.3 ADKD12 Verification

This section describes the verification of the ADKD12 tag received in 5th position in the first of the two MACK messages provided in section A.6.1 (WN = 1248, TOW = 345660 sec). The navigation data used for ADKD12 verification is the same as the one used for ADKD0, reported in section A.6.5.1 .

The message *m* used for the verification of the tag associated with ADKD 12 is reported below.

m =
0x02024E05463C0583A591051D692580076B3EEA8141BF03ADCB5AADB277AF6FCF21F

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

B98FF7E83AFFC370203B0D8E10EB14D1118E6B0E82001A000E5910006D31F00026805
4A02C22607F7FC00 (608 bits)

The message is hashed with the chain key transmitted with 10 sub-frame additional delay, corresponding to K_{14}, applying the HMAC-SHA-256 function. The result of the MAC function is:

0x91F230FE4D4D4809BEBC85AA29FE981E2CD829E20C1A51E121472549697FD29C
(256 bits)

The result is finally truncated to the tag length, 40 bits.

0x91F230FE4D (40 bits)

The value computed locally is then compared bit-by-bit with the retrieved tag. As the values are the same, the verification is successful.

A.7 DSM-PKR

The DSM-PKR is reported below:

DSM-PKR =
0x717CBE05D9970CFC9E22D0A43A340EF557624453A2E821AADEAC989C405D78BA069
56380BAB0D2C939EC6208151040CCFFCF1FB7156178FD1255BA0AECAAA253F7407B6C
5DD4DF059FF8789474061301E1C34881DB7A367A913A3674300E21EAB124EF508389B
7D446C3E2ECE8D459FBBD3239A794906F5B1F92469C640164FD87120303B2CE64BC20
7BDD8BC4DF859187FCB686320D63FFA091410FC158FBB77980EAB8884C0D33D6 (1352
bits)

A.7.1 DSM-PKR Interpretation

The following public key parameters are extracted from the DSM-PKR message.

Field	Field value	Corresponding value
NB_{DP}	0x7	13 blocks
MID	0x1	Merkle tree leaf m1
NPKT	0x1	ECDSA P-256
NPKID	0x2	2
P_{DP}	0xB8884C0D33D6	0xB8884C0D33D6

The message also contains the compressed ECDSA public key value:

0x0303B2CE64BC207BDD8BC4DF859187FCB686320D63FFA091410FC158FBB77980EA
(264 bits)

It also contains a set of four Merkle tree intermediate nodes. As per [AD.2], the intermediate nodes transmitted with the Merkle tree leaf m_0 are $x_{0,0}$, $x_{1,1}$, $x_{2,1}$ and $x_{3,1}$. They are reported below.

Idx	Node (256 bits)
$x_{0,0}$	0x7CBE05D9970CFC9E22D0A43A340EF557624453A2E821AADEAC989C405D78BA06
$x_{1,1}$	0x956380BAB0D2C939EC6208151040CCFFCF1FB7156178FD1255BA0AECAAA253F7
$x_{2,1}$	0x407B6C5DD4DF059FF8789474061301E1C34881DB7A367A913A3674300E21EAB1

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Idx	Node (256 bits)
$x_{3,1}$	0x24EF508389B7D446C3E2ECE8D459FBBD3239A794906F5B1F92469C640164FD87

A.7.2 DSM-PKR Verification

In order to perform the verification of the DSM-PKR, the received message is verified against a known Merkle tree root, which is retrieved from the GSC OSNMA server and is reported below:

0xA10C440F3AA62453526DB4AF76DF8D9410D35D8277397D7053C700D192702B0D (256 bits)

As a first step of the verification, the Merkle tree leaf identified by the MID, m_1, is obtained by concatenating the NPKT, NPKID and NPK fields, as per [AD.2].

m_1= 0x120303B2CE64BC207BDD8BC4DF859187FCB686320D63FFA091410FC158FBB77980EA (272 bits)

This message is then hashed to obtain the intermediate node $x_{0,1}$.

$x_{0,1}$= 0xDBF0047C20457BD70174BEA92E62F5980EF6B5E328F2AD165554A96D0F7A1F5A (256 bits)

This initial node is then used together with the received $x_{0,0}$ value to compute $x_{1,0}$, as per the formula in [AD.2]. The operation is iterated until the Merkle tree root, $x_{4,0}$, is obtained. The computed intermediate tree nodes are listed below.

$x_{1,0}$= 0x6853057D04A17DFFB7F0C2AC0ACFAF994CF8876D782C436DFE9031FC4DC15D93 (256 bits)

$x_{2,0}$ = 0x2A7BDEBA1495D079808B227DCEDE1EAC82D02B925CCC1BE9F29B7C183891B7D6 (256 bits)

$x_{3,0}$ = 0x601BCFC32E2CC7DAA0FCE303988152A3D111ACCD8D7BBC8221B7ECAF1F19E5F9 (256 bits)

$x_{4,0}$ = 0xA10C440F3AA62453526DB4AF76DF8D9410D35D8277397D7053C700D192702B0D (256 bits)

The locally generated Merkle tree root, $x_{4,0}$, is compared bit-by-bit to the stored value. As the values are the same, the received public key and its associated parameters are verified.

A.7.3 Verification of the P_{DP}

This section describes the verification of the DSM-PKR padding bits, P_{DP}. As this verification is not mandatory, it is not covered in the OSNMA Receiver Guidelines. The description below is provided for the sake of completeness and in the interest of the developer.

The message to be hashed for this verification is the concatenation of the Merkle tree root and the Merkle tree leaf, m_1, transmitted in the DSM-PKR:

0xA10C440F3AA62453526DB4AF76DF8D9410D35D8277397D7053C700D192702B0D120303B2CE64BC207BDD8BC4DF859187FCB686320D63FFA091410FC158FBB77980EA (528 bits)

The message is then hashed with the SHA-256 function, leading to:

0xB8884C0D33D62B78A858BF25477A912BFEA301D8E73D866B0ED5660FACEAB66C (256 bits)

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

This result is further truncated to the length of PDP and compared bit-by-bit to the received field.

`0xB8884C0D33D6` (48 bits)

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

ANNEX B OSNMA Test Vectors

A set of test vectors is provided to the user as a complement to testing with the SIS. These test vectors enable the assessment of a receiver capabilities to support different OSNMA configurations as well as the public key and TESLA chain management processes. Test vectors for Merkle tree renewal and the OSNMA alert message are also provided.

The test vectors are provided as attachments to this document and their content is described in this Annex. It shall be noted that the test vectors are provided to assess the correct implementation of the OSNMA protocol logic, and are not representative of the service performance.

The use of the test vectors, following the implementation of [AD.1], [AD.2] and the requirements from section 2 of this document, in particular regarding time synchronisation, shall not lead to any authentication failure.

B.1 Time Synchronisation considerations

In order to use the test vectors, which are set in the past, the receiver is required to set its time (local realisation of GST) to the start time of the test vector, or otherwise the receiver is expected not to process them following the checks specified in section 5.3.1. This implies that a mean to set the receiver time independently of the time synchronisation mechanism implementation resulting from section 2.1 shall be in place to use the test vectors.

The offsetting of the receiver time for testing purpose can also be used to verify the correct implementation of the GST sub-frame verification (section 5.3.1). An offset between the input data time and receiver time larger than the requirement shall lead to OSNMA data not being processed.

It is to be noted that receiver time offsetting shall only be performed as part of testing activities related to the validation of the OSNMA implementation, as it by-passes the time synchronisation requirement implementation set in section 2.1. Time synchronisation to GST is always required to process Galileo SIS.

B.2 Format of the Test Vectors

Eighteen test vectors are provided in attachment to this document, in the folder ‘osnma_test_vectors’. The structure of this folder is provided below.

```
+---configuration_1
|       16_AUG_2023_GST_05_00_01.csv
|
+---configuration_2
|       27_JUL_2023_GST_00_00_01.csv
|
+---crev_step1
|       06_OCT_2023_GST_21_45_01.csv
|
+---crev_step2
|       06_OCT_2023_GST_23_30_01.csv
|
+---crev_step3
```


© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

```
|       07_OCT_2023_GST_00_30_01.csv
|
+---eoc_step1
|       06_OCT_2023_GST_16_45_01.csv
|
+---eoc_step2
|       06_OCT_2023_GST_18_30_01.csv
|
+---nmt_step1
|       07_OCT_2023_GST_12_45_01.csv
|
+---nmt_step2
|       07_OCT_2023_GST_13_45_01.csv
|
+---nmt_step3
|       07_OCT_2023_GST_14_45_01.csv
|
+---npk_step1
|       07_OCT_2023_GST_02_45_01.csv
|
+---npk_step2
|       07_OCT_2023_GST_03_45_01.csv
|
+---npk_step3
|       07_OCT_2023_GST_04_45_01.csv
|
+---oam_step1
|       07_OCT_2023_GST_18_45_01.csv
|
+---oam_step2
|       07_OCT_2023_GST_19_45_01.csv
|
+---pkrev_step1
|       07_OCT_2023_GST_07_45_01.csv
|
+---pkrev_step2
|       07_OCT_2023_GST_09_30_01.csv
```


© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

```
|
\---pkrev_step3
        07_OCT_2023_GST_10_30_01.csv
```

The test vectors are provided in comma-separated value (.csv) format. The name of the file indicates the date and GST time at the start of transmission of the first bit of I/NAV data stream reported in the file. Note that the time reported reflects the fact that the I/NAV sub-frame transmitted in E1-B starts at T_0 + 1 sec, where T_0 is synchronised with the GST origin modulo 30 seconds, as per [AD.1].

Each file contains a one-line header, describing the structure of the data, followed by the data:

- The first column indicates the satellite ID for the satellite transmitting data, as per [AD.1], thus the file contains one line per satellite;
- The second column indicates the number of navigation bits contained in the reported I/NAV stream;
- The third column contains the I/NAV data stream in hexadecimal format (240 bits per page, 60 hexadecimal symbols).

In addition to the test vectors, the cryptographic material required to perform the verifications is also provided, in the formats described in [AD.4], in the folder 'cryptographic_material'. Different sub-folders are available for the three Merkle trees needed to verify the test vectors, as described below.

The tree root and the public key are provided together with the associated certificates[5]. The PublicKey sub folder of the Merkle_tree_1 folder contains all the PKI certificates that allows to test the full chain of trust as described in [AD.4]. The Merkle_tree_2 and Merkle_tree_3 folders provide the cryptographic material and associated certificates that can be found at the GSC web portal interface (see [AD.4] for further details). The certificates provided in these folders allow the verification of the Merkle Tree and Public Key products only up to the Issuing Certificate Authority (ICA) level described in [AD.4].

```
+---Merkle_tree_1
|   +---MerkleTree
|   |       OSNMA_MerkleTree_20230803105953_newPKID_1.xml
|   |       OSNMA_MerkleTree_20230803105953_newPKID_1.xml.md5
|   |
|   \---PublicKey
|           euspa_root_ca_test_phase.crt
|           euspa_root_ca_test_phase.crl
|           euspa_galileo_sca_test_phase.crt
|           euspa_galileo_sca_test_phase.crl
|           ica.crt
|           OSNMA_PublicKeyCRL_20230803105952_newPKID_1.crl
|           OSNMA_PublicKeyCRL_20230803105952_newPKID_1.crl.md5
|           OSNMA_PublicKey_20230803105952_newPKID_1.crt
|           OSNMA_PublicKey_20230803105952_newPKID_1.crt.md5
|           OSNMA_PublicKey_20230803105953_newPKID_1.xml
```

[5] Note that the URLs provided in the CRL Distribution Points field of the test certificates do not allow the automatic verification of the CRL. Also, CA issuers URLs (Authority Information Access field) are not provided in the test certificates, so the CA certificates cannot be automatically downloaded.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

```
|           OSNMA_PublicKey_20230803105953_newPKID_1.xml.md5
|
\---Merkle_tree_2
|   +---MerkleTree
|   |       OSNMA_MerkleTree_20230720113300_newPKID_2.crt
|   |       OSNMA_MerkleTree_20230720113300_newPKID_2.crt.md5
|   |       OSNMA_MerkleTree_20230720113300_newPKID_2.xml
|   |       OSNMA_MerkleTree_20230720113300_newPKID_2.xml.md5
|   |       OSNMA_MerkleTree_20230720113300_newPKID_2.xml.p256
|   |       OSNMA_MerkleTree_20230720113300_newPKID_2.xml.p256.md5
|   |       OSNMA_MerkleTree_20231007041500_PKID_7.crt
|   |       OSNMA_MerkleTree_20231007041500_PKID_7.crt.md5
|   |       OSNMA_MerkleTree_20231007041500_PKID_7.xml
|   |       OSNMA_MerkleTree_20231007041500_PKID_7.xml.md5
|   |       OSNMA_MerkleTree_20231007041500_PKID_7.xml.p256
|   |       OSNMA_MerkleTree_20231007041500_PKID_7.xml.p256.md5
|   |       OSNMA_MerkleTree_20231007081500_PKID_8.crt
|   |       OSNMA_MerkleTree_20231007081500_PKID_8.crt.md5
|   |       OSNMA_MerkleTree_20231007081500_PKID_8.xml
|   |       OSNMA_MerkleTree_20231007081500_PKID_8.xml.md5
|   |       OSNMA_MerkleTree_20231007081500_PKID_8.xml.p256
|   |       OSNMA_MerkleTree_20231007081500_PKID_8.xml.p256.md5
|   |       OSNMA_MerkleTree_20231007141500_PKID_9.crt
|   |       OSNMA_MerkleTree_20231007141500_PKID_9.crt.md5
|   |       OSNMA_MerkleTree_20231007141500_PKID_9.xml
|   |       OSNMA_MerkleTree_20231007141500_PKID_9.xml.md5
|   |       OSNMA_MerkleTree_20231007141500_PKID_9.xml.p256
|   |       OSNMA_MerkleTree_20231007141500_PKID_9.xml.p256.md5
|   |
|   \---PublicKey
|           OSNMA_PublicKeyCRL_20230720113300_newPKID_2.crl
|           OSNMA_PublicKeyCRL_20230720113300_newPKID_2.crl.md5
|           OSNMA_PublicKeyCRL_20231007041500_PKID_7.crl
|           OSNMA_PublicKeyCRL_20231007041500_PKID_7.crl.md5
|           OSNMA_PublicKeyCRL_20231007081500_PKID_8.crl
|           OSNMA_PublicKeyCRL_20231007081500_PKID_8.crl.md5
|           OSNMA_PublicKeyCRL_20231007141500_PKID_9.crl
|           OSNMA_PublicKeyCRL_20231007141500_PKID_9.crl.md5
```


© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

```
|           OSNMA_PublicKey_20230720113300_newPKID_2.crt
|           OSNMA_PublicKey_20230720113300_newPKID_2.crt.md5
|           OSNMA_PublicKey_20230720113300_newPKID_2.xml
|           OSNMA_PublicKey_20230720113300_newPKID_2.xml.md5
|           OSNMA_PublicKey_20231007041500_PKID_7.crt
|           OSNMA_PublicKey_20231007041500_PKID_7.crt.md5
|           OSNMA_PublicKey_20231007041500_PKID_7.xml
|           OSNMA_PublicKey_20231007041500_PKID_7.xml.md5
|           OSNMA_PublicKey_20231007081500_PKID_8.crt
|           OSNMA_PublicKey_20231007081500_PKID_8.crt.md5
|           OSNMA_PublicKey_20231007081500_PKID_8.xml
|           OSNMA_PublicKey_20231007081500_PKID_8.xml.md5
|           OSNMA_PublicKey_20231007141500_PKID_9.crt
|           OSNMA_PublicKey_20231007141500_PKID_9.crt.md5
|           OSNMA_PublicKey_20231007141500_PKID_9.xml
|           OSNMA_PublicKey_20231007141500_PKID_9.xml.md5
|
\---Merkle_tree_3
    +---MerkleTree
    |       OSNMA_MerkleTree_20231007201500_PKID_1.crt
    |       OSNMA_MerkleTree_20231007201500_PKID_1.crt.md5
    |       OSNMA_MerkleTree_20231007201500_PKID_1.xml
    |       OSNMA_MerkleTree_20231007201500_PKID_1.xml.md5
    |       OSNMA_MerkleTree_20231007201500_PKID_1.xml.p256
    |       OSNMA_MerkleTree_20231007201500_PKID_1.xml.p256.md5
    |
    \---PublicKey
            OSNMA_PublicKeyCRL_20231007201500_PKID_1.crl
            OSNMA_PublicKeyCRL_20231007201500_PKID_1.crl.md5
            OSNMA_PublicKey_20231007201500_PKID_1.crt
            OSNMA_PublicKey_20231007201500_PKID_1.crt.md5
            OSNMA_PublicKey_20231007201500_PKID_1.xml
            OSNMA_PublicKey_20231007201500_PKID_1.xml.md5
```


© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

The following table summarises the cryptographic material to be used with the different test vectors.

Table 9. Cryptographic material

Test Vector	Merkle Tree	NMT	PKID	NPKID
Configuration 1	1	N/A	1	N/A
Configuration 2	2	N/A	2	N/A
End of Chain Step 1	2	N/A	7	N/A
End of Chain Step 2	2	N/A	7	N/A
Chain Revocation Step 1	2	N/A	7	N/A
Chain Revocation Step 2	2	N/A	7	N/A
Chain Revocation Step 3	2	N/A	7	N/A
New Public Key Step 1	2	N/A	7	N/A
New Public Key Step 2	2	N/A	7	8
New Public Key Step 3	2	N/A	8	N/A
Public Key Revocation Step 1	2	N/A	8	9
Public Key Revocation Step 2	2	N/A	9	N/A
Public Key Revocation Step 3	2	N/A	9	N/A
Merkle Tree Renewal Step 1	2	3	9	N/A
Merkle Tree Renewal Step 2	2	3	9	1
Merkle Tree Renewal Step 3	3	N/A	1	N/A
OSNMA Alert Message Step 1	3	N/A	1	N/A
OSNMA Alert Message Step 2	3	N/A	1	N/A

The public key curves to be used for the verification of the digital signatures is summarised in the table below.

Table 10. PK curves

Merkle Tree	PKID	Public Key Curve
1	1	ECDSA P-256/SHA-256
2	2	ECDSA P-256/SHA-256
2	7	ECDSA P-256/SHA-256
2	8	ECDSA P-256/SHA-256
2	9	ECDSA P-521/SHA-512
3	1	ECDSA P-256/SHA-256

B.3 Test Vectors for Different OSNMA Configurations

The two test vectors provided correspond to different OSNMA configurations. These test vectors enable the assessment of a receiver capabilities to support different OSNMA configurations.

B.3.1 Configuration 1

The OSNMA parameters relative to Configuration 1 are summarised in **Table 11**.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

Table 11. OSNMA Configuration 1

Parameters	Settings
Tag size	40 bits
Key size	128 bits
Number of tags per MACK message (n_t)	6
Digital Signature algorithm	ECDSA P-256
Nb. of blocks in the DSM-HKROOT (NB_{DK})	8
Hash Function	SHA-256
MAC Function	HMAC-SHA-256
MACLT ID	33
MACLT Sequence	[00S, 00E, 04S, 00E, 12S, 00E]; [00S, 00E, 00E, 12S, 00E, 12E]
Merkle Tree	1
Public Key ID	1

The test vector corresponds to a scenario where the NMAS is set to “Test” and the CPKS flag to “Nominal”. It starts at 05:00:01 GST on the 16-08-2023 and has a duration of one hour. Note that the test vector includes some dummy tags transmitted by E10, E11, E12 and E31 at 05:07:30 GST, and that can be successfully verified as per [AD.2].

B.3.2 Configuration 2

The OSNMA parameters relative to Configuration 2 are summarised in **Table 12**.

Table 12. OSNMA Configuration 2

Parameters	Settings
Tag size	40 bits
Key size	128 bits
Number of tags per MACK message (n_t)	6
Digital Signature algorithm	ECDSA P-256
Nb. of blocks in the DSM-HKROOT (NB_{DK})	8
Hash Function	SHA-256
MAC Function	HMAC-SHA-256
MACLT ID	34
MACLT Sequence	[00S, FLX, 04S, FLX, 12S, 00E]; [00S, FLX, 00E, 12S, 00E, 12E]
Merkle Tree	2
Public Key ID	2

The test vector corresponds to a scenario where the NMAS is set to “Operational” and the CPKS flag to “Nominal”. It starts at 00:00:01 GST on the 27-07-2022 and has a duration of one hour.

Note that the public key in force is transmitted nominally at the start of the test vector, as defined in [AD.2]. The public key parameters are summarised in **Table 10**.

B.4 Test Vectors for the Renewal and Revocation Processes

B.4.1 Chain Renewal

As explained in [AD.2], the chain renewal is composed of two steps. Two test vectors are provided, that capture the start of each of these steps. These test vectors are independent, requiring the receiver time to be set to the date and time indicated in each file name. However, the successful verification of both is required in order to handle the full chain renewal process.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

B.4.1.1 Step 1

This test vector captures the first step of the chain renewal process, illustrated in **Figure 6**. It starts at 16:45:01 GST on the 06-10-2023 and has a duration of one hour. As per the notation used in [AD.2], the first number in the KROOT indicates the chain ID (CID) to which the KROOT belongs, and the second number indicates the ID of the public key (PKID) to be used for the KROOT verification.

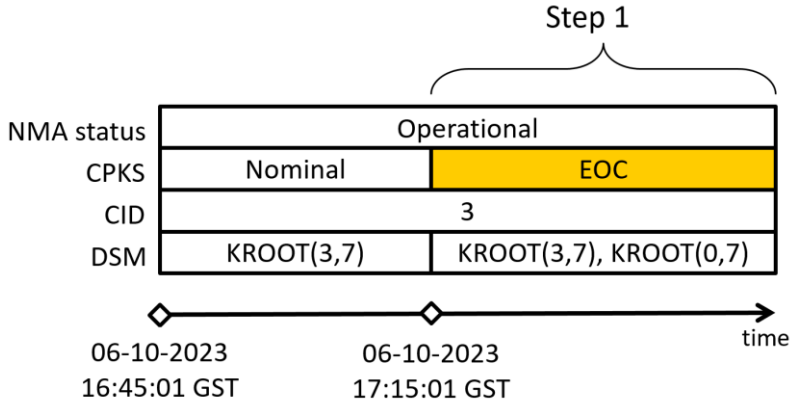


Figure 6. EOC Step 1 test vector

At 17:15:01 GST, the CPKS flag changes from “Nominal” to “End of Chain”, indicating the start of Step 1.

B.4.1.2 Step 2

This test vector captures the transition to Step 2 of the end of chain renewal process, shown in **Figure 7**. It starts at 18:30:01 GST on the 06-10-2023 and has a duration of one hour. At 19:00:01 GST, the CPKS flag reverses from “End of Chain” to “Nominal”, indicating the start of Step 2.

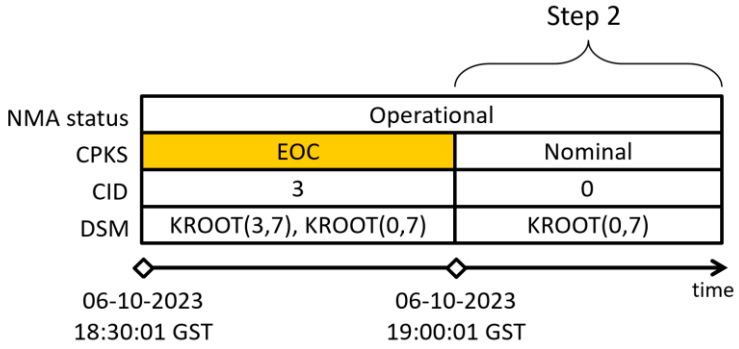


Figure 7. EOC Step 2 test vector

Note that the tags retrieved at the end of Step 1, before the chain transition, are generated with the first TESLA key of the new chain and can be successfully verified with this key.

B4.2 Chain Revocation

The chain revocation process is composed of three steps, the start of each of these steps is captured in three test vectors. They are described in the following sections. These test vectors are independent and the receiver time has to be set for each of them according to the time and date indicated in the file name. However, the successful verification of all three vectors is required in order to handle the full chain revocation process.

B.4.2.1 Step 1

This test vector captures the first step of the chain revocation process, illustrated in **Figure 8**. It starts at 21:45:01 GST on the 06-10-2023 and has a duration of one hour. At 22:15:01 GST, the NMA Status changes from “Operational” to “Don’t Use” and the CPKS flag from “Nominal” to “Chain Revoked”, indicating the start of Step 1.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

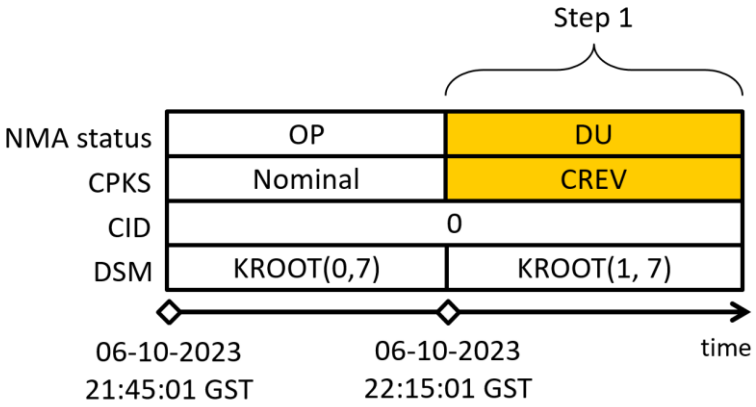


Figure 8. CREV Step 1 test vector

B.4.2.2 Step 2

This test vector captures the second step of the chain revocation, shown in **Figure 9**. It starts at 23:30:01 GST on the 06-10-2023 and has a duration of one hour.

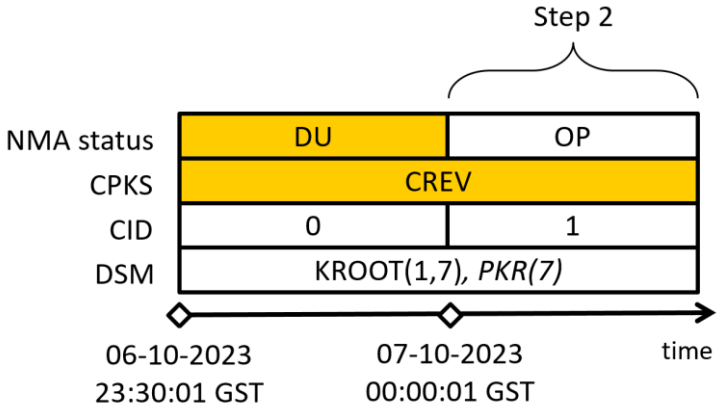


Figure 9. CREV Step 2 test vector

At 00:00:01 GST on the 07-10-2023, the NMA Status reverses from “Don’t Use” to “Operational” while the CPKS flag remains set to “Chain Revoked” indicating the start of Step 2. The new TESLA chain (CID = 1), which parameters are transmitted in the DSM-KROOT, enters into force and navigation data authentication can be resumed.
It should be noted that, as in the case of the TESLA chain renewal, the last tags transmitted before the start of Step 2 can be verified with the keys belonging to the new chain. Additionally, as per [AD.2], the applicable public key (PKID = 7) is transmitted nominally at 00:00:01 GST.

B.4.2.3 Step 3

This test vector captures the third step of the chain revocation, shown in **Figure 10**. It starts at 00:30:01 GST on the 07-10-2023 and has a duration of one hour.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

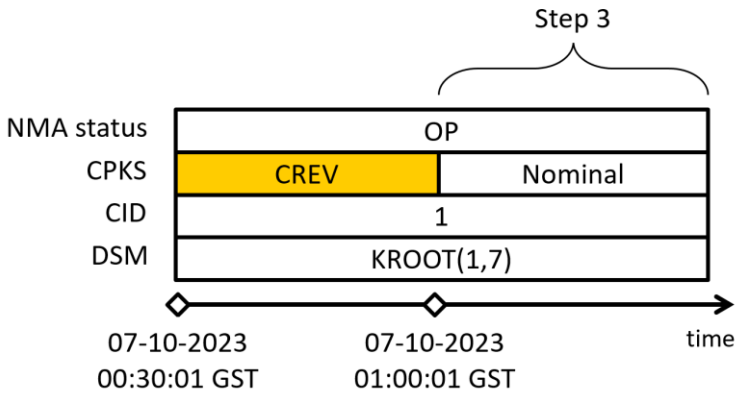


Figure 10. CREV Step 3 test vector

At 01:00:01 GST, with the NMA Status set “Operational”, the CPKS flag changes from “Chain Revoked” back to “Nominal”, indicating the start of Step 3, i.e. the end of the chain revocation process.

B.4.3 Public Key Renewal

The public key renewal process is composed of three steps, three test vectors are provided capturing the start of each of these steps. They are described in the following sections. These three test vectors are independent and the receiver time has to be set for each of them according to the time and date indicated in the file name. However, the successful verification of all three vectors is required in order to handle the full public key renewal process.

B.4.3.1 Step 1

This test vector captures the first step of the Public Key Renewal, shown in **Figure 11**. It starts at 02:45:01 GST on the 07-10-2023 and has a duration of one hour. At 03:15:01 GST, the CPKS flag changes from “Nominal” to “New Public Key", indicating the start of Step 1. As per the notation used in [AD.2], the number in the PKR indicates the ID of the public key (PKID) being transmitted.

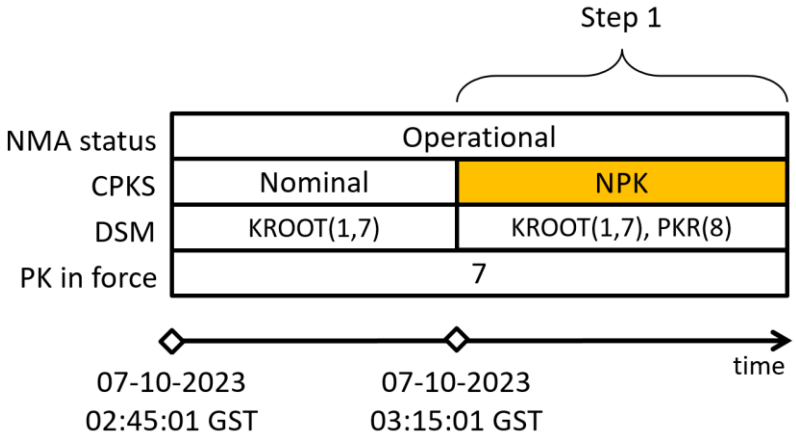


Figure 11. NPK Step 1 test vector

The DSM-KROOT for the chain in force (KROOT(1,7)) is transmitted in alternation with a DSM-PKR for the new public key (PKR(8)).

B.4.3.2 Step 2

This test vector captures the second step of the Public Key Renewal, shown in **Figure 12**. It starts at 03:45:01 GST on the 07-10-2023 and has a duration of one hour.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

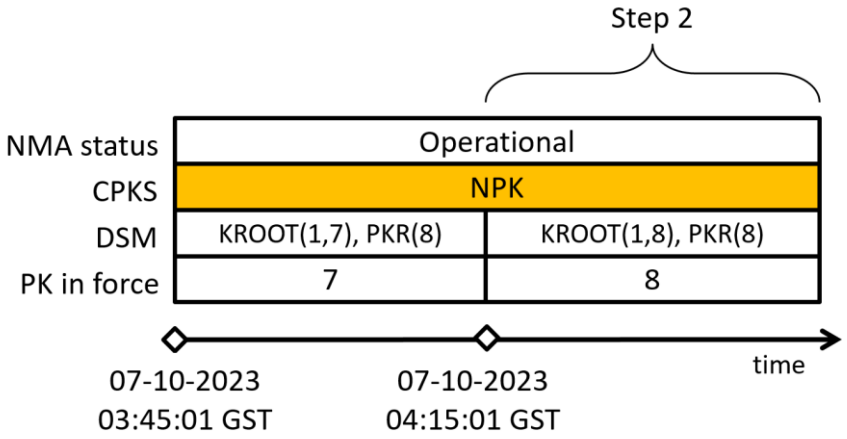


Figure 12. NPK Step 2 test vector

At 04:15:01 GST, the new public key (PKID = 8) enters into force with the transmission of a new DSM-KROOT verified with PKID = 8, indicating the start of Step 2.

B.4.3.3 Step 3

This test vector captures the third step of the public key renewal, shown in **Figure 13**. It starts at 04:45:01 GST on the 07-10-2023 and has a duration of one hour.

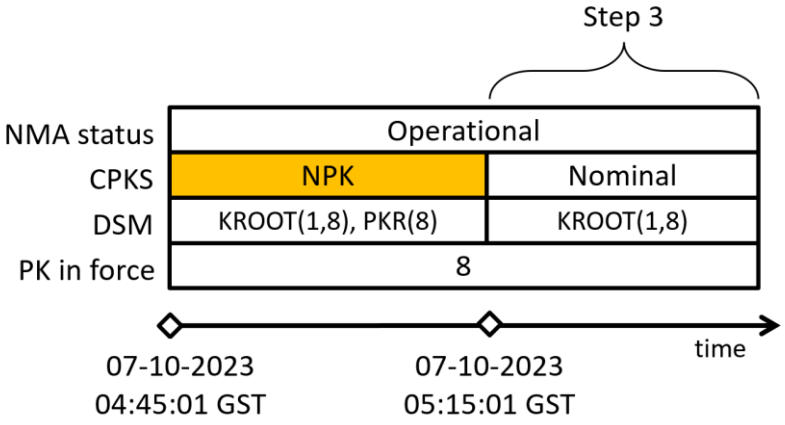


Figure 13. NPK Step 3 test vector

At 05:15:01 GST, the CPKS flag is reversed from “New Public Key” to “Nominal” and only the DSM-KROOT is transmitted, indicating the start of Step 3.

B.4.4 Public Key Revocation

The public key revocation process is also composed of three steps. The three test vectors that capture the start of each of these steps are described in the following sections. As for the previous sections, these three test vectors are independent and the receiver time has to be set for each of them according to the time and date indicated in the file name. However, the successful verification of all three vectors is required in order to handle the full public key revocation process.

B.4.1.1 Step 1

This test vector captures the first step of the public key revocation, illustrated in **Figure 14**. It starts at 07:45:01 GST on the 07-10-2023 and has a duration of one hour.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

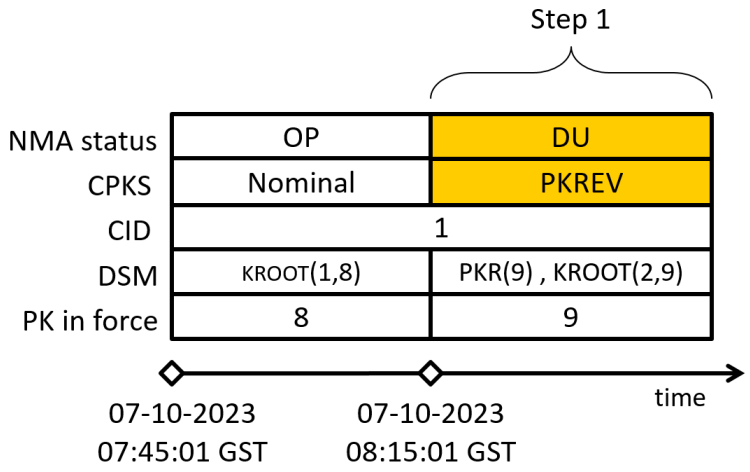


Figure 14. PKREV Step 1 test vector

At 08:15:01 GST, the NMA Status changes from “Operational” to “Don’t Use” and the CPKS flag from “Nominal” to “Public Key Revoked", indicating the start of Step 1.

B.4.1.2 Step 2

This test vector captures the second step of the public key revocation, shown in **Figure 15**. It starts at 09:30:01 GST on the 07-10-2023 and has a duration of one hour.

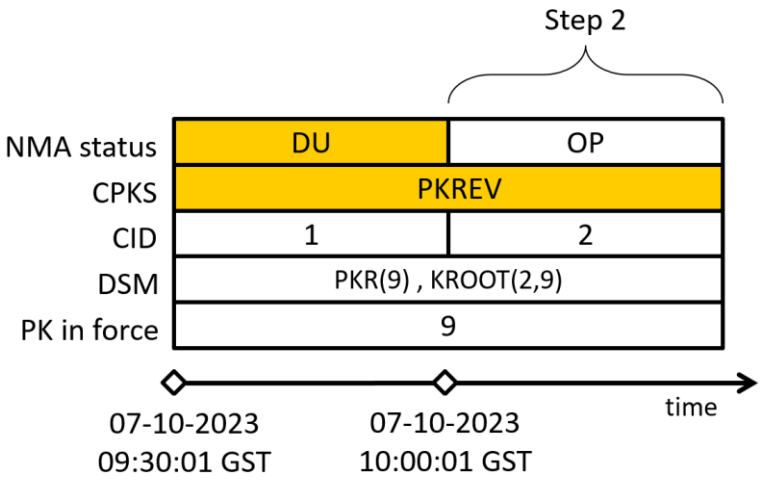


Figure 15. PKREV Step 2 test vector

At 10:00:01 GST, the NMA Status reverses from “Don’t Use” to “Operational” while the CPKS flag remains set to “Public Key Revoked", indicating the start of Step 2. The new TESLA chain (CID = 2), which parameters are transmitted in the DSM-KROOT, enters into force and navigation data authentication can be resumed.

As for the TESLA chain renewal, the last tags transmitted at the end of Step 1 can be verified the keys from the new chain.

B.4.1.3 Step 3

This test vector captures the third step of the public key revocation, shown in **Figure 16**. It starts at 10:30:01 GST on the 07-10-2023 and has a duration of one hour.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii
OSNMA Receiver Guidelines, Issue 1.3, January 2024

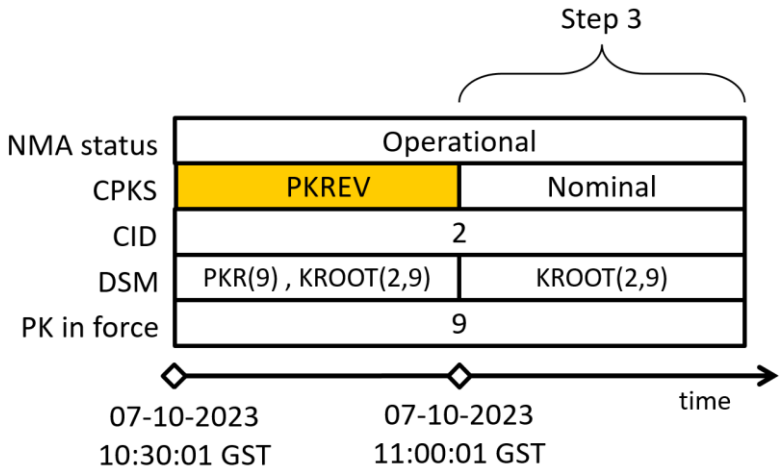


Figure 16. PKREV Step 3 test vector

At 11:00:01 GST, with the NMA Status set “Operational”, the CPKS flag changes from “Public Key Revoked" back to “Nominal”, indicating the start of Step 3.

B.4.5 Merkle Tree Renewal

The Merkle tree renewal process is composed of three steps and three test vectors are provided, capturing the start of each of these steps. They are described in the following sections. These three test vectors are independent and the receiver time has to be set for each of them according to the time and date indicated in the file name. However, the successful verification of all three vectors is required in order to handle the full Merkle tree renewal process.

B.4.5.1 Step 1

This test vector captures the first step of the Merkle tree renewal process, illustrated in **Figure 17**. It starts at 12:45:01 GST on the 07-10-2023 and has a duration of one hour.

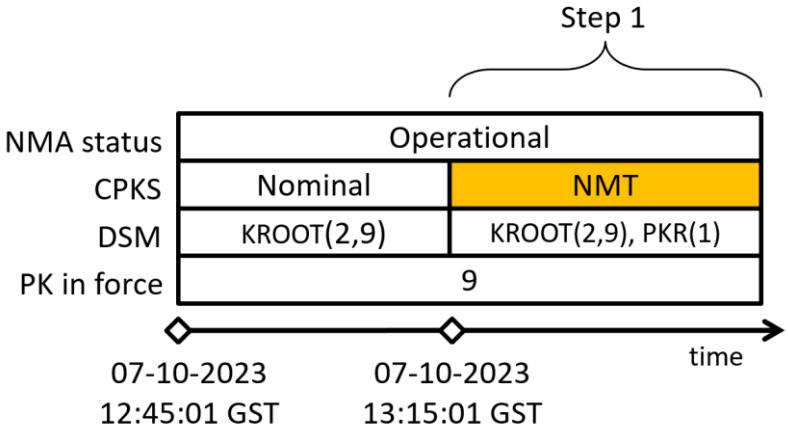


Figure 17. NMT Step 1 test vector

At 13:15:01 GST, the CPKS flag from “Nominal” to “New Merkle Tree", indicating the start of Step 1. The future public key (PKID = 1) associated with the future Merkle tree is alternated with the KROOT of the chain in force (CID = 9). The future Merkle tree, which will be retrieved from the GSC OSNMA Server during service provision, can be found in the cryptographic material folder.

B.4.5.2 Step 2

This test vector captures the second step of the Merkle tree renewal, shown in **Figure 18**. It starts at 13:45:01 GST on the 07-10-2023 and has a duration of one hour.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

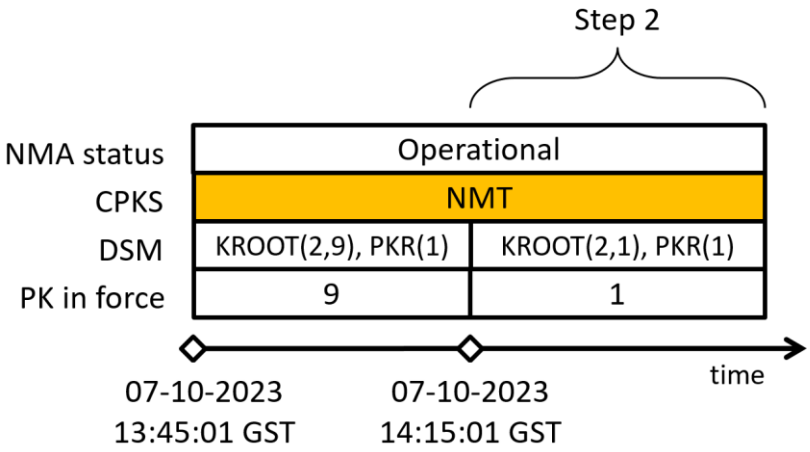


Figure 18. NMT Step 2 test vector

At 14:15:01 GST, while the CPKS flag remains set to “Public Key Revoked", a KROOT verified with the new public key (PKID = 1) is transmitted, indicating the entry into force of the new Merkle tree and the start of Step 2. Note that the chain is not renewed.

B.4.5.3 Step 3

This test vector captures the third step of the Merkle tree renewal, shown in **Figure 19**. It starts at 14:45:01 GST on the 07-10-2023 and has a duration of one hour.

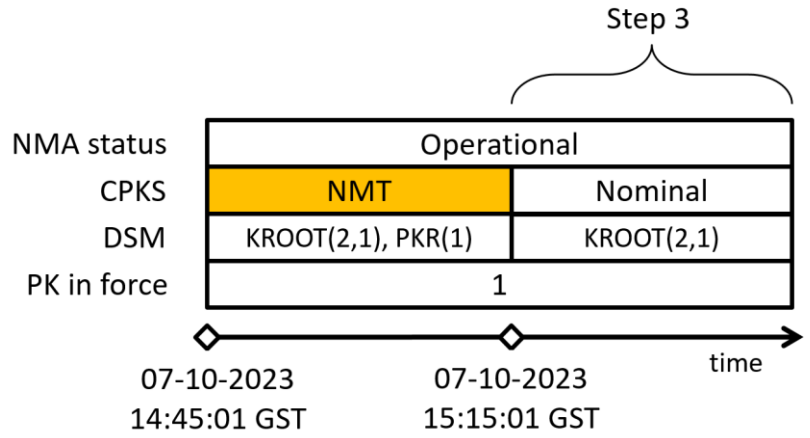


Figure 19. NMT Step 3 test vector

At 15:15:01 GST, the CPKS flag changes from “New Merkle Tree" back to “Nominal”, indicating the start of Step 3.

B.4.6 OSNMA Alert Message

The transmission of the OSNMA Alert Message is composed of two steps and two test vectors are provided, which capture the start of each of these steps. They are described in the following sections. These two test vectors are independent and the receiver time has to be set for each of them according to the time and date indicated in the file name. However, the successful verification of both vectors is required in order to handle the full OAM process.

B.4.6.1 Step 1

This test vector captures the first step of the OSNMA Alert Message transmission, illustrated in **Figure 20**. It starts at 18:45:01 GST on the 07-10-2023 and has a duration of one hour.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

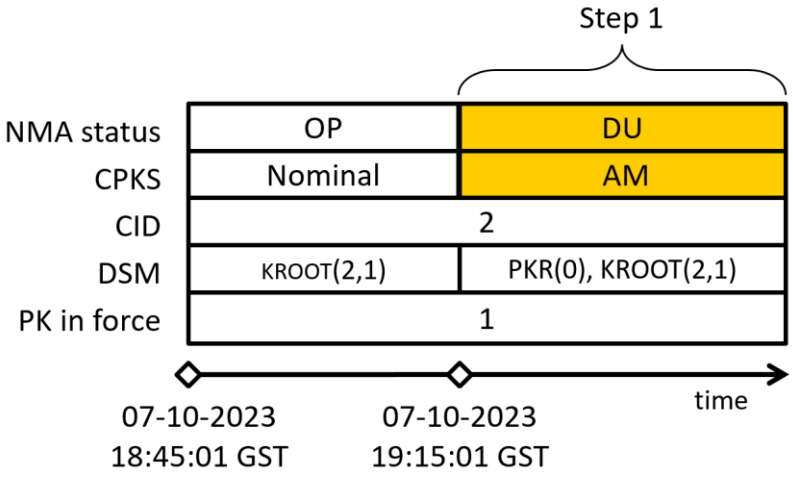


Figure 20. OAM Step 1 test vector

At 19:15:01 GST, the NMA Status changes from “Operational” to “Don’t Use” and the CPKS flag from “Nominal” to “Alert Message", indicating the start of Step 1. The alert message is transmitted in the PKR(0), with an NPKT = 4, and is verified as per [AD.2].

B.4.6.2 Step 2

This test vector captures the second step of the OSNMA Alert Message transmission, shown in **Figure 21**. It starts at 19:45:01 GST on the 07-10-2023 and has a duration of one hour.

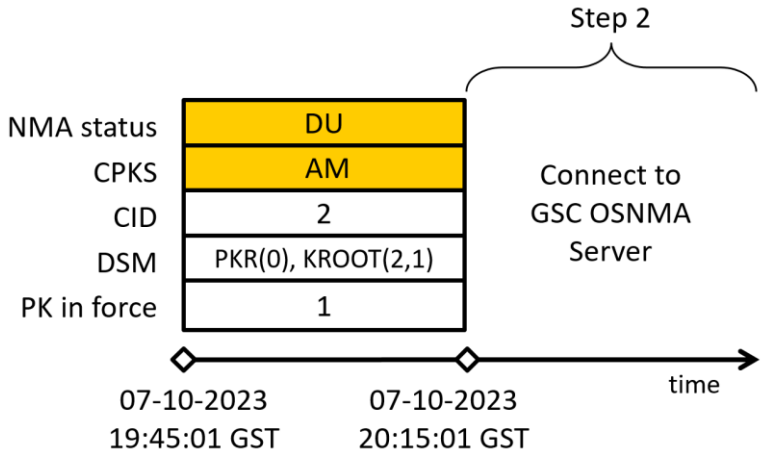


Figure 21. OAM Step 2 test vector

At 20:15:01 GST, the transmission of OSNMA is stopped. During service provision, the user will be expected to connect to the GSC OSNMA server.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

ANNEX C Receiver Initial Conditions and Fulfilment of the Time Synchronisation Requirement

As explained in section 2.1, the security of the OSNMA protocol depends on the fulfilment by the receiver of the time synchronisation requirement. This requirement can be expressed as function of the receiver time synchronisation uncertainty *B*, with respect to Galileo System Time, as follows:

$$B < \frac{\mathrm{T_L}}{2} \qquad \textbf{Eq. 6}$$

B is defined such that:

$$t_i^{\mathrm{GST}} \in [t_i^{Rx} - B; t_i^{Rx} + B] \qquad \textbf{Eq. 7}$$

Where t_i^{Rx} is the time of event i as reported in the receiver local time and t_i^{GST} is the time of event i as reported in Galileo System Time.

A receiver with a synchronisation uncertainty with respect to GST,B, such that $B < \frac{T_L}{2} + 150\ sec$, can process slow MAC with a 10 sub-frame delay (ADKD12 from [AD.2]).

If none of the above conditions on the receiver time synchronisation uncertainty B is verified, the OSNMA protocol shall not be used.

At start up, the receiver may be in one of two states:

- It has a knowledge of its time synchronisation uncertainty B,
- It has no time information.

These two cases are discussed in the following sections.

C.1 Known Time Synchronisation Uncertainty

The first case corresponds, for example, to a receiver with an internal Real Time Clock (RTC) running on battery when the RF section and main GNSS core are powered down. At the switch on, the receiver must get information on its time synchronisation uncertainty *B*, which may be estimated based on the stability of the clock used. The receiver can then proceed as described below:

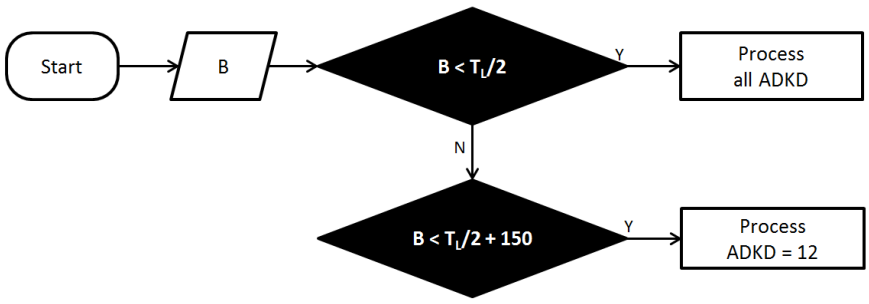


Figure 22. ADKD processing as a function of the time synchronisation uncertainty

If the time synchronisation requirement is fulfilled ($B<T_L/2$), the receiver can process all tags, regardless of their ADKD types. If the receiver does not fulfil the requirement, it may still be able to exploit the slow MACs, corresponding to ADKD = 12, as illustrated in Figure 22.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

To be noted that the timing solution computed with the authenticated data is not itself authenticated. Therefore, if this time solution is used to re-synchronise the clock (i.e. to reduce *B*), an associated risk shall be taken into account.

C2. Unknown Time Synchronisation Uncertainty

If the receiver has no time information, an estimation of the GST and its associated uncertainty shall be retrieved. Several strategies are possible, for example:

- An external clock can be used;
- For connected users, a secure source can be exploited, e.g., a secure network connection with a time transfer capability smaller than the synchronization requirement. This condition is achievable by conventional approaches, e.g., a secured implementation of the Network Time Protocol (NTP) can usually maintain time to within tens of milliseconds over the public internet, achieving better than one millisecond accuracy in local area networks under ideal conditions.

Care should be taken to estimate the Galileo System Time and therefore to perform any conversion, if required, taking into account any associated uncertainty.

As stated in section 2.1, the security of the protocol relies on the correct synchronisation of the receiver with the system time. Thus, considerations about the security and associated risks of accessing the timing information shall be taken into account.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

ANNEX D Increasing Resilience of Receivers to Spoofing Attacks and the role of OSNMA

OSNMA provides means to authenticate navigation data, which can then be used to compute a PVT. It should be noted though, that as this PVT is computed using non-verified ranging information, it cannot be considered authenticated. Thus, users can obtain a more robust PVT solution by combining OSNMA with additional checks in the receiver. It is to be noted that the insertion of OSNMA data within I/NAV increases the unpredictability of I/NAV data stream itself. In particular, all tags are unpredictable by definition and, for each sub-frame, the earliest-received key is also unpredictable. The receiver can use the unpredictability of the I/NAV symbols encoding OSNMA to make the signals more robust against replay attacks. In addition, some examples of receiver consistency checks are provided below.

At signal processing level:

- Search and detection of vestigial signals around possibly false tracked signal can be carried out;
- Tracking control loop parametrisation and correlation function can be monitored.

At measurement level:

- Consistency checks between the Automatic Gain Control (ACG) and Carrier-to-Noise (C/N_0) values can be performed to detect abnormal power emissions;
- Measurements can be monitored over time to detect abrupt changes;
- Receiver autonomous integrity monitoring (RAIM) techniques, including the detection of measurements inconsistent with the estimated position, can be implemented.

At PVT level:

- The position and velocity solutions can be monitored over time to detect abnormal values, sudden changes and trajectories and dynamics that are not consistent with the vehicle's dynamics;
- The receiver can monitor the time solution over time and cross-check it against the performance of the internal clock.

Beyond the GNSS receiver itself:

- Anti-tampering measures can be put in place to prevent the manipulation of the GNSS receiver and antenna;
- Additional sensors (e.g. odometers, inertial sensors) can be used to perform consistency checks;
- Multiple antennas enable the receiver to detect the presence of counterfeit signal from the direction of arrival of the signals.

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

ANNEX E Changes between the OSNMA Receiver Guidelines and the OSNMA Receiver Guidelines for the Test Phase[6]

This annex is meant to support users who developed prototypes to identify the changes between the elements of the OSNMA Receiver Guidelines for the Test Phase and those presented in these Receiver Guidelines.

OSNMA Receiver Guidelines for the Test Phase, Issue 1.1	OSNMA Receiver Guidelines, Issue 1.0	Description
3.1	3.1	Additional information of the Merkle tree renewal process is provided, in particular regarding the possibility to store the future Merkle tree root before it becomes applicable.
3.2	3.2	The Public Key retrieval strategy is updated to take into account the nominal provision of DSM-PKR at defined time intervals.
3.3	3.3	The TESLA root key retrieval strategy is updated so that the user may retrieve different elements of the protocol in parallel in order to optimise its implementation.
4.1.1.1	4.1.1.1	The description of the OSNMA cold start is updated to take into account the possibility that the Merkle tree root has been renewed.
4.2.2	4.2.2	The section is updated to include the description of the operations the user shall carry out in case the new Merkle tree (NMT) flag is received.
4.2.3	4.2.3	The section is updated to include the description of the operations the user shall carry out in case the OSNMA Alert Message (OAM) flag is received.
5.5.5	5.5.5	The tag verification is updated to include considerations about the verification of the dummy tags (as indicated by COP = 0).
5.5.6	5.5.6	The tag accumulation target is removed from the guidelines; its value will be provided in the future Galileo Authentication Service Definition Document. The tag accumulation principle is updated so that users can exploit the new COP parameter to identify tags that can be accumulated with each other.

[6] European Commission, Galileo Open Service Navigation Authentication (OSNMA) Receiver Guidelines for the Test Phase, Issue 1.1, October 2022

© European Union 2024
Document subject to terms of use and disclaimers p. i-ii

LINKING SPACE TO USER NEEDS

www.euspa.europa.eu

@EU4Space

@EU4Space

EUSPA

@space4eu

EUSPA

#EUSpace