Improving Galileo OSNMA Time To First Authenticated Fix

Aleix Galan, Ignacio Fernandez-Hernandez, Wim De Wilde, Sofie Pollin, Gonzalo Seco-Granados

***Abstract*—Galileo is the first global navigation satellite system to authenticate their civilian signals through the Open Service Galileo Message Authentication (OSNMA) protocol. However, OSNMA adds a delay in the time to obtain a first position and time fix, the so-called Time To First Authentication Fix (TTFAF). Reducing the TTFAF as much as possible is crucial to integrate the technology seamlessly into the current products. In the cases where the receiver already has cryptographic data available, the so-called *hot start* mode and focus of this article, the currently available implementations achieve an average TTFAF of around 100 seconds in ideal environments. In this work, we dissect the TTFAF process, propose two main optimizations to reduce the TTFAF, and benchmark these optimizations in three distinct scenarios (open-sky, soft urban, and hard urban) with recorded real data. Moreover, we also evaluate the optimizations using the synthetic scenario from the official OSNMA test vectors. The first block of optimizations centers on extracting as much information as possible from broken sub-frames by processing them at page level and combining redundant data from multiple satellites. The second block of optimizations aims to reconstruct missed navigation data by the intelligent use of fields in the authentication tags belonging to the same sub-frame as the authentication key. Combining both optimization ideas improves the TTFAF substantially for all considered scenarios. We obtain an average TTFAF of 60.9 and 68.8 seconds for the test vectors and the open-sky scenario, respectively, with a best-case of 44.0 seconds in both. Likewise, the urban scenarios see a drastic reduction of the average TTFAF between the non-optimized and optimized cases, from 127.5 to 87.5 seconds in the soft urban scenario and from 266.1 to 146.1 seconds in the hard urban scenario. These optimizations have been made available as part of the open-source OSNMAlib library on GitHub.**

***Index Terms*—Global navigation satellite system, Galileo, OSNMA, Authentication, TTFAF optimization, OSNMAlib**

I. INTRODUCTION

GLOBAL Navigation Satellite Systems (GNSS) signals are vulnerable to interference, including the transmission of false GNSS-like signals, or spoofing. Adding cryptographic information to civil GNSS signals was proposed decades ago as a way to detect spoofing [1], but it has taken time until its implementation. Meanwhile, several receiver-based anti-spoofing methods such as signal power monitoring [2] or inertial systems [3] have been proposed. Finally, GNSS signals are gradually starting to provide cryptographic information.

This research was partially funded by the Research Foundation Flanders (FWO) Frank de Winne PhD Fellowship, project number 1SH9424N (Aleix Galan).

A. Galan, I. Fernandez-Hernandez and S. Pollin are with KU Leuven. I. Fernandez-Hernandez is also with the European Commission. Wim De Wilde is with Septentrio NV. G. Seco-Granados is with UAB.

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Cryptographic techniques exploit the spoofer's ignorance of the cryptographic material when forging a signal. They can be applied to the spreading codes [4] [5] or to the navigation data bits, which is known as Navigation Message Authentication (NMA). Although in theory a signal can still be replayed [6], NMA facilitate the detection of such attacks [7] and provides very good protection against other common attack methods.

Galileo, the European GNSS, is the first GNSS to provide authentication to its civil signals, and does so implementing their own NMA-based protocol called OSNMA (Open Service Navigation Message Authentication). This protocol is the one used in this paper's research. It was proposed in the last decade [8], has been transmitted over the last years, and is expected to be launched operationally imminently [9].

When adding OSNMA, receivers should not experience a degradation in accuracy or availability [10]. However, the TTFF (Time To First Fix) will be impacted. This is mainly because OSNMA is based on TESLA (Timed Efficient Stream Loss-Tolerant Authentication) [11], a delayed disclosure protocol, adapted to GNSS. Delayed disclosure protocols for GNSS have several advantages, but the main disadvantages are that they require an external loose time reference, and that they purposely delay the disclosure of the cryptographic key required for the authentication verification, which increases the Time To First Authenticated Fix (TTFAF) with respect to TTFF [12]. For Galileo, TTFF has been typically in the order of 30-60 seconds, although some recent improvements (the so-called *I/NAV improvements*) in the navigation message will bring it to even lower values [13].

Specifically, we will focus on *hot start* TTFAF, where the cryptographic information required to bootstrap the receiver is already known. Hereinafter, we will refer to TTFAF as hot start TTFAF. The OSNMA impact in TTFAF has been previously analyzed in the literature. Reference [14] reaches an average TTFAF down to approximately 150 seconds including I/NAV improvements and 170 seconds excluding them. In [10], the best-case comparable to this work achieves 127 seconds. Reference [9] achieves a best-case of 120 seconds, and [15] achieves 90 seconds TTFAF. It is normal that these values vary, as they depend on the receiver implementation, which was not optimized to reduce TTFAF. We believe TTFAF optimization is relevant for potentially many OSNMA users, and is the focus of this paper. We propose several strategies to reduce OSNMA TTFAF down to 44 seconds in the best-case, and test them in different environments.

To implement the proposed optimizations we used OSNMAlib [16], an open-source library that implements the OSNMA protocol which we developed in 2022 and maintained

since then. As the library is written is Python, it is easy to modify and extend for research purposes, even though it might not be suitable for embedded purposes.

OSNMAlib is not a receiver by itself, therefore it needs a GNSS receiver to track the satellites and decode the navigation data bits. For that purpose, we used Septentrio GNSS receivers (mosaic-X5 [17] and PolaRx5TR [18]) to collect all the necessary data, which logging format is already integrated in our library.

The main contributions of this paper can be summarized as follows:

- We propose two ideas to improve the TTFAF: page-level processing and COP-IOD optimization. The first approach is to extract partial information from broken sub-frames. The second idea goes even further and allows the reconstruction of missing navigation data by the innovative use of new OSNMA fields to improve TTFAF significantly.
- We validate these optimizations in three relevant scenarios using real data. The scenarios are diverse (open-sky, soft urban and hard urban) to show that the two proposed methods are very complementary and both ideas are needed to enable robust gains in all scenarios. We also evaluate the ideas using the official OSNMA test vectors.
- We analyze the OSNMA cross-authentication algorithm and the implications it has in the TTFAF when leveraging on the COP-IOD optimization.
- We provide an open-source implementation of the methods described in the paper in the OSNMAlib library.

The paper is organized as follows. Next section provides a general description of the OSNMA protocol and a brief summary of the OSNMAlib library. Then, the hot start TTFAF process and the proposed optimizations are detailed. This is followed by a description of the test scenarios used and, after, the test results are presented and discussed. The paper finalizes with the conclusions and further improvement ideas.

II. GALILEO I/NAV, OSNMA AND OSNMALIB

A. Galileo I/NAV and OSNMA

Galileo OSNMA is transmitted in the I/NAV message, E1-B signal component [19]. The E1-B I/NAV message is composed of 30-second sub-frames of 15 two-second pages, each page including a Word Type (WT). WTs 1 to 5 contain the satellite ephemerides, ionosphere model and health flags, and WTs 6 and 10 include time parameters (the latter shared with almanacs). There are other WTs including only almanacs (WT 7 to 9) and spare words (WT 0). As part of the I/NAV improvements above-mentioned, Galileo has recently added new WTs: WT16 with a reduced ephemerides and WTs 17 to 20 with page recovery through Reed Solomon, which can be useful for OSNMA but we leave outside of our analysis for now. The word type order inside a sub-frame is represented in Fig. 1.

OSNMA is inserted in Galileo's E1-B page in a 40-bit field transmitted therefore every two seconds. As above-mentioned, OSNMA uses the TESLA protocol, with some variations and features such as key chain sharing across transmitting satellites and cross-authentication. The OSNMA 40-bit field is divided into the so-called HKROOT (Header and Root Key) section, of 8 bits, and the MACK (Message Authentication Codes and Key) section, of 32 bits. In this work we focus on the latter, which is the most relevant one for hot-start TTFAF.

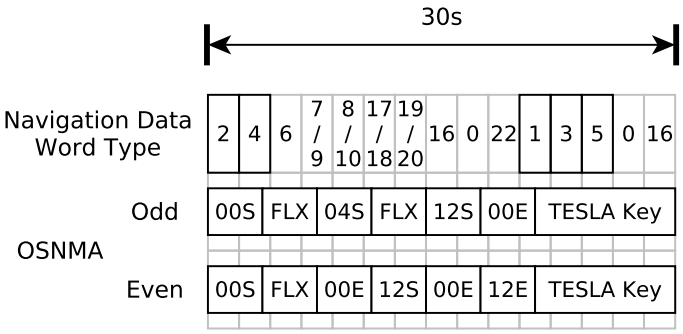


Fig. 1. Operational configuration of Galileo navigation data and OSNMA data for one sub-frame. Some WTs alternate between even and odd sub-frames. The WTs with bold border are used in the ADKD0 authentication. In this representation, the authentication tag size is 40 bits and the TESLA key size is 128 bits.

In the MACK section, six truncated MACs, or *tags*, are transmitted, preceding a key that authenticates the tags in the previous sub-frame (Fig. 1). Each tag has 40 bits and it incorporates a 16-bit *tag-info* section, which encodes the satellite number the tag applies to and the type of authentication. At the moment, there are three types of tags, defined by the so-called ADKD (Authentication Data and Key Delay) parameter. ADKD0 and ADKD12 authenticate WTs 1 to 5, but ADKD12 with a key transmitted five minutes later to relax the receiver loose sync requirement, and ADKD4 authenticates the time (WT 6 and 10).

Due to system limitations, not all satellites can transmit OSNMA data at the same time. We reefer to a satellite transmitting OSNMA as *connected* and a satellite not transmitting OSNMA as *disconnected*. To solve this limitation, OSNMA transmits cross-authentication tags that enable the authentication of disconnected satellites. The ADKD0 cross-authentication tag positions are named `00E` in Fig. 1. There are also flex positions (`FLX`), which tag type is not predefined and needs to be verified at run-time, that are currently only used for ADKD0 cross-authentication tags. Therefore, there are 3 ADKD0 cross-authentication tags on each sub-frame.

For further details, a broad explanation of OSNMA is provided in [20] and the full OSNMA specification can be found in the OSNMA SIS ICD [21] and receiver guidelines [22].

B. OSNMAlib

OSNMAlib [16] [23] is an open-source library written in Python that implements the OSNMA protocol. The library can be integrated into existing receivers and applications to incorporate NMA to the PVT calculation. It can read the Galileo I/NAV pages from an input, store the navigation and authentication data, perform the verification operations, and report the status. The library supports cold start, warm start and hot start procedures.

The input required for OSNMAlib to work is the navigation data bits from Galileo E1-B I/NAV message as nominal page, the Galileo System Time (GST) of the page transmission, and the Satellite Vehicle ID (SVID) to which the navigation data bits belong. Currently, OSNMAlib has the following input modules:

- Septentrio SBF: Post-process files or live data in real-time from a Septentrio receiver in Septentrio Binary Format (SBF) if it contains the GALRawINAV block.
- u-blox UBX: Post-process files or live data in real-time from a u-blox receiver in UBX format if it contains the UBX-RXM-SFRBX message.
- GNSS-SDR: Process the output of the GNSS software defined receiver project [24] from a UDP socket.
- Galmon network: Connect to the Galmon network [25] to process aggregated data from multiple receivers.

The library reports in chronological order about the OSNMA data received, the verification events and the authenticated navigation data. This logs also indicate when the receiver has enough authenticated data to calculate the first authenticated fix together with the time elapsed since it started to process information. This logging option can be used to obtain the TTFAF value under different protocol configurations. Finally, the library also has a status logging every sub-frame in JSON format, which is useful to see the general state of OSNMA and extract statistics about the scenario being processed. The status logging is used in the OSNMAlib webpage [26] to display live information of the OSNMA protocol.

III. PROPOSED TTFAF OPTIMIZATIONS

For standard (unassisted) TTFF, the user needs to acquire and track signals, and decode the ephemerides (WTs 1 to 5) from at least 4 satellites, and time (WTs 6 and 10) from at least one. For TTFAF, the receiver also needs to receive the tags authenticating each of the above, and a TESLA key in the next sub-frame. Therefore, a delay is introduced. For simplicity we use the shorthand TTFAF to refer to TTFAF *hot start*, i.e. when only the tags and one TESLA key are needed, and the receiver has the cryptographic information to authenticate the key with a so-called root key already in its possession. This is the standard operation mode, and the focus of our paper. Other start modes, such as *warm start* and *cold start*, are out of scope of this paper.

A. Page-Level Tag and Key Processing

At first glance, it may seem that OSNMA works at a per-satellite sub-frame level. The HKROOT is transmitted in numbered blocks that last one full sub-frame, and these sub-frame blocks need to be reordered to reconstruct the full message from multiple satellites. On the MACK side, a TESLA key is transmitted on every sub-frame to authenticate the tags of the previous sub-frame, and the tag order inside a sub-frame must be verified.

However, to optimize the OSNMA performance, a more granular approach should be taken. A Galileo sub-frame lasts 30 seconds and comprises 15 pages of 2 seconds each.

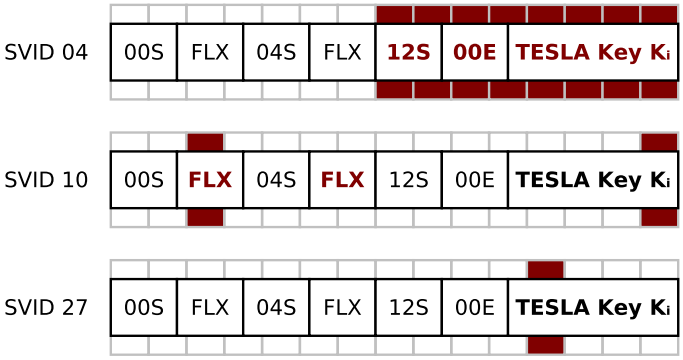


Fig. 2. Missing a page (colored in red) affects part of the cryptographic data of the sub-frame, but the rest is still valid. Missing one flex tag means all flex tags are missed because their position cannot be verified. The TESLA key is the same for all satellites so the receiver can reconstruct it by combining pages.

Discarding all well-received pages of a sub-frame because the receiver missed one of them is not the most optimal method. The intelligent use of these pages can lead to the recovery of much more OSNMA data, as analyzed in [27] with a previous configuration of OSNMA that did not include flex tags. Therefore, to obtain a better TTFAF in hot start mode, we have implemented a page-level processing technique consisting of the following ideas.

The first idea is to extract tag sections from correctly received pages of partially corrupted sub-frames. For the secure use of OSNMA, the tags' order within a sub-frame must still be verified using the MAC look-up table or the MAC sequence value for the flex tags. Yet no flex tag may, in principle, be used in a sub-frame if the MAC sequence value or any other flex tag is missing. Consequently, an clear downside of having multiple flex tags in a MAC look-up table configuration is that this optimization will lose efficacy.

The second idea is to reconstruct the TESLA key by exploiting the diversity in the transmission. During a strong fading and poor visibility scenario, the receiver may not be able to fully retrieve the TESLA key from any satellite in view during one sub-frame. Nonetheless, that does not mean the TESLA key of that sub-frame is lost. Since all Galileo satellites transmit the same key during the same sub-frame, it may be possible to reconstruct the key using correctly received pages from different satellites.

In Fig. 2, we show an example of how page-level processing helps extract valid cryptographic data. The tag sequence and key and tag sizes correspond to the OSNMA parameters transmitted during the OSNMA operational phase, illustrated in Fig. 1. The figure depicts a sub-frame where satellite 04 moves out-of-sight, and the receiver misses the last few pages of the sub-frame. Nevertheless, the first four tags are perfectly useful. Satellite 10 misses a page corresponding to a flex tag, which affects the other flex tag, but the other four tags are valid. Both Satellite 10 and Satellite 27 miss a page of the key, so the sub-frame ends without any key fully received. However, the optimization is able to reconstruct the key because the satellites missed a different page.

Naturally, these optimizations are especially useful in sce-

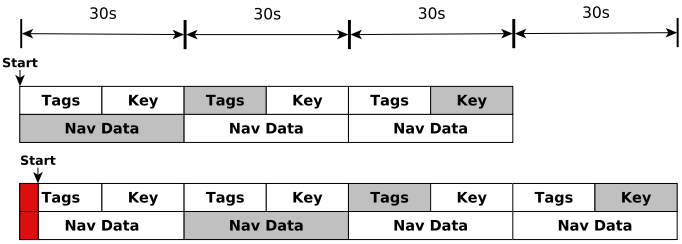


Fig. 3. Depiction of the OSNMA navigation data authentication process without any optimization; the gray elements are used together to authenticate the navigation data. If the receiver does not start aligned with a sub-frame, it has to wait until the next.

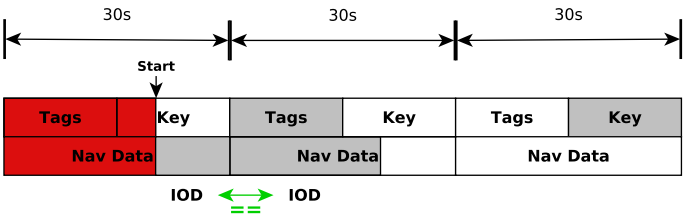


Fig. 4. Depiction of the authentication data process if the IOD of the data from the two sub-frames is the same. In this case, the missed navigation data can be retrieved from the next sub-frame.

narios with interference or fading where satellites are frequently out-of-sight. In a perfect open-sky scenario, only the low-elevation satellites entering or leaving the tracking horizon may have incomplete sub-frames.

B. IOD Navigation Data Link

Verifying the ADKD0 tags involves retrieving the navigation data, followed by the corresponding tag for this data in the subsequent sub-frame, and finally acquiring the TESLA key used for generating the tag in the third sub-frame. With this approach a TTFAF of 90 seconds can be achieved in the best-case scenario, but if the receiver misses the first pages of the first sub-frame, it has to wait until the next one to start with the process, hence delaying the TTFAF to a maximum of 119 seconds. Fig. 3 exemplifies these two cases, and Fig. 8 shows the TTFAF values depending on where the receiver starts in a sub-frame.

However, the ephemerides authenticated in ADKD0 change at a low rate and may be transmitted identically in several sub-frames. The data of multiple sub-frames can, therefore, be aggregated for authentication as long as it is the same. As discussed in the previous OSNMAlib paper [23] and the OSNMA receiver guidelines [22], one way to unambiguously reconstruct the navigation data from different sub-frames is to use the Issue of Data (IOD) value transmitted in the I/NAV words, except WT 5, which does not have an IOD. Hence it must be assigned based on the IOD of other words of the sub-frame.

With this optimization, the best case occurs when the receiver starts processing navigation data immediately before WT 3 because it is the latest word containing the sub-frame IOD. WT 3 is transmitted 8 seconds before the end of the sub-frame, and we will have to wait for another sub-frame for the tags and another one for the key. Therefore, the bes-case TTFAF is 60 seconds. If the navigation data does not change, the worst case occurs when the receiver starts immediately after WT 3 with a TTFAF of 97 seconds. A general example of this optimization is shown in Fig. 4, and Fig. 8 shows the TTFAF values for the IOD optimization as function of sub-frame offset.

C. Cut-Off Point Tag-Data Link

Originally, every tag included a 4-bit truncated IOD to link the tag with the data [28]. However, the unpredictability

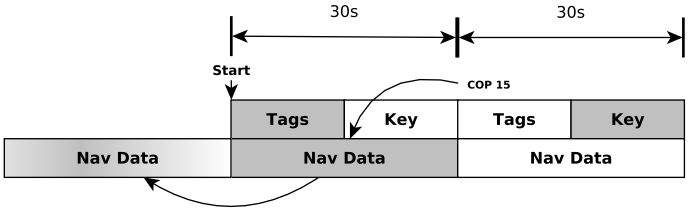


Fig. 5. If there is a tag in the key sub-frame that authenticates the navigation data with a COP higher than 1, the data received in the tags sub-frame is the same as the previous sub-frame and can be used to authenticate the first tag.

of the IOD evolution in the system could lead to failed authentications if not appropriately handled. After some years in which the field was defined as 'Reserved', the last OSNMA specification has replaced this field by the 4-bit Cut-Off Point (COP) field [21]. The COP indicates for how many sub-frames the navigation data authenticated with the tag has not changed. A value of 1 means that the authentication tag can only use navigation data from the previous sub-frame. A value of 15 (the maximum possible) indicates that the authentication tag can be verified using navigation data from the 15 previous sub-frames.

Although the original intention for the COP is to link the tag transmitting it with data from the previous sub-frames, we propose to use it to link other tags with the same data. With the traditional OSNMA approach, the receiver can never use the tags of the first sub-frame because the data transmitted in the previous sub-frame is unknown. However, this is the exact information given by the COP. If the navigation data has not changed, the COP of the tags in the key sub-frame will be greater than 1, indicating that the navigation data in the tags sub-frame is the same as in the prior sub-frame. Therefore, we can unambiguously link the tags received in the first sub-frame with the data of the first sub-frame (Fig. 5).

Nevertheless, for this optimization to work, the receiver must get one tag in two consecutive sub-frames for the same navigation data. The tag received in the first sub-frame is used to authenticate the navigation data when the key is disclosed in the second sub-frame. The COP of the tag received in the second sub-frame is used to verify that the data received in the first sub-frame can be linked with the first sub-frame's tag.

By using the COP value, it may seem that the previously discussed IOD optimization is no longer beneficial. However, both can be merged for even better TTFAF results. The same IOD logic to link navigation data from two sub-frames can be combined with the information provided by the COP as

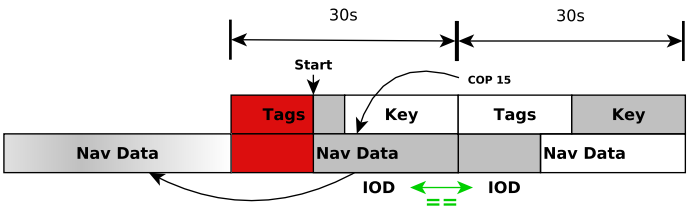


Fig. 6. The IOD value can be used to bind navigation data of different sub-frames, and the COP value can be used to ensure the link between the navigation data and the authentication tag.

depicted in Fig. 6. The IOD links the navigation data from two sub-frames, and the COP shifts that data to the previous sub-frame, linking it with the tag.

The operations performed by a receiver implementing the COP and IOD optimization are the following:

1) The receivers powers up in the middle of sub-frame SF_j, in time to get WTs 1, 3 and 5 from all Galileo satellites in view.
2) At the end of SF_j, the receiver has also extracted a few cross-authentication tags from connected satellites. These tags authenticate navigation data transmitted at the previous sub-frame (SF_{j-1}), data which the receiver missed because it was not powered on.
3) During the next sub-frame (SF_{j+1}), the receiver gets all the WTs from all satellites in view. For each satellite, if the IOD of these WTs is the same as the IOD of the WTs received at SF_j, the partial navigation data received at SF_j can be fully reconstructed.
4) Then, the receiver looks at the COP value of the authentication tags extracted during SF_{j+1}. If the COP value is greater than 1, it means that the reconstructed data for SF_j is the same as the navigation data transmitted at SF_{j-1} for the satellites targeted by the tags.
5) At this moment, the receiver knows the navigation data transmitted at SF_{j-1}, has the tags to authenticate it (received at SF_j) and the TESLA key to verify them (received at SF_{j+1}). Therefore, it can proceed with the navigation data verification.

Combining the COP and the IOD, we obtain, in the best-case scenario, a TTFAF of 44 seconds on the even sub-frames or 46 seconds on the odd sub-frames. The position of the last cross-authentication tag in the tag sequence (Fig. 1) defines the lowest possible TTFAF. If the navigation data does not change, the worst TTFAF is 73 seconds, when the receiver starts just after the last cross-authenticating tag.

Finally, there is a security consideration worth discussing with this optimization: by using the COP value of the tags in the key sub-frame, we use unauthenticated information. A possible attack would be to, in the event of a change of navigation data, reply the old data and modify the COP value of the following tag to indicate that the navigation data has not changed. With this method, the authentication would pass, and the receiver would think that 30 seconds old data was still being transmitted. However, the attack would be detected 30 seconds later when the modified tag is authenticated and. Moreover, by definition, the navigation data has a validity of 4 hours [29]. Therefore, there is no added security risk over a non-optimized scenario.

D. Benefit of a Lower TL

For OSNMA to work securely the receiver must know its synchronization accuracy with respect to Galileo System Time (GST). This synchronization lag is indicated by the TL value and defined in seconds. The rule is that no authentication tag or navigation data should be used after the first bit of the TESLA key is disclosed to the system to avoid the forging of cryptographic material. Therefore, to use the OSNMA ADKD0 tags for navigation data authentication, the maximum TL value is 30 seconds (Fig. 7a).

For the IOD optimization described in Section III-B, the receiver must be synchronized with the GST with a maximum lag of 25 seconds to achieve maximum performance. This value corresponds to the time between the last bit of the last relevant navigation data word for ADKD0 (WT 5) transmitted in the tag sub-frame and the first bit of the TESLA key (Fig. 7b). The IOD optimization would work with a TL of 30 seconds, but it could only link the WTs 2 and 4 with the navigation data of the previous sub-frame.

When using both the COP and the IOD to optimize the TTFAF as described in Section III-B, the maximum TL needed is 17 seconds. This value is the time between the last bit of the last relevant navigation data word for ADKD0 transmitted in the key sub-frame and the first bit of the TESLA key (Fig. 7c). Although with a TL of 1 second it would be possible also to use the WT 1, we decided to discard the case because the WT is transmitted at the same time as the key.

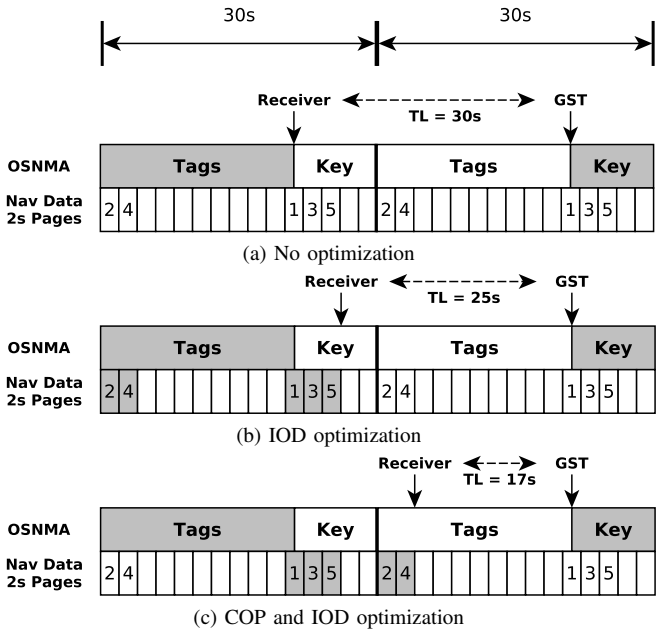


Fig. 7. The figure indicates the maximum time lag (TL) between the receiver and Galileo System Time (GST) for each relevant optimization. The darker color indicates the elements used for the authentication. The case 7a uses navigation data from the previous sub-frame.

Aside from the not-optimized case, the rest of TL calculations depend on the key size, the tag size, and the number of

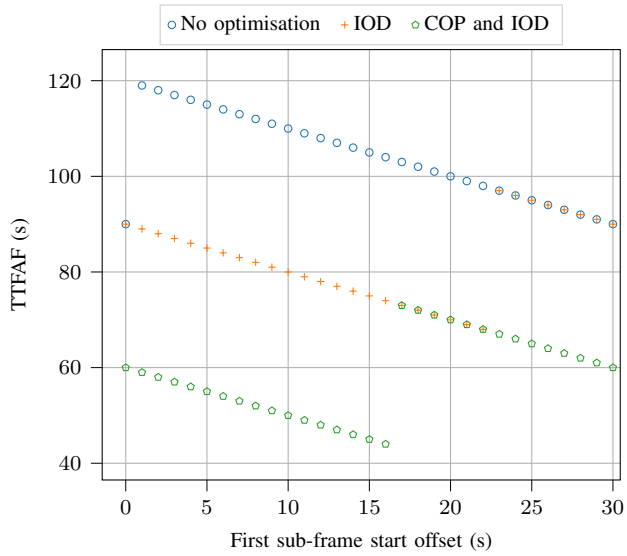


Fig. 8. Theoretical TTFAF values in the best-case scenario for the cases without optimization, with the IOD optimization, and with the COP-IOD optimization. The start time of the receiver within a sub-frame determines how long it will wait to get the first authenticated fix. The sub-frame start offset is relative to the first E1-B sub-frame from which the receiver starts decoding navigation data.

tags transmitted on each sub-frame. For these results, we used the configuration transmitted during the data recording for this paper on December 03, 2023, which is the same configuration used for the operational phase of OSNMA (Fig. 1). In Fig. 8, we show the best TTFAF value for the described optimizations as a function of offset of the first E1-B sub-frame for which the receiver starts getting navigation data. In all cases, we calculate the TTFAF using the appropriate TL value for the optimization and assume a nominal transmission of Galileo I/NAV E1-B pages without navigation data change.

Note that the navigation words order in the examples is extracted from the Galileo OS SIS ICD I/NAV Nominal Sub-Frame Structure for the E1-B signal [19], which is only indicative. Also, a multi-frequency receiver capable of decoding the I/NAV stream from the E5b-I signal would get different values for the TTFAF and TL.

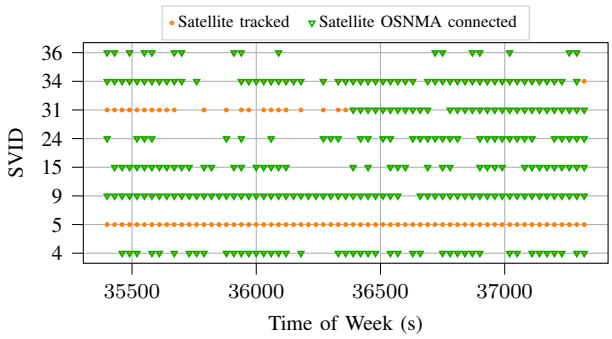
IV. SCENARIOS

To evaluate the performance of the discussed optimizations, we recorded Galileo data in three relevant scenarios: a dynamic Hard Urban scenario, a dynamic Soft Urban scenario, and a static Open-Sky scenario. Additionally, we have also processed the configuration 2 of the official OSNMA test vectors [22] because it contains the same tag sequence as the live transmitted data. For the dynamic recordings, we used a Septentrio mosaic-X5 with firmware version 4.14.0. For the static Open-Sky scenario, we used a Septentrio PolaRx5TR with firmware version 5.5.0.

The data recordings are saved in Septentrio Binary Format (SBF). This format contains the *GalRawINAV* block with all the information needed to post-process the files with OSNMAlib (Galileo I/NAV message bits, SVID, and receiver GST).



(a) Trajectory followed.



(b) Galileo satellites tracked and OSNMA connected.

Fig. 9. Hard Urban scenario recording of a walk in the European District of Brussels on December 03, 2023, from 09:50:00 to 10:22:30 UTC.

The recordings, containing all the GNSS logged information and format definition, are available in [30].

A. Hard Urban Scenario: Brussels, European District

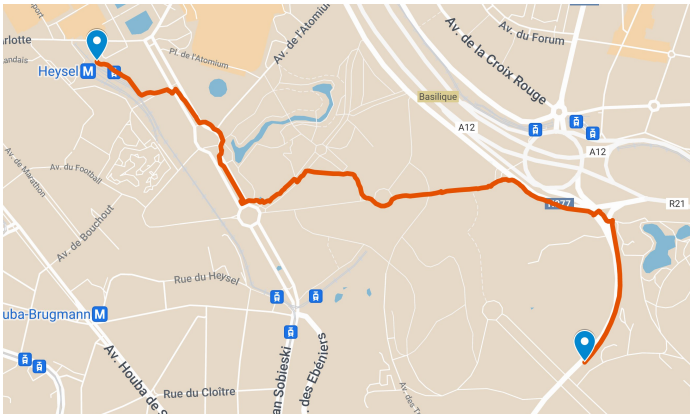
This scenario is a walk in the European District of Brussels, Belgium, on December 03, 2023, from 09:50:00 to 10:22:30 UTC; or GST 1267 35400 to 1267 37350. The trajectory (Fig. 9a) starts at the Parc de Bruxelles and quickly heads to the urban canyon of Rue Belliart, Rue de Trèves, and Rue de la Loi. Finally, it returns to the park and ends close to the start location.

During the trajectory, the receiver got navigation data from 8 different satellites (Fig. 9b). Only SVID 5 did not transmit OSNMA during the scenario; SVID 31 was initially disconnected but started transmitting OSNMA at half the scenario duration. The tracking is generally very volatile, as it corresponds with a hard urban scenario, with several entirely lost sub-frames.

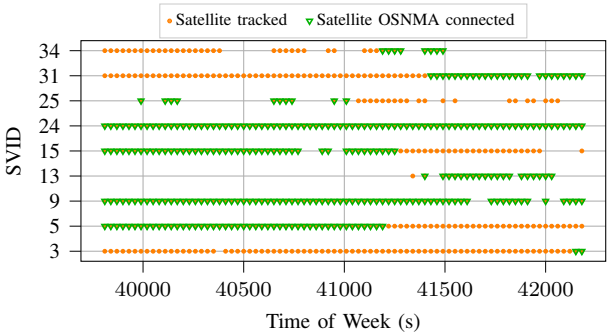
B. Soft Urban Scenario: Brussels, Atomium and Laeken Park

This scenario is a walk around the Atomium and surrounding parks in Brussels, Belgium, on December 03, 2023, from 11:03:24 to 11:43:53 UTC; or GST 1267 39804 to 1267 42233. The trajectory (Fig. 10a) walks close to the Atomium, enters Osseghem Park, and finally surrounds Laeken Park.

The receiver got navigation data from 9 satellites during the trajectory (Fig. 10b). The number of satellites connected



(a) Trajectory followed.



(b) Galileo satellites tracked and OSNMA connected.

Fig. 10. Soft Urban scenario recording of a walk around the Atomium of Brussels on December 03, 2023, from 11:03:24 to 11:43:53 UTC.

and disconnected is very balanced during the whole scenario, although there is a lot of change in which specific satellites transmit OSNMA.

C. Open-Sky: Leuven, Septentrio Offices

This scenario is a static recording of 60 minutes from the Septentrio Offices in Leuven, Belgium, on December 20, 2023, from 15:00:00 to 16:00:00 UTC; or GST 1269 313200 to 1269 316800.

The satellite visibility of this recording is excellent, as expected in an open-sky situation. A total of 11 satellites are received during the scenario, although the SVID 31 moves under the tracking horizon a few minutes in the recording (Fig. 11). All satellites move from connected to disconnected and vice-versa during the recording, but there are always at least four disconnected.

D. Test Vectors: Configuration 2

The OSNMA receiver guidelines [22] contain several test vectors to validate the implementation of the OSNMA protocol. The test vector titled 'Configuration 2' contains OSNMA data with the same structure as the operational live data described in Fig. 1, so it is helpful to test and compare the optimizations. We have run the first 30 minutes of this test vector, simulating from July 26 at 23:59:43 to July 27 at 00:29:43 UTC, or GST 1248 345601 to 1248 347401. These

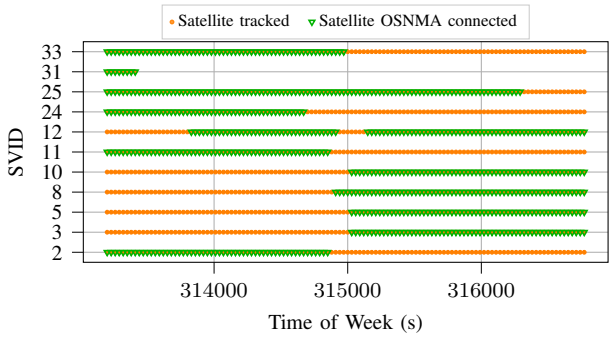


Fig. 11. Galileo satellites tracked and OSNMA connected in the Open-Sky static recording of 60 minutes from the Septentrio offices in Leuven, Belgium, on December 20, 2023; from 15:00:00 to 16:00:00 UTC.

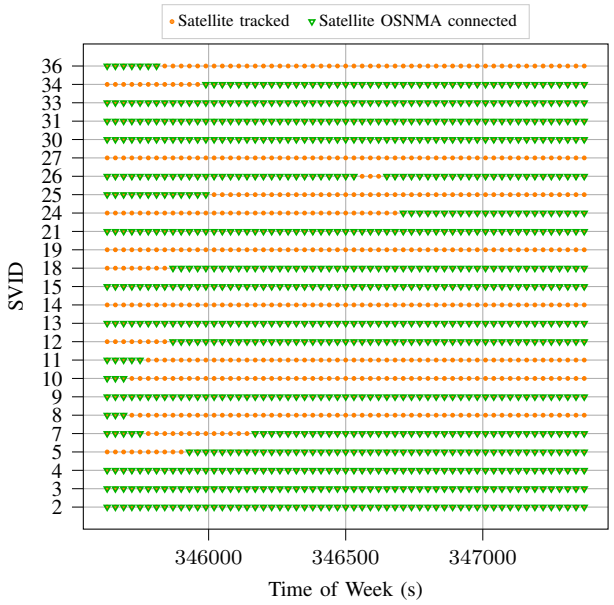


Fig. 12. Galileo satellites tracked and OSNMA connected in the Configuration 2 of the test vectors from the OSNMA receiver guidelines [22]. It is a synthetic scenario with all Galileo satellites visible.

test vectors must be formatted correctly to run in OSNMAlib because they are not chronologically sorted in the original format.

A particular characteristic of the test vectors is that they contain data from 25 Galileo satellites, which is impossible in a live recording (a characteristic seen in Fig. 12). Therefore, we cannot directly extrapolate the results to a real scenario.

V. TEST RESULTS

We implemented the optimizations in OSNMAlib in a flexible way so that they can be turned on or off at choice. To obtain multiple TTFAF values from the continuous recordings, we replayed the logs in OSNMAlib but started to process them each time one second later. With this technique, we can emulate a receiver powering up at any moment of the recording and obtain all the TTFAF values needed to evaluate the optimizations.

We decided to group the described optimizations into three accumulative groups to visualize their effects easily:

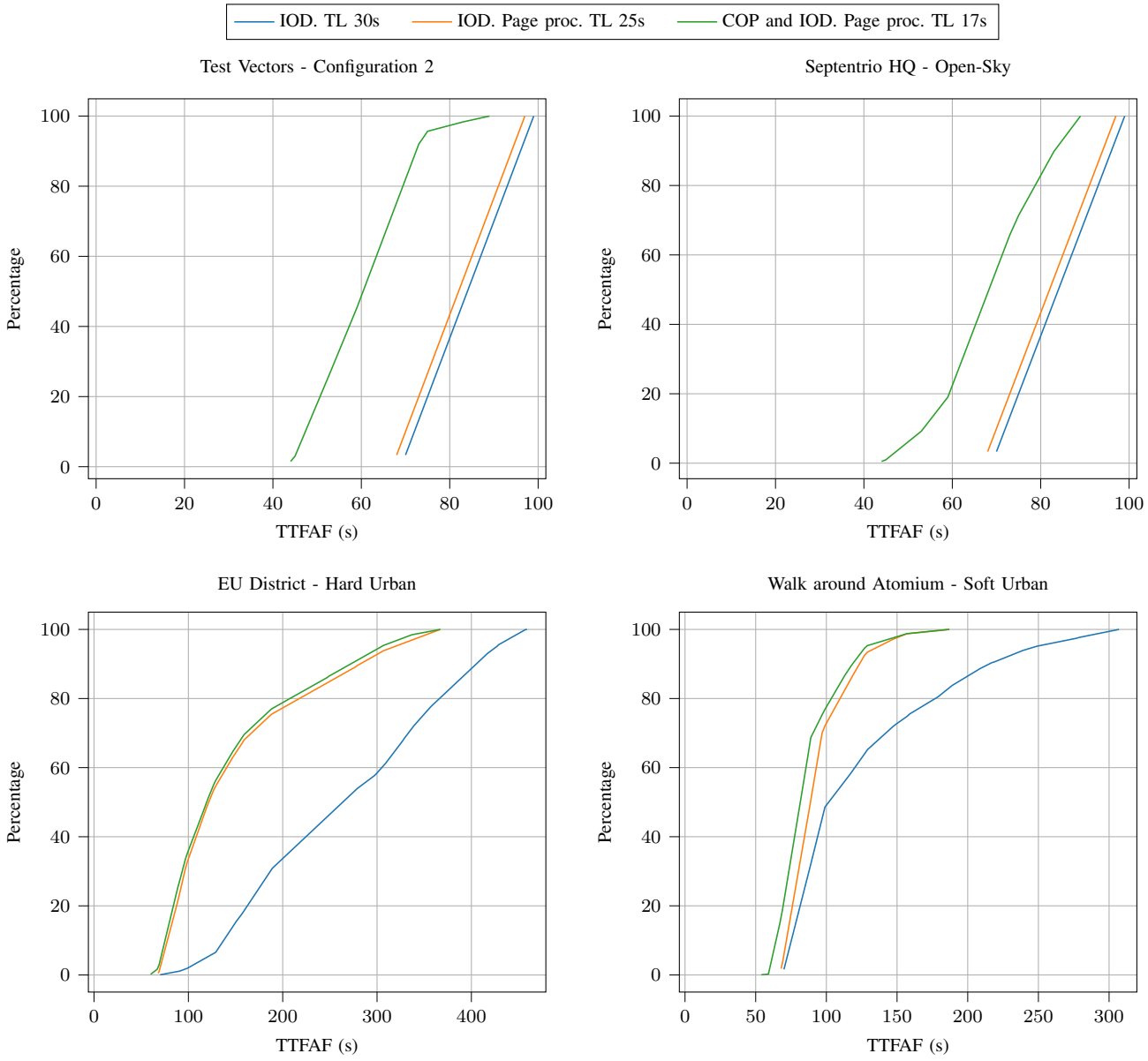


Fig. 13. The page-level processing optimization improves the TTFAF on the scenarios where pages are lost, such as the urban scenarios. The COP-IOD optimization improves as expected the TTFAF only in the scenarios where a lot of satellites are visible, while it struggles to bring any benefit in the urban scenarios.

- Standard OSNMA: Uses the IOD optimization to regenerate navigation data and the default TL of 30 seconds. While a standard OSNMA may not include the IOD optimization, it is briefly described in the OSNMA ICD, was present in the first version of OSNMAlib, and is already used in other state-of-the-art implementations. Hence, we use this configuration as a baseline.
- Page-Level Processing and Reduced TL: Uses the IOD optimization, a reduced TL of 25 seconds to use the IOD optimization at its full potential, and the page-level processing technique to extract valid navigation data from broken sub-frames.
- COP and IOD, with Page-Level Processing and Reduced TL: Uses the COP-IOD optimization to regenerate and propagate navigation data, a TL of 17 seconds to use completely the COP optimization, and page-level processing.

The results are presented in a cumulative distribution function (CDF) for each scenario to provide a global view of the optimization performance (Fig. 13). Additionally, in Fig. 14 we present the minimum TTFAF value obtained in each subframe to evaluate how the optimizations improve the TTFAF at different time periods.

A. *Page-Level Tag and Key Processing*

The page-level processing optimization works as expected: it improves the TTFAF in cases where Galileo I/NAV pages are lost. The two urban scenarios show a clear improvement between the case with page-level processing and the case without it (Fig. 13). Due to the buildings and trees, nearly any

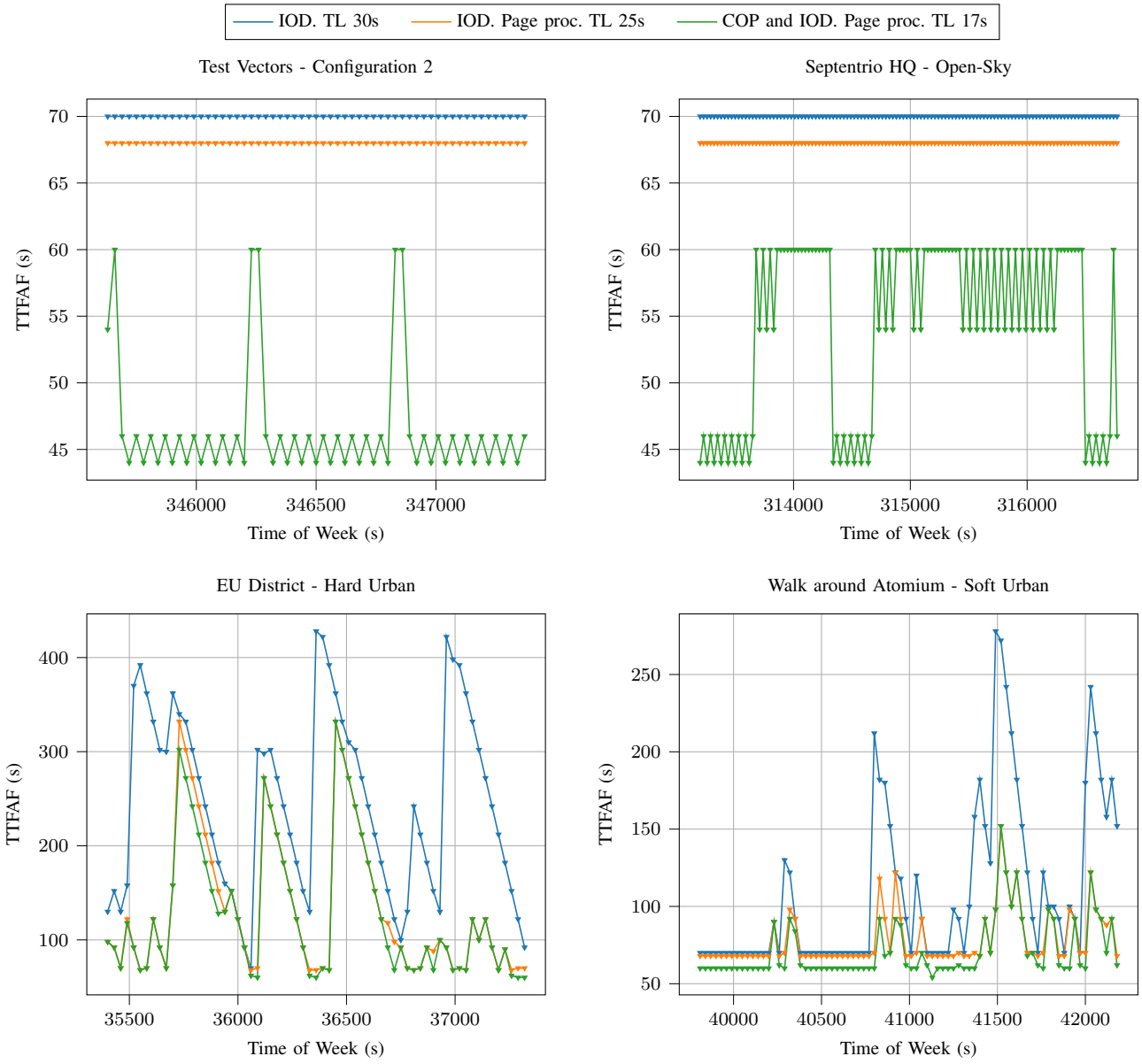


Fig. 14. The figures display the minimum TTFAF value obtained on each sub-frame. The theoretical minimum value with the COP-IOD optimization is 44 or 46 seconds. This value is only consistently achieved in the test vectors (except for the sub-frames with change of navigation data) and in some sub-frames of the Open-Sky scenario. The COP-IOD minimum value is never reached in the urban scenarios due to the high number of lost pages. However, for the same reason, the page-level processing optimization substantially improves the TTFAF values in the urban scenarios.

satellite has dropped pages at some point, and the optimization extracts all it can from the left pages. For example, in the Hard Urban scenario, nearly 80% of the TTFAF values are lower than 200 seconds when using page-level processing, while the TTFAF increases to 360 seconds for the case without this optimization. Unsurprisingly, the improvement is more significant in the Hard Urban scenario than in the Soft Urban case, where fewer pages are lost.

When looking at the minimum TTFAF value per sub-frame (Fig. 14) for the same urban scenarios, the effect of the harsh environment is displayed in the form of time spikes. In some cases, the page processing optimization follows the same spike as the not-optimized case but with slightly lower values. However, when this does not happen, the improvement is substantial (for example, around Time of Week 37000 in the Hard Urban scenario).

In the Open-Sky scenario and test vectors, the two-second improvement seen in the minimum TTFAF per sub-frame and in the displacement of the CDF is due to the reduction of the TL to 25 seconds and not to the page-level processing. Reducing the TL allows linking navigation data of two sub-frames using the IOD of the WT 3 instead of 1. The WT 3 is transmitted 2 seconds after the WT 1, hence the improvement of 2 seconds in the minimum TTFAF when reducing the TL to 25 seconds.

The ineffectiveness of the page-level processing for the test vectors is expected: the synthetic nature of the scenario implies that no pages are lost. In the open-sky scenario we recorded no

satellite loses relevant pages for OSNMA, which is a possible situation. Nevertheless, note that this can differ for other open-sky scenarios: some low-elevation satellites might lose pages.

B. IOD-COP Tag-Data Link

The IOD-COP tag-data link optimization struggles to yield any improvement in the urban scenarios (Fig. 13). The essential requirement of obtaining two tags for the same satellite and navigation data in two consecutive sub-frames is hardly met due to the fading characteristic of these recordings. Also, the reduced number of satellites in view makes this requirement even harder to fulfill. Still, it improves slightly more in the Soft Urban than in the Hard Urban scenario.

However, the optimization works according to the theory in the test vectors, improving the TTFAF very substantially. Some cases worse than expected can be seen as sub-frames with a minimum TTFAF of 60 seconds in Fig. 14 because the test vectors contain sub-frames with change of navigation data. When sufficient navigation data changes, the optimization cannot make assumptions based on the COP value for all tags, degrading the TTFAF. Despite that, it is always better than the cases without the optimization. Moreover, we can see how the best case for each sub-frame alternates between 44 and 46 seconds, determined by whether the tag sequence is for the odd or even sub-frame (see in Fig. 1 the position of the last cross-authentication tag `E00`).

Strangely, in the Open-Sky scenario, the optimization does not seem to work as well as theorized, even when tracking 10 satellites for most of the time. The results are all right; the improvement, when compared with the non-optimized values, is clear and huge, but we are in several sub-frames far away from the 44 to 46 seconds mark. Additionally, we can see how the minimum TTFAF value for the sub-frames is discrete: 60, 54, 46, and 44 seconds. These values are directly linked to the position of the ADKD0 tags in the tag sequence, described in Fig. 1.

When a receiver implementing the COP-IOD optimization starts aligned with the beginning of the sub-frame, it receives four ADKD0 tags on that sub-frame from each connected satellite. If four of these tags are repeated in the next sub-frame, a TTFAF of 60 seconds can be obtained. This case is very likely with 10 satellites in view for the Open-Sky scenario. Thus, we do not see any sub-frame with a minimum TTFAF greater than 60 seconds.

If the receiver starts later within the sub-frame and misses the first tag, it also loses the ability to authenticate the flex tag positions, effectively losing all flex tags. Therefore, it can only use the ADKD0 tags indicated with `00E` in Fig. 1. The discrete TTFAF values for the Open-Sky scenario in Fig. 14 are obtained when the receiver starts just before this ADKD0 tags. However, obtaining these reduced TTFAF values is more challenging because the receiver acquires fewer tags on the starting sub-frame.

For each of the three tested optimization combinations, we have chosen the best, average, and percentile 95 values as relevant TTFAF metrics and displayed them in Tables I, II, and III. The combination of IOD and Cut-Off Point tag-data link with the page-level processing and a TL of 17 seconds always brings the best results regardless of the circumstance.

TABLE I
TTFAF METRICS USING THE IOD DATA LINK OPTIMIZATION WITH A TL OF 30 SECONDS

Optimization	Best (s)	Average (s)	P95 (s)
Test Vectors	70.0	84.5	98.0
Open-Sky	70.0	84.5	98.0
Soft Urban	70.0	127.5	248.0
Hard Urban	70.0	266.1	427.0

TABLE II
TTFAF METRICS USING THE IOD DATA LINK OPTIMIZATION WITH A TL OF 25 SECONDS AND PAGE-LEVEL PROCESSING

Optimization	Best (s)	Average (s)	P95 (s)
Test Vectors	68.0	82.5	96.0
Open-Sky	68.0	82.5	96.0
Soft Urban	68.0	94.1	137.0
Hard Urban	68.0	151.1	318.5

TABLE III
TTFAF METRICS USING THE IOD-COP TAG-DATA LINK OPTIMIZATION WITH A TL OF 17 SECONDS AND PAGE-LEVEL PROCESSING

Optimization	Best (s)	Average (s)	P95 (s)
Test Vectors	44.0	60.9	75.0
Open-Sky	44.0	68.8	87.0
Soft Urban	54.0	87.5	129.0
Hard Urban	60.0	146.1	305.0

C. OSNMA Cross-Authentication Algorithm

The reason why, even in an open-sky scenario, the COP optimization is not working as expected lies in the OSNMA cross-authentication algorithm. Currently, OSNMA only transmits cross-authentication tags for disconnected satellites. This behavior creates an imbalance in the number of ADKD0 tags a satellite receives during a sub-frame conditioned by its connection status.

If a satellite is connected, the receiver will only obtain one ADKD0 tag for that satellite for the whole sub-frame: the self-authenticating tag, which is always transmitted at the first position of the tag sequence (see `00S` in Fig. 1). However, if the satellite is disconnected, the receiver will get authentication tags from other satellites transmitted in the cross-authentication positions of the tag sequence. These cross-authentication positions are currently three per sub-frame since the FLX positions are used for cross-authenticating (see `00E` and `FLX` in Fig. 1). Moreover, the self-authenticating tag is always transmitted at the first position, so if the receiver starts to receive data two seconds after the start of the sub-frame, it is sure not to receive any tag for that connected satellite for the rest 28 seconds of the sub-frame.

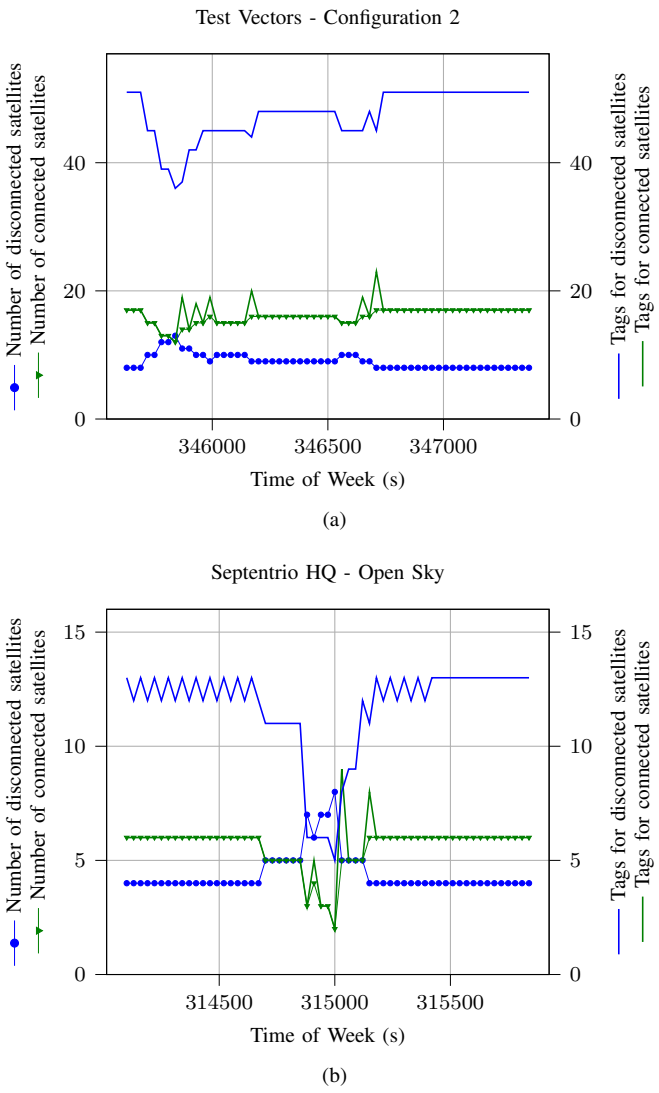


Fig. 15. On the left y-axis, connected and disconnected satellites. On the right y-axis, number of authentication tags received per sub-frame for connected and disconnected satellites. A connected satellite only gets 1 tag per sub-frame while a disconnected satellite gets up to 5 tags per sub-frame in average.

This cross-authentication sequence may maximize all-in-view satellite authentication with few connected satellites. However, it creates the paradox that the more disconnected satellites in view, the better the COP-IOD optimization works. In particular, at least four disconnected satellites are needed for the COP-IOD optimization to work to its full extent. These satellites will receive cross-authentication ADKD0 tags and be used in the PVT calculation to get the first authenticated fix. An extra two connected satellites are needed to have the possibility to receive, in two consecutive sub-frames, four ADKD0 tags for the same four different satellites. Thus, the optimization works partially with six satellites: two connected and four disconnected. Another two connected satellites are needed to have the optimization working completely as theorized. Now, with eight satellites (four connected and four disconnected), it is possible to get four ADKD0 tags for four different satellites at the last cross-authentication position of the first sub-frame and have the minimum TTFAF of 44/46 seconds. However, the necessity for eight satellites is a lower bound; it will depend on the specific satellites being authenticated in the cross-authentication sequence.

The cross-authentication tag unbalance becomes apparent when examining the number of tags received per sub-frame for the test vectors and the Open-Sky scenario (Fig. 15). The number of tags per sub-frame for connected satellites is always the same as the number of connected satellites (hence, one tag per satellite). Yet, there are some exceptional sub-frames where there is more than one tag per satellite: when a previously disconnected satellite joins the OSNMA transmission. In those cases, because the tags are transmitted for data in the previous sub-frame, the system still transmits tags for the satellite's data before it started to transmit OSNMA.

On the other hand, the number of tags received for disconnected satellites is up to 5 times the number of disconnected satellites in the test vectors (Fig. 15a). The ratio is a bit lower for the Open-Sky scenario (Fig. 15b) because not all satellites are in view, so some tags are lost. In either case, the tags received for disconnected satellites are much more than those for connected satellites, which restricts the scenarios where the COP-IOD optimization may work at its best.

VI. CONCLUSION

Two concrete ideas have been proposed in this paper to improve the TTFAF: page-level processing and COP-IOD optimization. The analysis of the proposed optimizations over three distinct scenarios (Open-Sky, Hard Urban, and Soft Urban) and the test vectors show how the TTFAF can be greatly improved by treating the navigation data received optimally. Moreover, both methods are proven to be complementary when examined in diverse environments.

The page-level processing for authentication tags and TESLA keys is extremely effective for the urban scenarios, improving the average TTFAF from 127.5 seconds to 94.1 seconds in the Soft Urban scenario and from 266.1 seconds to 151.1 seconds in the Hard Urban scenario. Due to the low satellite visibility and fading, the COP-IOD optimization only improves marginally the average TTFAF for the Soft and Hard Urban scenarios, obtaining 87.5 and 146.1 seconds, respectively.

However, the opposite occurs for the test vectors and the Open-Sky scenario: the page-level processing does not improve the TTFAF, but the COP-IOD optimization reduces it substantially. In both cases, the lack of missed pages inhibits the page-level processing gains. Nonetheless, the COP-IOD optimization benefits from the good satellite visibility of the Open-Sky scenario and the ample number of satellites present in the test vectors. By using this last optimization, the average TTFAF improves from 82.5 seconds to 60.9 seconds for the test vectors and 68.8 seconds for the Open-Sky scenario. The improvement for the best TTFAF value is even more impressive, from 68.0 seconds to 44.0 seconds in both cases.

The COP-IOD optimization does not work entirely as expected in the Open-Sky scenario due to the cross-authentication algorithm followed by OSNMA. The algorithm never sends cross-authentication tags for satellites transmitting

OSNMA, which generates an imbalance in the number of tags received for each satellite. This behavior adds extra constraints in the minimum number of satellites in view for the COP-IOD optimization to reach the minimum TTFAF of 44.0 seconds consistently.

To further improve the OSNMA metrics, it could be useful to implement a multi-frequency library that also uses the I/NAV messages transmitted at E5b. Moreover, Galileo recently implemented four word types that allow the recovery of missed clock and ephemeris pages using Reed-Solomon encoding, which can significantly improve the performance of OSNMA in urban scenarios [31].

ACKNOWLEDGMENT

The authors would like to thank Sibren De Bast for his insights and comments.

REFERENCES

[1] L. Scott, "Anti-spoofing & authenticated signal architectures for civil navigation systems," in *International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS)*, 2003, pp. 1543–1552.

[2] K. D. Wesson, J. N. Gross, T. E. Humphreys, and B. L. Evans, "GNSS signal authentication via power and distortion monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 739–754, 2017.

[3] Ç. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan, "An INS monitor to detect GNSS spoofers capable of tracking vehicle position," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 131–143, 2017.

[4] J. M. Anderson, K. L. Carroll, N. P. DeVilbiss, J. T. Gillis, J. C. Hinks, B. W. O'Hanlon, J. J. Rushanan, L. Scott, and R. A. Yazdi, "Chips-message robust authentication (Chimera) for GPS civilian signals," in *International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+)*, 2017, pp. 2388–2416.

[5] I. Fernandez-Hernandez, J. Winkel, C. O'Driscoll, S. Cancela, R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, A. Dalla Chiara, C. Sarto, D. Blonski *et al.*, "Semi-Assisted Signal Authentication for Galileo: Proof of Concept and Results," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[6] K. Zhang, E. G. Larsson, and P. Papadimitratos, "Protecting GNSS open service navigation message authentication against distance-decreasing attacks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 2, pp. 1224–1240, 2021.

[7] T. E. Humphreys, "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073–1090, 2013.

[8] I. Fernandez-Hernandez, V. Rijmen, G. Seco-Granados, J. Simon, I. Rodríguez, and J. D. Calle, "A navigation message authentication proposal for the Galileo open service," *NAVIGATION: Journal of the Institute of Navigation*, vol. 63, no. 1, pp. 85–102, 2016.

[9] M. Götzelmann, E. Köller, I. Viciano-Semper, D. Oskam, E. Gkougkas, and J. Simon, "Galileo open service navigation message authentication: Preparation phase and drivers for future service provision," *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 3, 2023.

[10] L. Musumeci, N. Batzilis, G. Caparra, S. Circiu, P. Crosta, D. Ibañez, X. Otero, N. Sirikan, S. Wallner *et al.*, "OSNMA User Performance Assessment at ESA/ESTEC–System Qualifications Tools and Methodologies," in *International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, 2023, pp. 538–556.

[11] A. Perrig, J. Tygar, A. Perrig, and J. Tygar, "TESLA broadcast authentication," *Secure Broadcast Communication: In Wired and Wireless Networks*, pp. 29–53, 2003.

[12] I. Fernandez-Hernandez, T. Ashur, and V. Rijmen, "Analysis and recommendations for MAC and key lengths in delayed disclosure GNSS authentication protocols," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 3, pp. 1827–1839, 2021.

[13] M. Paonni, M. Anghileri, T. Burger, L. Ries, S. Schlötzer, B. Schotsch, M. Ouedraogo, S. Damy, E. Chatre, M. Jeannot *et al.*, "Improving the Performance of Galileo E1-OS by Optimizing the I/NAV Navigation Message," in *International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, 2019, pp. 1134–1146.

[14] L. Cucchi, S. Damy, M. Paonni, M. Nicola, and B. Motella, "Receiver testing for the galileo E1 OSNMA and I/NAV improvements," in *International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*, 2022, pp. 808–819.

[15] T. Hammarberg, J. M. V. García, J. N. Alanko, and M. Z. H. Bhuiyan, "An Experimental Performance Assessment of Galileo OSNMA," *Sensors*, vol. 24, no. 2, p. 404, 2024.

[16] A. Galan, I. Fernandez-Hernandez, G. Seco-Granados. (2024) OSNMAlib. GitHub repository. [Online]. Available: https://github.com/Algafix/OSNMA

[17] Septentrio NV. (2024) mosaic-X5 GNSS receiver module. Accessed: February 28, 2024. [Online]. Available: https://www.septentrio.com/en/products/gps/gnss-receiver-modules/mosaic-x5

[18] ——. (2024) PolaRx5TR GNSS receiver. Accessed: February 28, 2024. [Online]. Available: https://www.septentrio.com/en/products/gps/gnss-reference-receivers/polarx-5tr

[19] "European GNSS (Galileo) Open Service, Signal-In-Space ICD, Issue 2.1," European Union, Tech. Rep., Nov 2023.

[20] I. Fernandez-Hernandez, S. Damy, M. Susi, I. Martini, J. Winkel *et al.*, "Galileo Authentication and High Accuracy: Getting to the Truth," *Inside GNSS*, February 2023, accessed: March 5, 2024. [Online]. Available: https://insidegnss.com/galileo-authentication-and-high-accuracy-getting-to-the-truth/

[21] "European GNSS (Galileo) Open Service, Galileo OSNMA SIS ICD, Issue 1.1," European Union, Tech. Rep., Oct 2023.

[22] "European GNSS (Galileo) Open Service, Galileo OSNMA Receiver Guidelines, Issue 1.3," European Union, Tech. Rep., Jan 2024.

[23] A. Galan, I. Fernandez-Hernandez, L. Cucchi, and G. Seco-Granados, "OSNMAlib: An Open Python Library for Galileo OSNMA," in *Workshop on Satellite Navigation Technology (NAVITEC)*, 2022, pp. 1–12.

[24] C. Fernández-Prades. (2024) GNSS-SDR. CTTC. Open-source GNSS software-defined receiver. [Online]. Available: https://gnss-sdr.org

[25] B. Hubert. (2024) Galmon network. GitHub repository. [Online]. Available: https://github.com/berthubert/galmon

[26] A. Galan and S. Fontfreda. (2024) OSNMAlib website. Accessed: February 8, 2024. [Online]. Available: https://osnmalib.eu/

[27] S. Damy, L. Cucchi, and M. Paonni, "Performance Assessment of Galileo OSNMA Data Retrieval Strategies," in *Satellite Navigation Technology (NAVITEC)*, 2022.

[28] I. Fernández, V. Rijmen, T. Ashur, P. Walker, G. Seco, J. Simón, C. Sarto, D. Burkey, and O. Pozzobon, "Galileo Navigation Message Authentication Specification for Signal-In-Space Testing – v1.0," *European Commission*, vol. 11, 2016.

[29] "European GNSS (Galileo) Open Service, Service Definition Document, Issue 1.3," European Union, Tech. Rep., Nov 2023.

[30] A. Galan, I. Fernandez-Hernandez, and W. De Wilde. (2024) GNSS Recordings for Galileo OSNMA Evaluation. IEEE Dataport. [Online]. Available: https://dx.doi.org/10.21227/a0nm-kn45

[31] S. Damy, L. Cucchi, B. Motella, and M. Paonni, "Increasing OSNMA Performance with Galileo I/NAV Improvements: Tests in Degraded Reception Conditions," in *International Technical Meeting of The Institute of Navigation (ITM/PTTI)*, 2024, pp. 390–402.