High-Accuracy and Resilient GNSS Receiver for an Autonomous Vehicle

Filipe Carvalho, Ricardo Prata, Bruno Cardeira, Carlota Cardoso, Rui Nunes, *Deimos Engenharia*
António Fernández, *Deimos Space*

BIOGRAPHY (IES)

Filipe Carvalho received a Master degree in Engineering Physics from Instituto Superior Técnico, Lisbon, in 2019. He joined Deimos Engenharia's Receivers & Applications division, within the GNSS Business Unit later that year, where he has been working as a software developer and technical manager. His work within the GNSS unit includes the development of software for uplink communication and GNSS navigation data processing, including the Galileo OSNMA and HAS services.

Ricardo Prata received the degree in Systems Engineering of Telecommunications and Electronics from Polytechnic Institute of Lisbon (IPL), Portugal, in 2006. He was co-founder of an IT company for 8 years focused on developing low power consumption products. He joined DEIMOS in 2017, where he has worked in GNSS, TT&C and QKD (Quantum Key Distribution) areas, including the development of terrestrial and Space receivers. Currently, he is the responsible of Receivers & Applications division within GNSS Business Unit.

Bruno Cardeira is a Senior Research Engineer, Lic. (5 years, 2004), M.Sc. (2009) degrees in electrical engineering from the Instituto Superior Técnico (IST), Portugal. He was a research engineer at IST, working on unmanned autonomous robotics. At NATO STO CMRE, he acquired expertise in real-time embedded implementation of underwater acoustics (UW) signal processing algorithms, for detection, tracking, classification and UW communications for AUVs. He joined Deimos Engenharia, in 2023 where he currently works in the GNSS Receivers and Applications group.

Carlota Cardoso received the B.Sc and M.Sc. degrees in Engineering Physics from Instituto Superior Técnico, Portugal. Worked for a year, under a research grant, at the Laboratory of Instrumentation and Experimental Particle Physics (LIP) in an ESA project to provide expert support to two high energy radiation detectors: BERM (BepiColombo mission) and RADEM (JUICE mission). In 2022, she joined Deimos Engenharia, integrating the GNSS Receivers and Applications group and working primarily on the development of the G3 receiver.

Rui Nunes is a Portuguese Aerospace Engineer specializing in GNSS receivers and applications. His Master's thesis focused on a simplified Galileo E5 receiver. Since 2017, he has been working at Deimos Engenharia, with four years spent at ESA's ESTEC on the Galileo G1 project as a receiver testing engineer. His expertise spans receiver architecture, performance, and testing. Rui has also participated in the Jammertest 2023, gaining hands-on experience with jamming and spoofing signal testing.

Antonio Fernández received his M.S. degree in Aeronautical Engineering from the Polytechnic University of Madrid in 1994, and in 2003 he obtained a MS in Physics from the UNED University of Spain. He has been working in GNSS since 1996, in the fields of GNSS receivers and systems. He co-founded DEIMOS Space in 2001, where he is currently managing the GNSS Business Unit.

ABSTRACT

In this paper, the G3STAR GNSS receiver is described in the context of GAMMS, a Horizon 2020 project that aims at the development of a mapping robot for high-definition navigation map production, useful for the navigation systems of autonomous vehicles. An overview of the GAMMS project is provided and the role of the G3STAR receiver within the project is highlighted. This GNSS receiver is a Galileo-based receiver, making use of its Open Service features. These include the new Galileo services: the High Accuracy Service (HAS) and the Navigation Message Authentication service (OSNMA). In this paper, these Galileo features are described at a high level, with the main focus on their implementation on the receiver. Preliminary results are then shown and discussed. The G3STAR receiver is able to obtain and decode HAS messages from the Galileo E6-B signals, with the use of a GNSS constellation simulator tool. It is also able to process the OSNMA bits from live Galileo E1-B I/NAV messages and validate navigation data, including the handling of special OSNMA operations, namely chain renewals and public key revocations.

1. INTRODUCTION

High-Definition (HD) maps are one of the key enabling technologies for automated driving [1]. These maps are significantly more detailed, accurate and reliable than conventional navigation maps, and are required by Level of Automation (LoA) 4 Autonomous Vehicles (AVs) to operate safely [2]. Such maps should be cm level accurate and frequently updated, on a daily to weekly basis. Getting the "appropriate" LoA 4 maps is, today, one of the challenges of the automotive industry. These maps are built with the use of Mobile Mapping Systems (MMS) and identify obstacles, lanes, traffic signals and other environment variables, relevant for the navigation of an AV.

To achieve the desired positional accuracy, the development of highly accurate sensors is essential. One type of such sensors is the GNSS receivers, which will be the focus of this paper, particularly the development of the G3STAR receiver by Deimos Engenharia [3], in the scope of the GAMMS project [4].

The G3STAR receiver is a triple-band GNSS receiver, capable of processing the Galileo E1, E5a and E6 signals. In this project, it will be integrated and tested in an AV, together with other sensors, and navigation and mapping software, to assess the feasibility and performance of such an Autonomous terrestrial Mobile Mapping System (AMMS).

The receiver needs to be accurate enough to comply with the cm level positioning accuracy demanded by the HD mapping. For this it will take advantage of the Galileo High-Accuracy Service (HAS), which provides data that increases the accuracy of the navigation solution. Furthermore, for AV applications, resilience to spoofing attacks, which consist on the generation and transmission of false GNSS signals aimed at misleading the navigation solutions of receivers, is a necessity. For this reason, G3STAR will also make use of the Galileo's Open Service: Navigation Message Authentication (OSNMA), which provides the means to authenticate Galileo signals, ensuring the integrity of the GNSS service.

1.1. Galileo HAS

The High-Accuracy Service of Galileo is a Precise Point Positioning (PPP) service broadcast by the Galileo E6 Signal-in-Space (SiS) and provided over the internet. It provides accurate satellite data (clocks, orbits and biases) with a global coverage, free of charge [5]. It was declared operational on the 24th of January 2023.

The G3STAR receiver will obtain the Galileo HAS data broadcast via SiS during its operations, which will be used by the navigation algorithm to improve the navigation solution. This data is transmitted within the Galileo E6 C/NAV navigation message in the E6-B signal component of Galileo. These messages include one or several sets of data applicable to specified satellite vehicles (SVs) and Galileo or GPS signals: Orbit corrections, Clock Full-Set corrections, Clock Subset corrections, Code Bias and Phase Bias corrections [6].

Orbit corrections include corrections to the radial, in-track and cross-track axis of specified SVs. Clock Full-Set and Clock Subset corrections include Delta Clock Corrections for each specified SV, as well as Delta Clock Multipliers that can be applied to the Delta Clock Corrections. Code Bias corrections are provided for each specified signal of each specified SV (for instance, to the E1-B I/NAV OS signal of Galileo SV 1, or the L5-Q signal of GPS SV 10). Similarly, Phase Bias corrections are provided for each specified signal of each specified SV. These corrections also include Phase Discontinuity Indicators, which are incremented every time there is a phase bias discontinuity which requires a re-initialisation of the fixed ambiguity for the corresponding satellite and signal.

1.2. Galileo OSNMA

The OSNMA data is transmitted in the E1-B Galileo Open Service signal, within the Galileo I/NAV navigation message. It is a service that provides cryptographic data to authenticate Galileo Open Service navigation messages [7]. It is based on the *Elliptic*

Curve Digital Signature Algorithm (ECDSA) family [8], which is a form of public-key or asymmetric cryptography, and on the *Timed Efficient Stream Loss-tolerant Authentication* (TESLA) authentication protocol [9]. The receiver is in possession of a Public Key, used to authenticate data that was signed with a Private Key, which is secret and only known by the Galileo Service Centre (GSC), which generates the cryptographic data, broadcast by the Galileo constellation. The data that is signed and authenticated include a so-called TESLA root key (KROOT). This key is the root of a pre-generated one-way chain of keys (TESLA chain). Each key in this chain can be recursively authenticated by the previous key of the chain, until KROOT is reached, which had been previously authenticated by the ECDSA. Each key in the TESLA chain is also associated with a set of Message Authentication Codes (MACs) [10] [11], which are generated from a set of bits from a navigation message and the corresponding TESLA key. This protocol includes a delay between the transmission of the MAC data, truncated into tags, and TESLA keys, to make it difficult to predict the key that generated the MAC and prevent a spoofer from generating valid tags from false navigation messages. A simple schematic of this protocol is shown in Figure 1.

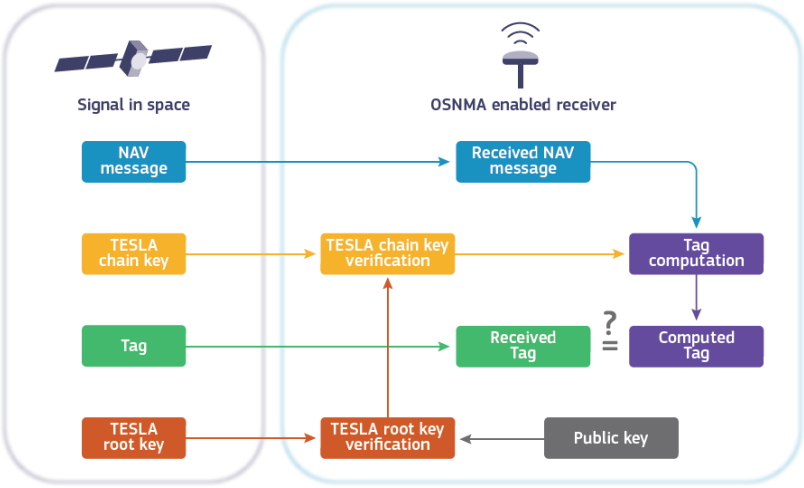

Figure 1. OSNMA Schematic. Taken from [12].

This protocol uses the same TESLA chain for all satellites, which allows a given satellite to cross authenticate data transmitted by other satellites. This way, even satellites that are not broadcasting OSNMA data are able to be authenticated.

2. GAMMS OVERVIEW

GAMMS (Galileo/GNSS-Based Autonomous Mobile Mapping System) is an European Union Agency for the Space Programme (EUSPA) funded Horizon2020 project, which aims at developing an Autonomous terrestrial Mobile Mapping System, based on the tight integration of AV, navigation/geodetic, and Artificial Intelligence technologies. The GAMMS consortium comprises different Industry and academic partners, namely Deimos Engenharia, GeoNumerics, GEOSAT, ENIDE, École Polytechnique Fédérale de Lausanne (EPFL), Pildo Labs, Virtual Vehicle and Solid Potato.

In summary, the goal of GAMMS is to develop a proof-of-concept prototype that automates the HD navigation map-making process, using state-of-the-art technologies. For this, an autonomous vehicle is used, to reduce the need for mobile mapping crews. A Vehicle Dynamic Model is implemented to assess its impacts on the improvement of the navigation system, for the vehicle and map making. A sensor suite, including GNSS receivers, are employed to provide the inputs for the navigation system. The mobile mapping system, that generates the HD maps, is aided by a multispectral laser scanner for Distress Mapping. Mapping software to automate the HD mapping process is implemented, to achieve the fully autonomous and automated terrestrial mobile mapping system – a mapping robot.

In this context, a robust and accurate navigation system is being developed, to allow the generation of HD maps. This navigation system of GAMMS includes the Galileo-based GNSS receiver that is the subject of this paper – G3STAR –, an Inertial Measurement Unit (IMU), Distance Measuring Instruments (DMIs) and a trajectory determination software module, which will receive information from the GNSS receiver and the sensors to compute a highly accurate trajectory of the autonomous vehicle. The DMIs include an odometer and cameras for visual odometry. The trajectory determination software – NEXA – is a software engine that computes trajectories with sensor fusion in real-time and post-processing, based on an Invariant Extended Kalman Filter [13]. A simple schematic of the GAMMS navigation system is depicted on Figure 2.

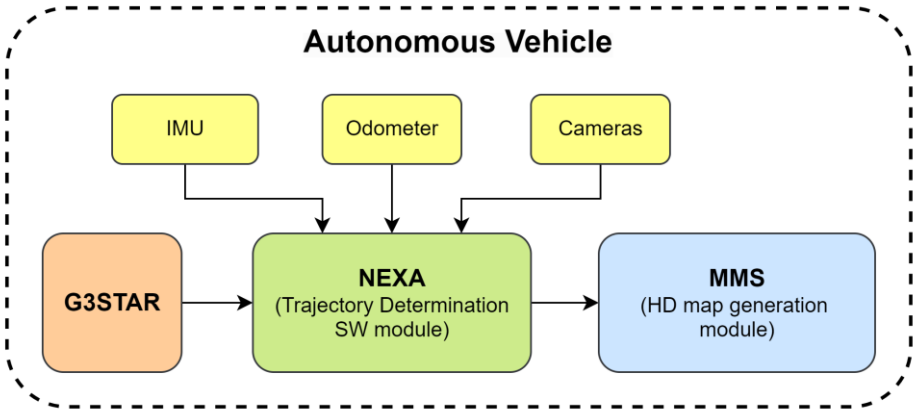

Figure 2. GAMMS navigation system schematic.

The G3STAR receiver will also feature a Chip-Scale Atomic Clock (CSAC) to improve its performance, the quality of its GNSS measurements and the Position, Velocity and Time (PVT) solution [14].
The GAMMS project is planned to finish at the end of July 2025. Until then, two test campaigns will take place, where the G3STAR receiver system integration will be tested, together with the other GAMMS systems and components, in different environments.

3. G3STAR RECEIVER IN GAMMS

The G3STAR receiver includes a version that is a space qualified GNSS receiver. The version used for the GAMMS project is a ground-based version of the space-qualified one, albeit using some less expensive components that are not required for a ground experiment. A couple of pictures of the receiver assembled for GAMMS is shown on Figure 3.

Figure 3. Ground version of the G3STAR receiver used in GAMMS.

The receiver hardware platform is built around a stack of heterogeneous boards. The HW Carrier Board provides mechanical and electrical support to all other boards, assembled with spacers and then enclosed in an aluminium case. The carrier boards allow to stack several boards:

- Radio Frequency Front End (RFFE) board with the GNSS RF Integrated Circuit (IC);
- RF Antenna Adapter board;
- A System on Module (SOM) daughter board built around the Xilinx/AMD Zynq UltraScale+ Multi-Processor System-on-Chip (MPSoC);
- DC/DC Power Adapter board contains the external connectors (power and communications interface).

The Radio Frequency module is based on a triple frequency front-end. The Digital Signal Processing (DSP) and the Software Defined Radio (SDR) receiver-based solution are implemented on a specific high-performance programmable SoM integrating all required functions with the possibility of in-field reconfiguration through a Xilinx/AMD Zynq UltraScale+ MPSoC. The

GNSS receiver is implemented in the MPSoC complex device, namely in the Field Programmable Gate Array (FPGA) fabric and in the Advanced RISC (Reduced Instruction Set Computer) Machine (ARM) dual-core Cortex-R5, whose Central Processing Unit (CPU) is designed for real-time applications. The receiver can accommodate a total of two RFFE that in turn can be connected to the respective RF Adapter board (connected to two antennas that can be selected individually or combined) effectively allowing to wire a total of four antennas and dynamically select two of them or a combination of the four.
Regarding the power interface, the G3STAR receiver includes an adapter board supporting unregulated power bus with a voltage range from 9 V to 40 V. The Carrier Board interfaces include the following communication interfaces:

- Universal Serial Bus 2.0 (USB);
- Recommended Standard RS-232;
- Recommended Standard RS-422;
- Controller Area Network (CAN) Bus;
- Joint Test Action Group (JTAG);
- Pulse-Per-Second (PPS) port;
- Ethernet Port.

The main elements of the GAMMS' G3STAR GNSS receiver system are detailed in the list below.

- The System-on-Chip (SoC) Design for FPGA includes the firmware components:
    - G3Star GNSS Receiver Firmware core;
    - HW Monitoring auxiliary modules and interfaces;
    - RFFE driver (ADC interface and configuration interface);
    - Other components for SoC: AXI bus components, ARM configuration, pinout definition, timing constraints, routing of signals, etc.
- The Application Space SW based on Linux OS distribution, that runs on the general-purpose ARM quad-core Cortex-A53 processor, and which has the following SW components:
    - System Management SW: Boot Control, Monitoring functions, RFFE drivers and control, Fault-Detection, Fault-Isolation and Recovery (FDIR) algorithms, G3 TC/TM and Space Protocol wrapper;
    - Support Applications for the G3Star platform (e.g. monitor application);
    - OSNMA processor module;
    - Interface SW for NEXA (trajectory determination software module).
- The G3Star Receiver Software (SW) core from Deimos, deployed in a dedicated ARM Cortex-R5 core, which boots after the Application Space SW.
- The interprocessor messaging protocol, based on the OpenAMP library, which mediates the communications between the two ARM cores using a shared memory.

A system model of the receiver is depicted on Figure 4. The triple-band RFFE captures incoming signals that are processed by the G3STAR Input Module and are assigned to GNSS channels. FFT acquisition and tracking is performed, for the On-Board (OB) software modules to obtain the GNSS measurements and navigation data, required for a PVT solution. The OB SW includes the processing of the Galileo HAS and OSNMA data, with dedicated modules. An interface with NEXA is needed in order to communicate the results from the receiver to NEXA, using agreed formats and protocols.
Despite its modifications from the space-based version, the receiver's qualifications should be more than enough to withstand the environment conditions of the AV.
The receiver can be used as a standalone sensor, generating and storing its results for post-processing, or it can be integrated with a Navigation Software and provide its outputs in real-time. In GAMMS, as previously stated, two field tests will be conducted. The first test will be a preliminary run, with the purpose of testing the integration of the different systems and sensors, and will serve as a preparation for the second field test. This will allow the consortium partners to identify any issues present in the first field test and have the opportunity to correct them before the second field test. The G3STAR team from Deimos will assess the physical integration of the receiver with the other systems, namely the GNSS antenna, the CSAC and the Navigation Computer. The team will also analyse the GNSS receiver regarding the quality of its measurements and PVT solution, in different environments such as open-sky and urban environment, by having a commercial receiver as a reference. The influence of multipath caused by signal reflections on obstacles in urban environments and the impact of the CSAC on the receiver's results will also be examined. Finally, the results obtained from the G3STAR receiver, including HAS and OSNMA, combined with the outputs from the other sensors, will be used by the Navigation Algorithm, and the overall GAMMS solution will be evaluated.

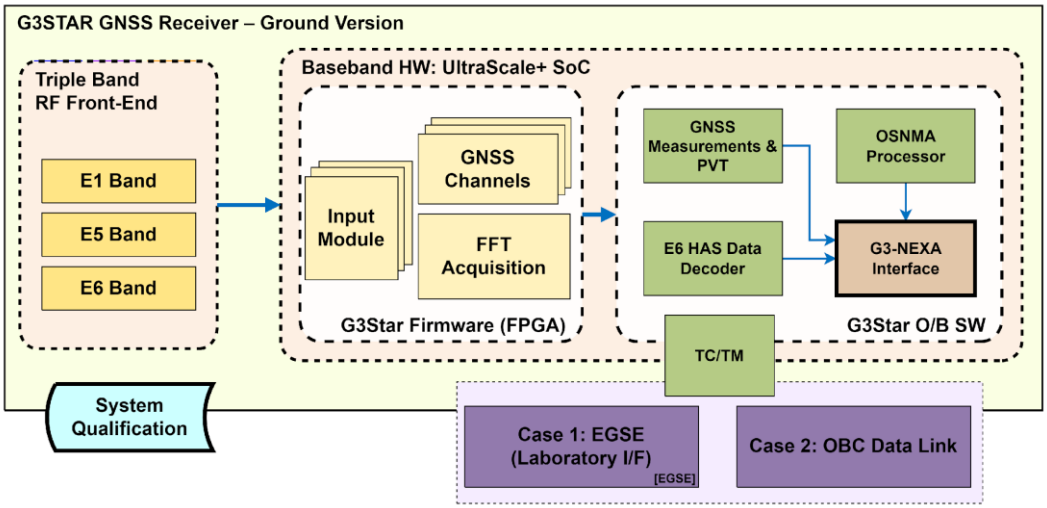

Figure 4. GAMMS G3STAR receiver System Model and Use Cases.

The issues identified from the first field test will then be solved or minimised before the second field test. In this test, the receiver is expected to operate in real-time with the Navigation Algorithm running on the Navigation Computer, i.e. the receiver will provide its measurements from tracking E1 and E5 signals, the acquired navigation data from E1 I/NAV, its PVT solution, and the demodulated and decoded HAS corrections, through a real-time interface. OSNMA data processing will be used by the receiver to filter out signals that broadcast navigation data that fails to authenticate.

3.1. HAS Decoding

The G3STAR receiver has the capability of processing Galileo E6 signals. The symbols obtained after demodulating the E6 signals are processed according to the Galileo ICD for the Signal-in-Space [6]. Similarly to other Galileo data signals, the 492 E6 data bits of each E6 C/NAV page are encoded with a Forward Error Correction (FEC) convolutional encoding. They are then interleaved with a block interleaver, which for E6 data has dimensions of 123 x 8 (total of 984 symbols). To recover the data bits from the broadcast interleaved symbols, the receiver features a de-interleaver module and a Viterbi algorithm [15], to perform the decoding of the symbols, recovering the original 492 bits of the page. Each C/NAV page includes a HAS page and a CRC checksum (Figure 5). The receiver computes the same checksum and compares it with the one provided by the C/NAV page, to ensure a successful reception and demodulation.

C/NAV Page				Total (bits)
Reserved	HAS Page	CRC	Tail	
14	448	24	6	492

Figure 5. E6 C/NAV Page Layout (taken from the Galileo HAS SIS ICD [6]).

Unlike other Galileo navigation messages, HAS data is not immediately ready for processing. Different satellite vehicles broadcast different E6 C/NAV pages, all contributing to the same HAS message, which is composed of multiple HAS pages. The HAS message is further encoded through an outer layer scheme called High Parity Vertical Reed-Solomon (HPVRS) [16]. This encoding scheme allows the receiver to be able to reconstruct a full HAS message with a length of N pages, by obtaining any N different HAS pages from the signal-in-space. This speeds up the reception of the full message, because it avoids the receiver needing to wait for specific missing pages.

The G3STAR receiver includes a software module dedicated to decoding the Reed-Solomon encoded HAS messages from the corresponding number of different HAS pages. The Reed-Solomon algorithm used is described in [6]. The decoder used by the receiver uses the Reed-Solomon Generator Matrix to reverse the encoding operation applied to the data and attain the fully decoded HAS message.

Once the decoded HAS message is obtained, its raw bits are assigned to their corresponding parameters, according to the Galileo HAS ICD. These parameters can be outputted in the standard RTCM format and used for processing by third parties. In GAMMS, these HAS parameters will be provided to the Navigation Algorithm (NEXA) during the field tests, to compute a Precise Point Positioning for the AV's trajectory. Figure 6 depicts a diagram of the main operations involving the HAS data processing.

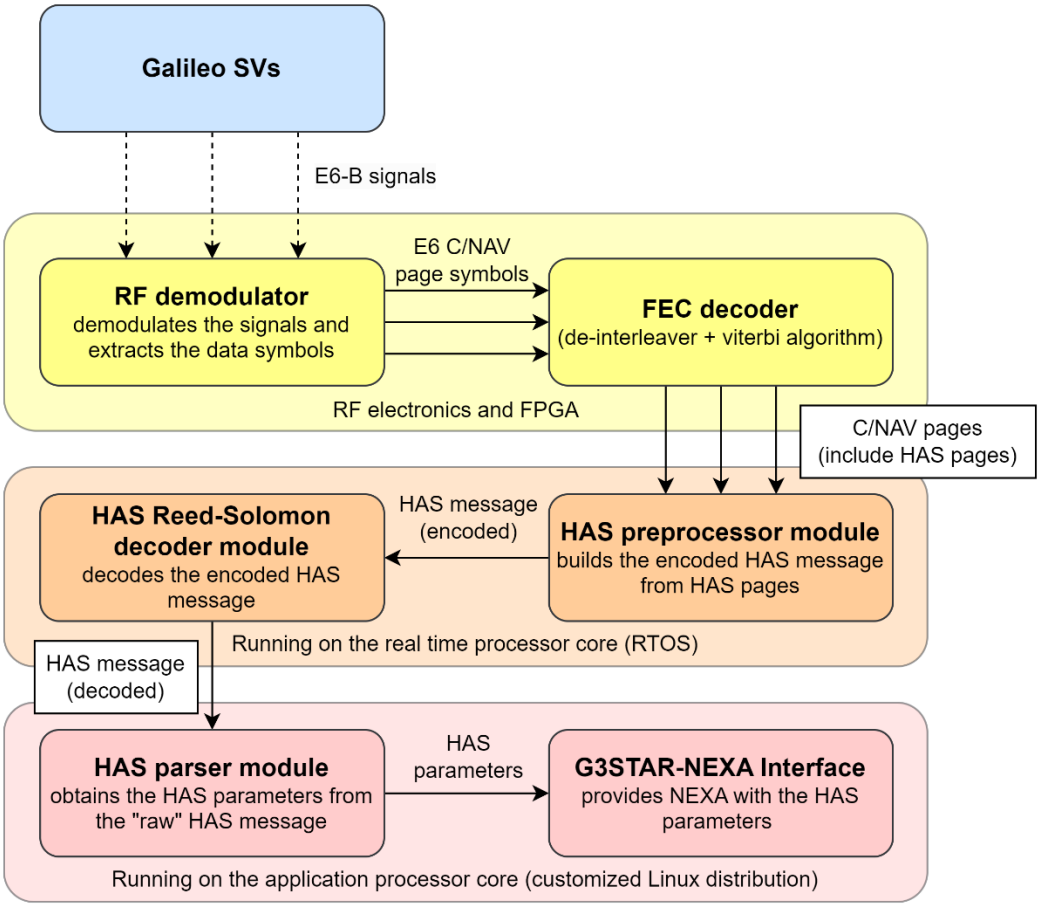

Figure 6. Logical implementation of HAS on G3STAR.

3.2. OSNMA Processing

The G3STAR receiver will feature a dedicated module to process the OSNMA bits broadcast on the I/NAV message. The authentication of the navigation data via OSNMA is made with the use of several cryptographic operations. The OSNMA module uses the Federal Information Processing Standard (FIPS) certified wolfSSL library *wolfCrypt Crypto Engine* to run the required cryptographic operations [17].

OSNMA is broadcast within the E1-B I/NAV navigation message as shown on Figure 7, where it can be seen that the OSNMA bits are divided into two sections: HKROOT (Header and Root key) - 8 bits per nominal page - and MACK (MAC and Key) - 32 bits per nominal page. The HKROOT stands for Header and KROOT, where KROOT is the term used to refer to the Root Key of a TESLA chain. The HKROOT bits form a block of data every sub-frame, depicted on Figure 8.

The first nominal page carries the information regarding the NMA Header. The remaining pages include what is called a Digital Signature Message (DSM) block. These DSM blocks contain the information that form the basis of the authentication scheme of the OSNMA: the Digital Signature. Together, they can form the DSM-KROOT message, which can consist of up to 14 DSM blocks (1456 bits). They can also contain information regarding the Public Key provision, by building the Public Key Renewal (PKR) message: DSM-PKR.

The several DSM Blocks that make up a DSM-KROOT or DSM-PKR are broadcast in parallel by several satellites. This considerably speeds up the reception of the full DSM message.

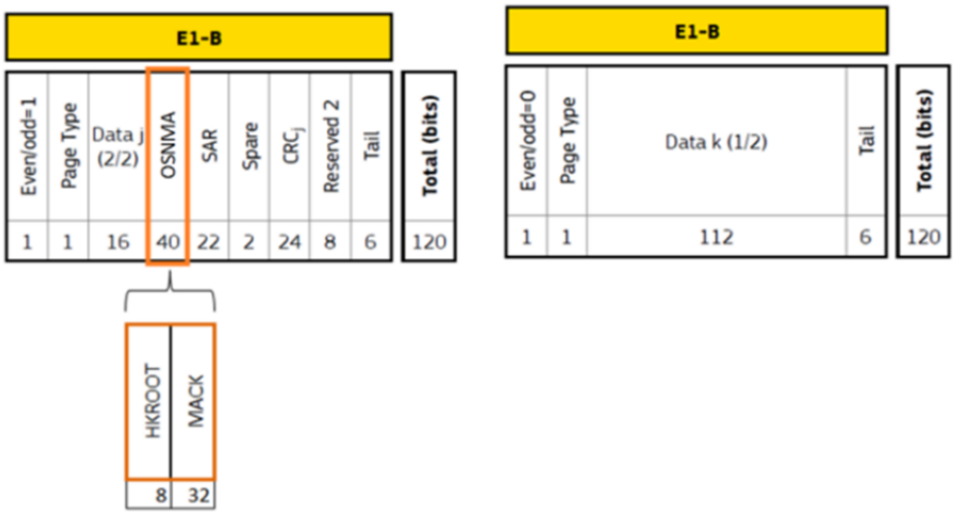

Figure 7. OSNMA Broadcasting scheme. Taken from [7].

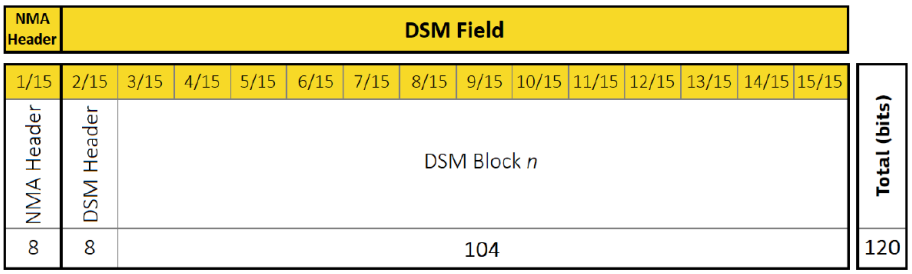

Figure 8. OSNMA DSM Field schematic. Taken from [7].

The DSM-KROOT message includes the general configurations of the OSNMA (including which cryptographic functions to be used) and two very important parameters: the KROOT and the Digital Signature. The KROOT is the "Root Key", a key that is the base of the TESLA chain. The TESLA protocol is based on a chain of keys that are generated by each other. An initial TESLA key is generated. Then, a hash function [18] is applied to this key, concatenated with time-sensitive parameters, generating a new string, which is the next key of the chain. This new key is also hashed with time-sensitive parameters, and so on and so forth. After N iterations, N+1 keys have been generated. Due to the nature of the hash functions, these keys are unilaterally related, in the sense that it is trivial to generate a chain in one direction, but not in the opposite direction.

It is a convention that the last key in a chain to be generated is considered the key 0 (or K_0, the chain's root key). The first key to be generated is key N (K_N). This way, a user with any given key can recreate the whole sub-chain that is lower than the index of that key, but cannot easily recreate the keys higher up in the chain. Therefore, if the user knows that the root key is authentic, it can take any key of index *n* and verify it by hashing it recursively *n* times, until K_0 is computed and compared with the authentic root key. If the root keys match, the key being verified is authentic.

The KROOT from OSNMA is signed, so the user needs to take the signature transmitted by the DSM-KROOT and use a specified ECDSA to authenticate the DSM-KROOT parameters, which include KROOT.

The MACK section of OSNMA is transmitted using 32 of the 40 OSNMA bits per nominal page of each I/NAV message. Unlike HKROOT which transmits single messages with contributions from all SVs in view of the receiver, the contents of the MACK section are more specific to each SV. MACK stands for MAC (Message Authentication Codes) and Key. They include tags, which are truncated MACs, that are used to validate I/NAV data fields from the transmitting SV and of other Galileo SVs. These tags are verified with the use of their corresponding TESLA key, which is also transmitted by the SVs.

This ensemble of tags and keys makes up the TESLA protocol. Each TESLA key is verified by hashing the previous key in the chain, concatenated with its time of transmission and an unpredictable chain pattern provided by DSM-KROOT, recursively *n* times, until K_0 is reached. This result is compared with KROOT (previously authenticated), and if they match, the TESLA key received is valid. The inclusion of the times of transmission on the validation of the TESLA keys makes the process time-sensitive. The hash functions used by the OSNMA protocol to generate and validate the TESLA chain keys are the SHA-256 and SHA3-256 hash algorithms [19] [20].

The MACK section can only be used after the DSM-KROOT message has been received and KROOT has been validated via Digital Signature. Tags are verified by taking the I/NAV data and the corresponding (authenticated) TESLA key: if the result of the MAC operation that takes as input the data and the key matches the tag received, the tag is considered valid. The OSNMA protocol has a requirement that a given number of tag bits associated with each navigation data set need to be validated, for the navigation data to be considered valid. Therefore, after enough tag bits associated with a navigation data set are successfully verified, the data is considered valid. A schematic detailing the OSNMA tag validation is depicted in Figure 9.

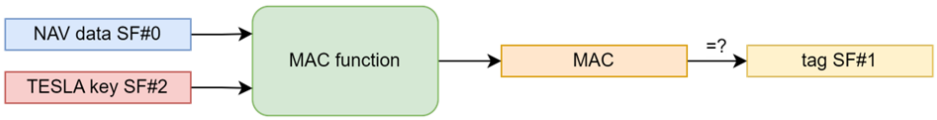

Figure 9. Tag authentication scheme.

There is a time delay on the TESLA protocol between the broadcasting of the tag and its corresponding key. This is what ensures the security of the protocol, since the keys are openly provided. If the tag and associated key were broadcast at the same time, a spoofer could retrieve the TESLA key, falsify a navigation message, create a valid tag with the correct key, and transmit the falsified message that would be validated by the receiver. Thus, OSNMA imposes a delay of one I/NAV sub-frame (30 seconds) between the tag and corresponding TESLA key. This way, the receiver has time to retrieve a tag while the key that generated it is secret. Then, when the secret key is openly broadcast, potential spoofers no longer have the ability to forge a verifiable tag, since that tag should have been broadcast on the previous sub-frame, and receivers will not associate it with its corresponding TESLA key.
The G3STAR receiver will base its authentication of the Galileo signals on the authentication of the Galileo I/NAV Ephemeris, Clock and Status (ECS) data bits, which consist on 549 bits of selected data from I/NAV Word Types 1 to 5, authenticated by specific tags which are defined by the OSNMA protocol. The specific bits of this selected data are stated on the annex B of the OSNMA SiS Interface Control Document [7]. Thus, the nominal operation of the OSNMA module of G3STAR, in its simplest form, is the following:

1. OSNMA data starts to be processed - DSM-KROOT message is received from all SVs in view (broadcasting OSNMA) and assembled;
2. The contents of DSM-KROOT are authenticated via Digital Signature (ECDSA), using the Public Key in force (stored on the receiver);
3. The MACK section bits of each SV, and corresponding I/NAV ECS data to authenticate, are stored: from each I/NAV sub-frame (which consists of 15 I/NAV nominal pages) a set of tags and a TESLA key are retrieved, as well as the specific data bits from the navigation message that will be authenticated;
4. Tags are associated with the I/NAV data of the previous sub-frame and the TESLA key is associated with the I/NAV data (and corresponding tag) of two sub-frames prior, i.e., after processing sub-frame N, the tag received corresponds to the data from sub-frame N-1 and the TESLA key received corresponds to data from sub-frame N-2 (so I/NAV data from sub-frame N-2 can be authenticated using the tag received from sub-frame N-1 and the TESLA key received from sub-frame N);
5. The TESLA key is authenticated using the trusted KROOT, verified on step 2;
6. The ECS navigation data is used, together with the corresponding TESLA key, to compute a tag;
7. The received tag, associated with that navigation data and TESLA key, is compared with the computed tag of step 6: if they match, the tag is verified;
8. The tag validation process is repeated until enough tags associated with the navigation data are verified, at which point the data is considered valid.

The diagram shown on Figure 10 provides a visual guide on how ECS navigation data, TESLA keys and tags are associated. This ignores added complexities as with the data Cut-Off Point (COP) field of the tags, explained on the OSNMA ICD [7]. It is the simplest case of self-authentication (the data, tag and key are transmitted by the same SV and are associated with each other). TESLA keys are common to all tags and data of a given sub-frame, i.e. all SVs will broadcast the same key at the same epoch. There can be cross-authentication, where the tag transmitted by a given SV is associated with data from another SV (the same delays apply). There is also the case of "slow MACs", which are tags that are associated with TESLA keys that have a larger delay than normal: instead of being authenticated by TESLA keys broadcast on the next sub-frame, they are authenticated by TESLA keys broadcast 11 sub-frames in the future (so the delay is extended by 10 sub-frames, or 300 seconds).

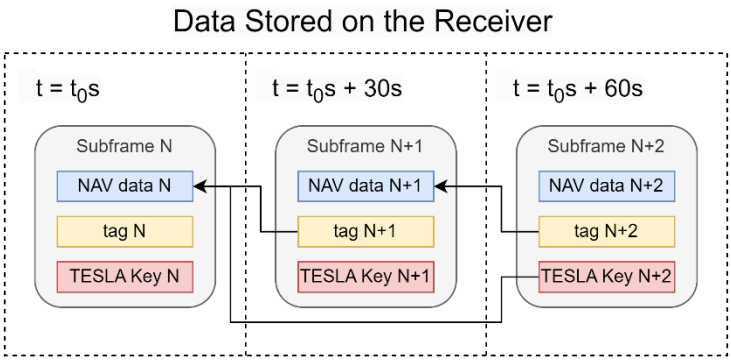

Figure 10. Simplest case of navigation data, tag and key association.

The OSNMA protocol also allows the reception of Public Keys, used to verify the Digital Signatures received. These keys may expire, or become compromised, in which case the receiver needs to be updated with a new Public Key. This is done via the DSM-PKR messages and the use of a Merkle Tree [21] to verify the broadcast Public Key. The details about the authentication of the new Public Keys are described on the OSNMA ICD [7] and guidelines [22], and will not be further detailed in this paper. Figure 11 provides a graphical overview of the implementation of OSNMA on G3STAR.

The relevant "unit" for OSNMA processing is the I/NAV sub-frame. Each sub-frame from each SV contains one DSM block (and headers), a set of tags and a TESLA key, as well as the Ephemeris, Clock and Status fields that are authenticated. The receiver stores the received navigation data and tags on dedicated buffers, and associates each tag to its corresponding navigation message and TESLA key. The navigation data buffer also includes a counter for the number of authenticated tag bits associated with them. When the TESLA keys received are verified, those are stored on a buffer as well, and may be used to generate the tags from the navigation data they are associated with, to validate the received tags on the tag buffer. When tags are verified, they are removed from the tag buffer, and the counter of valid tag bits of their corresponding navigation data is incremented by the number of bits the tag has. When that counter reaches the determined number of valid tag bits required to authenticate its respective navigation data, an output is produced, where the navigation data and satellite providing it are validated, and the data is deleted from the NAV buffer. The output is used by the receiver to filter data and signals that fail to authenticate.

The receiver has a stored Public Key, which is loaded into memory upon start. If a new Public Key is provided via DSM-PKR, the receiver stores it if its authentication is successful, to be loaded in future boots.

The G3STAR receiver will process the OSNMA data as explained, to authenticate the ECS data of the satellites being tracked. If the authentication of the ECS data obtained from a given E1-B signal fails, the full navigation message from which that data was taken is discarded. Similarly, the measurements from that signal and its associated E5 signal (if also being tracked) are discarded, and not provided to the navigation algorithm that computes the navigation solution.

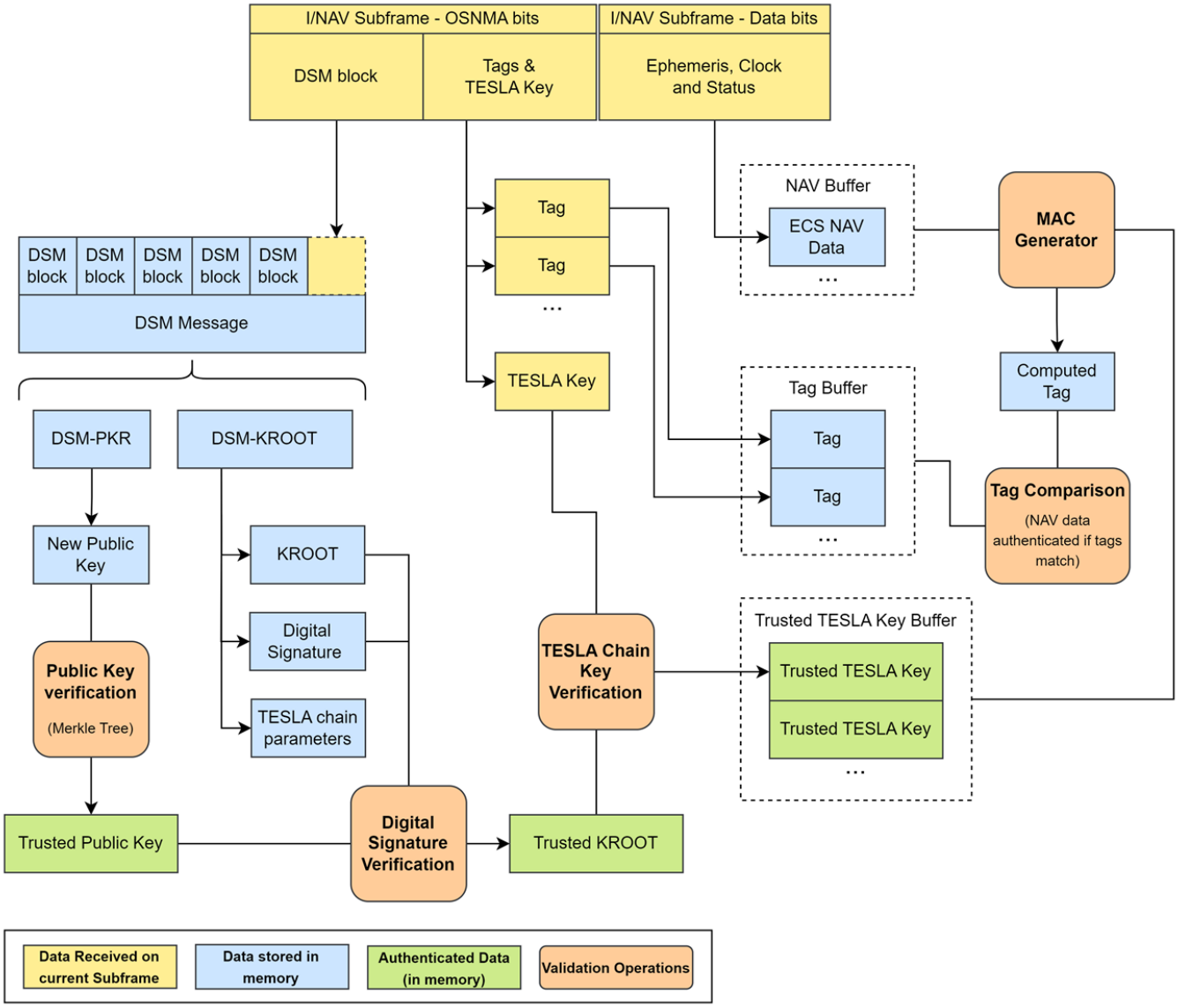

Figure 11. OSNMA logic implementation on the G3STAR receiver.

4. PRELIMINARY RESULTS

4.1. Acquisition of HAS Corrections

The receiver is currently able to process E6 signals and obtain the HAS data and save it in a file, in RTCM format. This process still requires validation from a navigation algorithm. So, preliminary results feature only the RTCM files obtained from testing the receiver.

To test the receiver, Skydel, an RF Constellation Simulator (RFCS), is used [23]. This tool simulates the GNSS signals and navigation data from the Galileo constellation, which are transmitted to the receiver for real-time processing. One of the features of this simulation tool is the possibility to add errors to the simulation, namely pseudorange offsets, pseudorange errors and ephemeris errors. By adding these errors to the simulation, Skydel generates HAS messages that can be used to correct them.

By default, Skydel does not add any errors to the simulation, so the HAS messages generated and provided contain corrections that are equal to '0', i.e. there are no corrections available. These messages nonetheless require the necessary number of different HAS pages to be received, grouped in single HAS messages and decoded using the Reed-Solomon decoder. The G3STAR receiver correctly processes these and generates "empty" RTCM files, without corrections, since all '0' fields are suppressed, as shown on Figure 12.

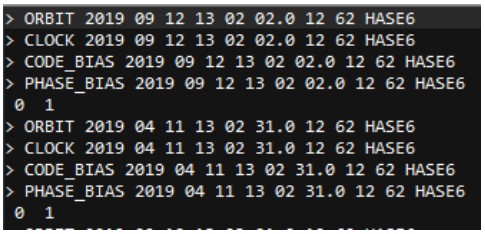

Figure 12. Excerpt from an RTCM file generated by G3STAR processing Skydel simulated signals without errors added.

When errors are provided to Skydel non-zero HAS corrections are generated and transmitted via the simulated E6 signals. To test this, a Skydel scenario containing errors on Galileo SVIDs 1, 2, 3, 4 and 5 was run. The G3STAR receiver successfully obtained and processed the HAS data consequently generated, which is shown on Figure 13.

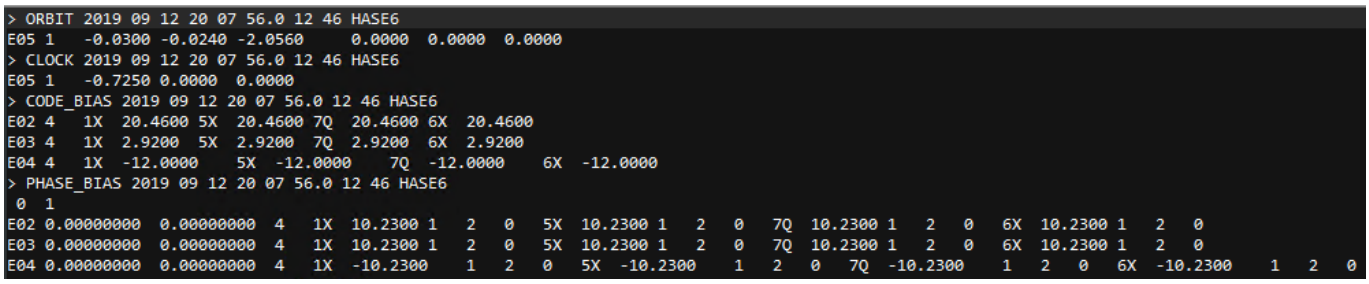

Figure 13. Excerpt from an RTCM file generated by G3STAR processing Skydel simulated signals with errors added.

For the purposes of this test, the errors added to the simulation were arbitrary, and their impact on the navigation solution not assessed. This is because only the HAS data processing capability has been analysed as of yet. Once the integration of the receiver with the GAMMS Navigation Algorithm is completed, these errors may be assessed, and the navigation solution without HAS corrections may be compared with the navigation solution with HAS corrections.

4.2. Processing of OSNMA

Regarding the OSNMA data processing, the OSNMA module used on the G3STAR receiver was developed during the early stages of the service definition, which evolved through some modifications until the present day. Therefore, some updates are being done on this module, in the course of the GAMMS project, for it to be fully functional.

Currently, the ECS (Ephemeris, Clock and Status) data can be processed by the module, meaning that the Digital Signature of the HKROOT can be successfully verified and the TESLA keys and tags relating with the ECS data can be authenticated. To assess whether the data is validated, visual inspection of the authentication logs is required.

A live test was done on December 1st, 2023, using a stationary antenna placed at the top of Deimos' office building. On this day, at 14:00 UTC a TESLA Chain renewal set up by GSC (European GNSS Service Centre) took place. The results of the standard OSNMA operation are shown on Figure 14.

4.2.1. OSNMA standard operation

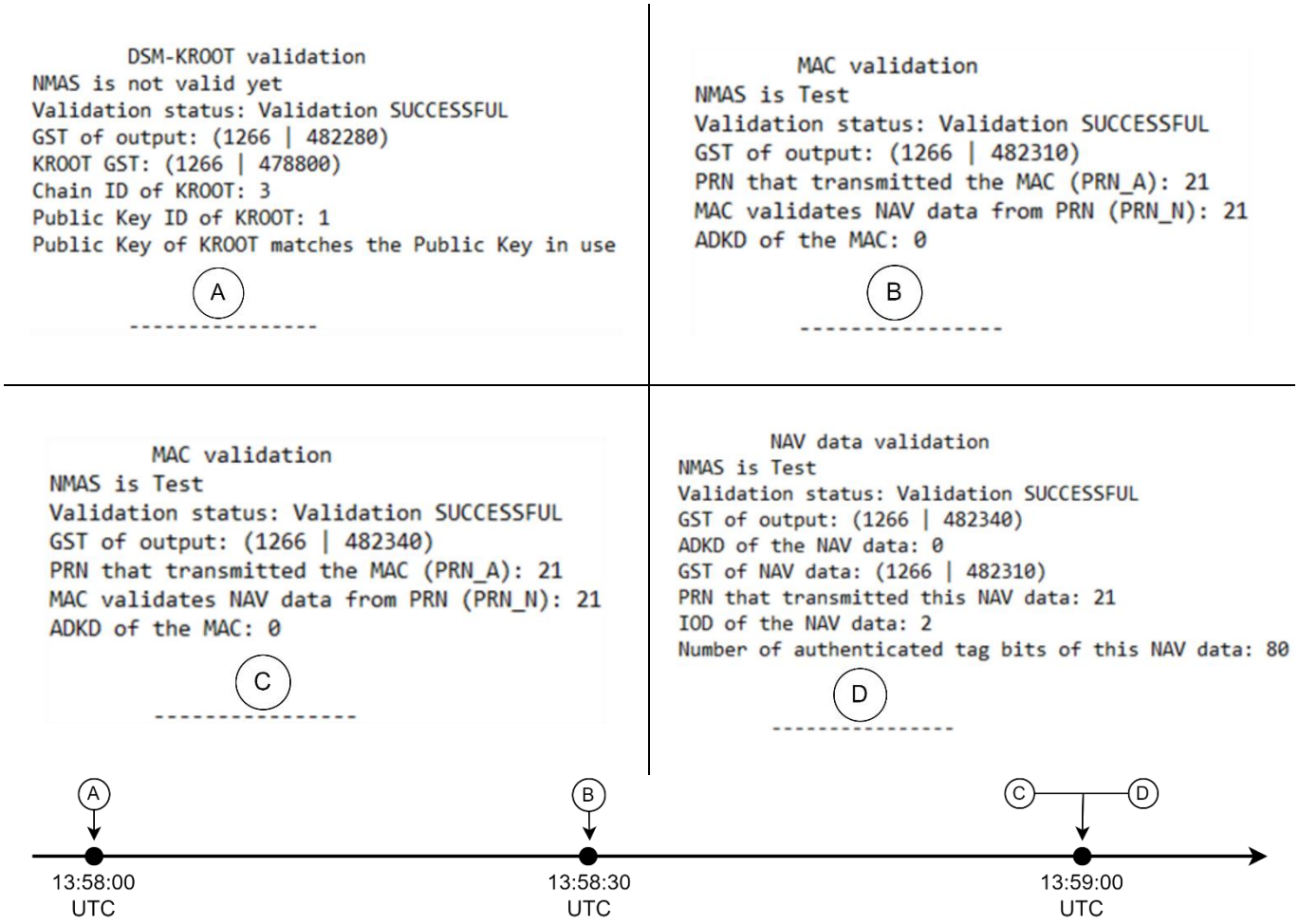

Figure 14. OSNMA validation of ECS navigation data. The prints were generated sequentially from left to right, top to bottom.

The timings used on OSNMA are based on I/NAV sub-frames, i.e. the GST (Galileo System Time) time tags associated with the OSNMA parameters typically correspond to the beginning of the sub-frame that transmitted those parameters. This is reflected on the OSNMA module logs, which also use the same convention. Thus, the "GST of output", and other time tags, are the time of the starting of the sub-frame that was last processed. Therefore, the time at which each OSNMA computation took place is the "time tag" plus 30 seconds (a full sub-frame is received before its data is processed).

The print on the top-left side of Figure 14 (A) shows the output from the Digital Signature authentication of DSM-KROOT, at 2023/12/01 13:58:00 UTC ("GST of output"). The TESLA root key has a GST time tag that corresponds to 2023/12/01 13:00:00 UTC ("KROOT GST"). This means that the TESLA keys received need to be authenticated by computing the TESLA chain until KROOT is reached. Since a new TESLA key is provided for each sub-frame (30s), and the time difference between the *GST of output* and *KROOT GST* is 3480 seconds, the TESLA key received at the time of the DSM-KROOT validation was key number 117 (it's not 116 because it's a convention that KROOT is not used for MAC computation, so it's actually associated with the sub-frame that comes before "KROOT GST"). So the receiver computed all 117 keys and compared the result with the broadcast KROOT. To avoid receivers needing to eventually compute thousands or millions of keys, *floating KROOTs* are provided by the service, which are keys that are part of the chain, not so distant in the past, authenticated with the DSM-KROOT message. The receiver also optimises the TESLA key validation by storing the last validated key and using it to validate any new keys received.

At this stage, the receiver has stored the navigation data from the previous sub-frame, and the TESLA key from the current sub-frame. So the tags from the next sub-frame will be used to validate the navigation data, using the TESLA key.

This is shown on the log on the top-right side of Figure 14 (B). The output is from 2023/12/01 13:58:30 UTC and shows the self-authentication of a tag by Galileo SVID 21. The *ADKD of the MAC* (tag) being '0' means that the data bits being validated are the Galileo I/NAV Ephemeris, Clock and Status (ECS) data bits.
As previously stated, the OSNMA protocol requires that a given number of tag bits associated with each navigation data set need to be validated for the navigation data to be considered valid. In this case, it's a total of 80 bits, and the tags have a length of 40 bits each – so two different tags need to be verified. In this test, only one SV broadcasting OSNMA is tracked, so it has to self-authenticate. The next self-authentication tag is received on the next sub-frame, at 2023/12/01 13:59:00 UTC and its validation log is shown on the bottom left side of Figure 14 (C). Since at this point all 80 tag bits had been successfully authenticated, the navigation data is considered genuine, shown on the bottom-right side of Figure 14 (D).

4.2.2. OSNMA chain renewal (EOC)

At 14:00 UTC, the TESLA chain was renewed, in a process named "End Of Chain" (EOC). This means that the TESLA keys transmitted after that point are no longer related with the keys from the previous chain. A new KROOT is provided, and the OSNMA parameters transmitted by the DSM-KROOT message were changed. The service also increments the value of the chain ID field on the NMA header, to indicate the new chain in force.
Shortly before a chain renewal process, DSM-KROOT messages for both the old and the new chains are provided. The receiver can use the DSM-KROOT messages of the old chain to continue using it, and validate and store the configuration and parameters of the new chain, which will become active at the indicated time. In this test, the receiver could not acquire the DSM-KROOT concerning the new chain before the end of the old chain. This was most likely due to lack of time: when the receiver started to receive OSNMA bits, it could only receive the DSM-KROOT message of the previous chain, which was authenticated at 13:58:30 (the GST of the output corresponds to the beginning of the sub-frame, but the computations take place at the end). Thus, when the new chain started to be used, the receiver noted that the chain ID provided by the NMA header did not match the chain ID of the KROOT it had previously stored and authenticated (which in this case is *Chain ID=3*, as seen on the top left corner of Figure 14), from the previous chain. Because of this, the receiver flagged the occurrence as an error, and did a soft-reset of the OSNMA module. Afterwards, it started acquiring OSNMA bits as normal, until a DSM-KROOT message concerning the new chain was received and authenticated. This output is shown on Figure 15.

```
        DSM-KROOT validation
NMAS is not valid yet
Validation status: Validation SUCCESSFUL
GST of output: (1266 | 482610)
KROOT GST: (1266 | 482400)
Chain ID of KROOT: 0
Public Key ID of KROOT: 1
Public Key of KROOT matches the Public Key in use


----------------
```

Figure 15. Authentication of the DSM-KROOT message from the new chain.

The GST associated with this output corresponds to 2023/12/01 at 14:03:30 UTC. It can also be seen that "KROOT GST" corresponds to 14:00:00 UTC – the time at which the new chain was enforced. The chain ID of the new KROOT is now 0 (Figure 15), as opposed to the previous value of 3, from the KROOT of the old chain (top left corner of Figure 14).

4.2.3. OSNMA public key revocation (PKREV)

Similarly to the TESLA chain, the Public Key in force, used to authenticate the Digital Signatures received with DSM-KROOT, can also be changed by the service. GSC did a "Public Key Revocation" exercise, analogous to the chain renewal, on December 14th, 2023. The G3STAR receiver was tested on its ability to handle a renewal of the Public Key, with the live signals broadcast by Galileo satellites, using Deimos' stationary antenna.
When the receiver started to obtain OSNMA bits, it obtained the DSM-KROOT message, however, the Public Key ID (PKID) that authenticates the message received did not match the ID of the Public Key stored on the receiver (which was revoked). This meant that the receiver had to wait for the reception of the new Public Key, in order to authenticate the DSM-KROOT messages. This is shown on Figure 16, where it can be seen that the receiver flags the validation with a warning, and added a print stating that the "Public Key might be expired", due to the mismatch of the Public Key IDs (the previous PKID was '1' and the PKID associated with KROOT is '2').

```
        DSM-KROOT validation
NMAS is not valid yet
Validation status: There were PROBLEMS with the validation
GST of output: (1268 | 301920)
KROOT GST: (1268 | 306000)
Chain ID of KROOT: 1
Public Key ID of KROOT: 2
Public Key might be expired


----------------
```

Figure 16. DSM-KROOT validation without the matching Public Key.

This output has a time tag associated of 2023/12/13 11:52:00 UTC. Please note that the time tag associated with KROOT is 2023/12/13 13:00:00 UTC, which means that this TESLA chain is only starting to be applicable at that time. In parallel with DSM-KROOT messages, the service also sent DSM-PKR messages, which allow the renewal of the Public Key in force. The validation of the DSM-PKR is described in the OSNMA ICD [7] and will not be detailed here, since this functionality is not critical for the OSNMA operation in short time frames (the Public Key can be obtained on the GSC OSNMA server and updated manually on the receiver). Nonetheless, it is a good feature to have implemented, in case of an emergency (and therefore unexpected) Public Key revocation. The output print of the authentication of the new Public Key is shown on Figure 17.

```
        DSM-PKR validation
NMAS is not valid yet
Validation status: Validation SUCCESSFUL
GST of output: (1268 | 301950)
New Public Key ID: 2
New Public Key Type: ECDSA P-256
New Public Key (33 bytes): [redacted]

----------------
```

Figure 17. DSM-PKR validation.

The new Public Key (redacted on the log print) was authenticated successfully and stored on the receiver, at 2023/12/13 11:52:30 UTC. Its PKID is '2', which is the one that verifies the DSM-KROOT messages being received. Thus, DSM-KROOT messages received afterwards can be authenticated, as shown on Figure 18.

```
        DSM-KROOT validation
NMAS is set to Don't use
Validation status: Validation SUCCESSFUL
GST of output: (1268 | 305700)
KROOT GST: (1268 | 306000)
Chain ID of KROOT: 1
Public Key ID of KROOT: 2
Public Key of KROOT matches the Public Key in use


----------------
```

Figure 18. DSM-KROOT authentication with the new Public Key.

This validation, with associated time tag at 2023/12/13 12:55:00, shows that the new Public Key obtained via the signal-in-space can be used for OSNMA authentication. The system is flagging the service as "Don't use", because the revocation of Public Keys is associated with the service being compromised, thus the need to change the keys used in the Digital Signature. In this situation, no other OSNMA bits are processed (TESLA keys and tags) until the system flags the OSNMA as "Operational" (in nominal conditions) or "Test".

5. CONCLUSIONS

The GAMMS consortium is developing a mapping robot to create HD navigation maps for AVs, using cutting-edge technology, including the G3STAR GNSS receiver. This receiver, developed for this project, will make use of the open service features of Galileo to provide high quality GNSS data. Preliminary tests were made, specifically targeting the receiver's capability of processing the Galileo data from the new services: HAS and OSNMA. The results obtained were quite good and prove that the G3STAR receiver is capable of making use of those services. However, the receiver has not yet been tested as part of the full GAMMS experiment. The HAS data was successfully obtained, but validation from a navigation software that uses it to enhance the accuracy of its navigation solution is still required, to assess its impact. The OSNMA validation was also successfully performed in the live tests reported in this paper, in laboratory conditions. Further field testing is needed to assess the availability of the service in different environments, including urban areas. The impact of the CSAC on the receiver's performance also needs to be evaluated with tests on different environments. Lastly, the overall outcome of employing the G3STAR receiver in the navigation system of GAMMS, and consequently on the HD map generation, should be compared with that of a commercial off-the-shelf receiver. All of the aforementioned tests and analysis will be performed during the two test campaigns planned in the scope of GAMMS project.

ACKNOWLEDGEMENTS

Some of the work detailed in this paper was performed within the scope of the GAMMS project, which has received funding from EUSPA through the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004255. The authors of this paper would like to thank EUSPA and the European Commission for the funding of the project and all the GAMMS project consortium partners: GeoNumerics, GEOSAT, ENIDE, EPFL, Pildo Labs, Virtual Vehicle and Solid Potato. Without them, this project would not have been possible. The authors also thank all the team members of Deimos who directly or indirectly contributed to the work developed, as well as the support staff who made that development possible.

REFERENCES

[1] R. Liu, J. Wang, and B. Zhang, "High Definition Map for Automated Driving: Overview and Analysis", *Journal of Navigation*, vol. 73, no. 2, pp. 324-341, 2020.

[2] SAE International, "J3016_202104: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles", 2021.

[3] Elecnor Deimos [Online], "G3STAR – GNSS Receiver for New Space", https://elecnor-deimos.com/wp-content/uploads/2024/05/Product-G3Star_1405.pdf, 2024.

[4] GAMMS. [Online], "GAMMS - Robots mapping for robots", gamms.eu, 2024.

[5] European Union Agency for the Space Programme, "Galileo High Accuracy Service (HAS) Info Note", 2020.

[6] European Commission, "Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD)", Issue 1.0, May 2022.

[7] European Commission, "Galileo Open Service Navigation Message Authentication (OSNMA) Signal-In-Space Interface Control Document (SIS ICD)", Issue 1.1, October 2023.

[8] National Institute of Standards and Technology, "FIPS PUB 186-5 - Digital Signature Standard (DSS)," U.S. Department of Commerce, 2023.

[9] A. Perrig, D. Song, R. Canetti, J. Tygar, B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.

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

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

[12] European Union Agency for the Space Programme, “Galileo Open Service Navigation Message Authentication (OSNMA) Info Note”, 2021.

[13] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," in IEEE Transactions on Automatic Control, vol. 62, no. 4, pp. 1797-1812, April 2017.

[14] E. Fernández, D. Calero, M.E. Parés, CSAC Characterization and Its Impact on GNSS Clock Augmentation Performance”, Sensors, vol. 17, issue 2, no. 370, 2017.

[15] J. G. Proakis, M. Salehi, “Digital Communications”, 5th edition, McGraw-Hill, 2008.

[16] I. Fernández-Hernández, T. Senni, D. Borio, G. Vecchione, “High-parity vertical Reed-Solomon codes for long GNSS high-accuracy messages”, Journal of the Institute of Navigation, June 2020, vol. 67, issue 2, pp. 365-378.

[17] wolfSSL. [Online], “wolfSSL Embedded SSL/TLS Library | Products – wolfSSL”, https://www.wolfssl.com/products/wolfssl/, 2024.

[18] B. Preneel, “Cryptographic hash functions”, European Transactions on Telecommunications, vol. 5, issue 4, pp. 431–448, 1994.

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

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

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

[22] European Commission, “Galileo Open Service Navigation Message Authentication (OSNMA) Receiver Guidelines”, Issue 1.3, January 2024.

[23] NOFFZ Technologies GmbH. [Online], “GNSS Simulator sUTP 5017”, https://www.noffz.com/en/products-components/gnss-simulator/, 2024.