Galileo Signal Authentication Service (SAS)

I. Fernandez-Hernandez[*], J. Winkel[†], C. O'Driscoll[‡], G. Caparra[§], R. Terris-Gallego[⁑], J. A. López-Salcedo[⁑], G. Seco-Granados[⁑], T. Willems[††], B. Motella[*], D. Blonski[§], J. de Blas[‡‡]
[*]*EC,* [†]*Independent Consultant,* [‡]*CODC,* [§]*ESA,* [⁑]*UAB,* [††]*CGI,* [‡‡]*EUSPA*

BIOGRAPHIES

Ignacio Fernandez is responsible for authentication and high accuracy services at the European Commission, DG DEFIS, GNSS Unit. He is an ICAI engineer and has a PhD in Electronic Systems from Aalborg university.

Jón Winkel received a Diploma in Physics from the University in Hamburg and a PhD from the university FAF Munich. He has over 25 years' experience in GNSS, with over 80 publications. He was head of the receiver department at IFEN for 17 years, but joined the Vehicle Motion and Positioning Sensor team at Bosch GmbH in 2018. In 2019 he joined the G2G payload team at Airbus. Since 2022 he is an advisor to the EC on GNSS, focusing on authentication services.

Cillian O'Driscoll holds a Ph.D. in Electrical and Electronic Engineering from University College Cork (UCC). He has been active in satellite navigation research and development since 2001, with more than 70 publications in international conferences and journals. He previously worked as a research engineer in the University of Calgary, Canada, and with the European Commission contributing to the EU satellite navigation system Galileo. He is currently an independent consultant specializing in signals, receivers and algorithms for PNT.

Gianluca Caparra is a Radionavigation System Engineer at the European Space Agency at ESTEC. Previously he was PostDoc researcher at the University of Padova, where he received a Ph.D. on Information Engineering in 2017 and a master degree in in Telecommunication Engineering in 2013. His research interests focus on PNT assurance, signal processing and machine learning, mainly in the context of GNSS and SBAS.

Rafael Terris-Gallego received his M.Sc. degrees in Telecommunication Engineering from the Universitat Politècnica de Catalunya (Spain) and in Digital Communications Systems from Telecom Bretagne (France), both in 2001. He has worked as an engineer in mobile and satellite communications for more than 13 years. Currently, he is a researcher in GNSS at the Institute of Space Studies of Catalonia (IEEC) and teaches at both the Universitat Autònoma de Barcelona (UAB) and the Universitat Oberta de Catalunya (UOC).

José A. López-Salcedo received the Ph.D. degree in telecommunication engineering from Universitat Politècnica de Catalunya (UPC) in 2007. In 2006 he joined the Department of Telecommunication and Systems Engineering, Universitat Autònoma de Barcelona (UAB), where he is a professor and served as coordinator of the Telecommunications Engineering. His research interests lie in the field of signal processing for communications and navigation, with special emphasis on GNSS signal processing techniques.

Gonzalo Seco-Granados received the Ph.D. degree in telecommunications engineering from the UPC in 2000, and the MBA degree from the IESE Business School, Spain, in 2002. From 2002 to 2005, he was member of the European Space Agency. He is currently Professor at Signal Processing for Communications and Navigation (SPCOMNAV) research group of the Universitat Autonoma de Barcelona. He is also affiliated with the Institute of Space Studies of Catalonia.

Tom Willems obtained a PhD degree from Ghent University in 2006. From 2006 to 2021, he worked as embedded software engineer in GNSS signal processing and as system engineer at Septentrio and Antwerp Space. At both companies, he was deeply involved in the Galileo Test User Receiver projects for ESA. Next, he started working as an independent consultant. Since 2024, he is employed as Senior Consultant at CGI where his current assignment is to provide advisory services to the EC.

Beatrice Motella is a Project Officer at the Joint Research Centre of the European Commission, within the Space, Connectivity and Economic Security Unit. She holds a Ph. D. degree in Electronics and Communications Engineering obtained at Politecnico

https://doi.org/10.33012/2024.19707

di Torino. Her activities cover different aspects of radio navigation, with a major focus on Galileo signal authentication and signal processing.

F. Javier de Blas holds an MSc in Aeronautical Engineering and in Airport Systems from the Technical University of Madrid (UPM). He has more than 18 years of experience in the aerospace industry. He currently works at EUSPA as Galileo Commercial and High Accuracy Service Manager.

Daniel Blonski is a Navigation System Performance Engineer at ESTEC, ESA, where he is contributing to the development of the European Navigation Systems as a member of the ESA Directorate of Navigation.

ABSTRACT

This paper presents for the first time the full specification of the Galileo Signal Authentication Service (SAS), formerly known as Assisted Commercial Authentication Service (ACAS). It includes a general description of the concept and a detailed description of the service, with focus on the ground interfaces and cryptographic operations, and with the aim of helping developers to test SAS receiving prototypes in view of the forthcoming SAS early testing phase. The paper also summarizes some of the field and lab testing results obtained so far and outlines some recommendations for future SAS receivers.

INTRODUCTION

GNSS authentication can help detect and mitigate GNSS spoofing (Scott, 2003) (Psiaki & Humphreys, 2016). Galileo is developing a Signal Authentication Service (SAS) based on the encryption of the E6-C signal component and some ancillary information. This authentication service was first conceived as part of the so-called *Commercial Service* (CS). The 2017/224 EU Decision (EC, 2017) defined the CS by three capabilities: a payable high accuracy service , a free data authentication service that would eventually become OSNMA, and a payable "Commercial Authentication Service" or CAS , based on the management of private keys in the receiver and real-time signal decryption (Fernandez-Hernandez, et al., 2014) .

In the following years CAS was subdivided in two different implementations: SCAS and ACAS. SCAS (Standalone Commercial Authentication Service) maintained the real-time signal decryption approach but was kept on hold due to implementation challenges. ACAS (Assisted Commercial Authentication Service) defined a novel concept based on 'semi-assisted' authentication, not requiring receiver private keys and easier to implement. In the meantime, the fee-based high accuracy service became a free service through 2018/321 EU Decision (EC, 2018), eventually becoming the HAS service launched by Galileo in 2023 (EU, 2023) (Fernandez-Hernandez, et al., 2022). EU Decision 2024/1882 last summer (EC, 2024) officialized the provision of signal authentication by Galileo for free, based on the ACAS 'semi-assisted' concept, now renamed simply as Galileo SAS.

After this introduction, the SAS concept is outlined. Next, we provide the detailed SAS specification for both satellite signal-in-space and ground interfaces, as it is being currently prototyped in the system. We later present how to integrate SAS processes into the receiver architecture and summarize some performance results obtained so far. We finalize with some conclusions and next steps.

SAS GENERAL DESCRIPTION

The main objective of SAS is to provide signal authentication (understood in this work as *spreading code* authentication) in a simple way for both receivers and the current Galileo system. For the receiver, this means no private key management or continuous assistance channel. For the system, this means using the existing capabilities of Galileo 1st Generation (G1G), in particular OSNMA (Open Service Navigation Message Authentication) and the E6-C pilot signal, at the moment transmitted openly but to be encrypted soon. The SAS 'semi-assisted' concept is based on the re-encryption of portions of encrypted E6-C with future OSNMA keys, and their publication in a server. A receiver that downloads these *Re-Encrypted Code Sequences* (RECS) can then record a snapshot of E6-C samples and correlate it a posteriori once the related OSNMA key is disclosed.

This concept has some shortcomings. One is the authentication latency, which in the case of SAS can be as low as a few seconds. Another one is the need for the receiver to download and store RECS data (e.g. some Mbytes), depending on the desired

autonomy period. Finally, the encryption of the E6-C pilot signal component is also a drawback for users interested in robust Galileo E6 carrier phase measurements. These shortcomings are mainly because SAS uses an existing infrastructure not designed for this purpose, and they will be recovered in Galileo 2nd Generation (G2G), with ad-hoc new (but backward-compatible) authentication signals. This may include the encryption of the E6-C with OSNMA-dependent keys, so that users do not need to pre-store the RECS (Anderson, Lo, Neish, & Walter, 2023). On the positive side, an advantage of the SAS concept is that it can be implemented virtually over *any* satnav system providing an encrypted signal and an open signal with some regularly transmitted unpredictable bits which can be used as a cryptographic key.

Galileo SAS is based on offline and real-time processes, according to the following steps:

- Offline process:
  - The selection by the ground infrastructure, of E6-C *Encrypted Code Sequences* (ECS), to be transmitted in the future, and their re-encryption with as yet-undisclosed OSNMA-based keys, hereinafter called *Re-Encrypted Code Sequences* (RECS).
  - The periodic upload of the RECS, I/NAV-E6-C BGD (Broadcast Group Delay) and SAS status information to a public secured server.
  - The download and storage of RECS and BGD files by a user receiver for the desired autonomy period. At this step, the receiver must process the SAS Status Log to determine the SAS status. The receiver may also synchronize its reference time with the GNSS Service Centre's (GSC).
- Real-time process:
  - The continuous transmission of the encrypted E6-C by Galileo satellites.
  - The recording in the receiver of a snapshot of E6 digital samples including the ECS.
  - Once the related OSNMA key is disclosed, the generation of the RECS key and the decryption of the RECS to obtain the corresponding ECS.
  - The a-posteriori correlation of the E6 pre-recorded samples with the ECS.
  - The authentication verification based on the code delay measurements from the correlation with the ECS, the I/NAV data authenticated with OSNMA, and the authenticated BGDs.

By using the OSNMA TESLA keychain, the E6-C signal, the RECS and the BGD, the receiver can operate in standalone mode (i.e. without ground assistance) for the duration of the pre-downloaded data, without the need of storing any secret key. FIGURE *1* presents the system and receiver high level functional diagrams, and FIGURE *2* presents the SAS sequence of operations to obtain E6-C measurements from the encrypted stream.

The next sections include the detailed specification of SAS, as it currently stands, and as it is foreseen to be launched as part of the Initial Capability (or "initial signals supply phase" (EC, 2024)) in the following months. First, we briefly describe the existing Signal In Space (SIS) elements. Then, we focus on the ground interface, which is the core of SAS. The ground interface defines the RECS files, the BGD (Broadcast Group Delay) files, and the Status and Log (SLOG) file, as well as the required cryptographic operations.

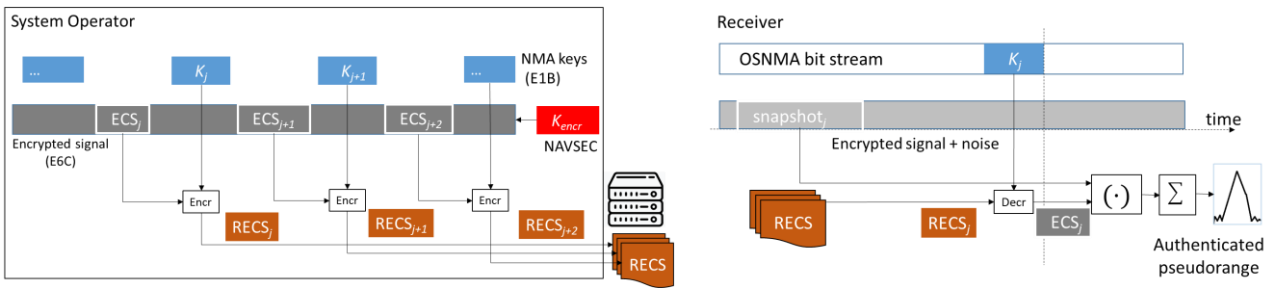


FIGURE 1 – High level diagram of Galileo SAS system operator (left) and receiver blocks (right)

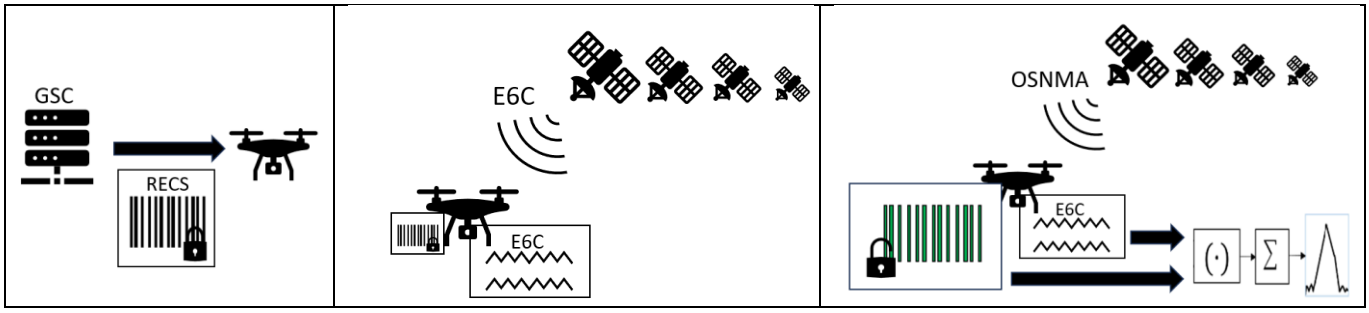


FIGURE 2 – Galileo SAS operation. Left: Download RECS from the server. Center: Capture an E6-C snapshot. Right: Decrypt RECS with OSNMA and perform a-posteriori correlation with E6-C snapshot

SIGNAL-IN-SPACE SPECIFICATION

Galileo SAS is based on the SIS interfaces for the E6-C (EU, 2019) and the E1-B signal components (EU, 2023b). As the E6-C signal component stream is assumed to be encrypted, the receiver receives and can process pseudo-random, non-repeatable sequences (i.e. the ECS), different for each satellite. SAS receivers also receive the Galileo E1-B signal (I/NAV message) OSNMA data (EU, 2023c).

GROUND INTERFACE SPECIFICATION

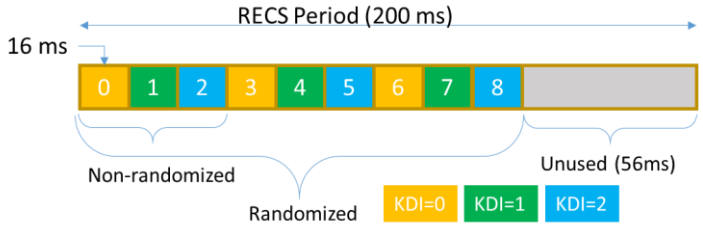
The SAS ground interface is based on SAS RECS, BGD and SLOG files transmitted through a secure interface and a secure time transfer service. All data values here described are encoded using the following bit and byte ordering criteria: the most significant bit/byte is numbered as bit/byte 0, and, for bit/byte ordering, the most significant bit/byte is transmitted first. Signed integers are represented in two's complement, unless otherwise specified. The tables referred to in the following sections can all be found in the Annex. The RECS, BGD files and SLOG files will be available via https from the Galileo GSC server. The Galileo server (or GSC server) is accessible from the SAS root endpoint https://gsc-europa.eu/SAS (to be confirmed at the start of testing).

RECS Data

The SAS server generates and regularly uploads the RECS into a secure server. Various blocks of 16ms RECS are generated every 200 ms, which is the duration of a *RECS period*. The RECS include randomization and different delays with respect to the OSNMA key, according to the following parameters:

- Key Delay Index, KDI: defines the key delay, D_K, or number of I/NAV subframes between the subframe $j + D_k$ containing the OSNMA key K_{j+D_K}, used to generate the RECS key K'_{j+D_K}, and the subframe j containing the RECS that is decrypted with K'_{j+D_K}. KDI ∈ [0,1,2] for a D_K of zero, one and 11 subframes (see TABLE 9).
- Randomization flag, RAND: defines whether the RECS is randomized in time (RAND=1) or not (RAND=0). When RAND=1, $\Delta\tau^k_{j,i}$ defines the random delay added to a RECS for SVID=*k*, RECS 30-second subframe *j* and RECS index within the subframe *i,* in 200-ms time units, where *i* ∈ [1,…,150]. The receiver can determine $\Delta\tau^k_{j,i}$ only after the related OSNMA key K_{j+D_K} is disclosed. When randomization is off (RAND=0), the start of the RECS is located at an offset of $KDI \cdot 16$ ms relative to the start of the RECS period (see *FIGURE 3*).

The structure of a RECS period is depicted in *FIGURE 3* left. Depending on the KDI, RAND and $\Delta\tau^k_{j,i}$, the RECS position will be one of the nine 16-ms slots, as per *FIGURE 3* right.



KDI	0	1	2	0	1	2
RAND	0	0	0	1	1	1
RECS position	0	1	2	0 or 3 or 6	1 or 4 or 7	2 or 5 or 8

FIGURE 3 – Structure of a RECS period (left) and RECS combinations vs. position in RECS period (right)

RECS are expected to be accessible through a query. The preferred option is that a SAS user submits a query via `https` containing the parameters described in TABLE 2. Upon reception of the query, the SAS server assembles all the requested pre-generated RECS files into one binary file per satellite with filename `II_YYDDDHHMMSSS_dddddd_TTTTT_ss_K_R_N_VV.bin`, where the parameters take the value from the query, except '`VV`' which stands for the file version, consistent with the unitary RECS file version described below. For example, the following query is be answered with the following *aggregated RECS files*:

```
https://gsc-europa.eu/sas/recs?intv=01&tstart=241010800008&tdur=007200&rtba=00300
&svids=01+02+05+24&kdi=1&rand=0&nchip=3

 01_241010800008_007200_00300_01_1_0_3_01.bin
 01_241010800008_007200_00300_02_1_0_3_01.bin
 01_241010800008_007200_00300_05_1_0_3_01.bin
 01_241010800008_007200_00300_24_1_0_3_01.bin
```

Each aggregated RECS file is composed by the byte-by-byte concatenation of the *unitary RECS files* stored in the server, as described below. The SAS server may also regularly pre-generate *aggregated RECS files* for certain configurations and make them available for download, as is the case for the BGD or SLOG files.

Each 16ms RECS is stored in the SAS server in a *unitary RECS file*. Unitary RECS filenames are expressed in alphanumeric characters according to the following convention: `XXXXII_YYDDDHHMMSSS_ss_K_R_VV.RCS`, where `XXXX` is an identifier of the file provider, `II` is the interface version, `YYDDDHHMMSSS` is the start of the RECS period in t_{START} format as per TABLE 2, `ss` is the satellite for which RECS are provided, expressed as SVID as per (EU, 2023b), `K` is the KDI, `R` is the RAND flag, and `VV` is the file version. Each of the fields above are right-aligned and padded with zeros where needed. For example, the filename `GSC201_240060800002_09_1_1_01.RCS` means that GSC2 is providing the RECS according to interface version '01', from 6/Jan/2024 08:00:00.2, for SVID=9, a key delay of one subframe ('1'), randomized ('1'), and with file version '01'.

Unitary RECS files are composed of a header and a body section. The header must be consistent with the filename as described in TABLE 3. The unitary RECS file body contains the RECS as specified in the file header and shown in TABLE 4. The unitary RECS files include the maximum RECS length of 16.016 ms, or 81920 chips where the last 80 chips (up to a multiple of 128 bits) are set to zero, so the RECS length does not surpass the 16-ms maximum. When a unitary RECS file content is copied into an aggregated RECS file, and the RECS query defines a number of chips (Nchip) shorter than 81920 chips, only the bytes including the Nchip length are copied. For example, if Nchip=2, only 2560 bytes (20480 bits) will be copied. Note also that, prior to encryption, the ECS +1/-1 signal levels have been converted to binary (0, 1) values according to TABLE 5, and the inverse mapping needs to be applied by the user after decryption.

BGD Data

From the SAS root endpoint, the BGD files are stored from path `./BGD/XXXX/YY/`, where string '`XXXX`' denotes the origin of the file and strings '`YY`' defines the last two digits of the year. BGD files contain the estimations of the BGDs when applying I/NAV clock corrections to E6-C pseudorange measurements obtained from the RECS correlations. BGD filenames are expressed in alphanumeric characters according to the following convention: `XXXXII_YYDDDHHMMSS_LLL_PP_VV.BGD`, where `XXXX` is an identifier of the file provider, II is the interface version, `YYDDDHHMMSS` is the start time of the file, where `YY` are the last two digits of the year, DDD is the day of the year, starting at 001 (1st Jan), `HH` is the hour of the day, `MM` is the minute, and `SS` is the second,

all expressed in GST (Galileo System Time); `LLL` is the duration of the period including BGD estimations, in 10-minute units, `PP` is the period between BGD estimations, in 10-minute units, and `VV` is the file version, as per *TABLE 6*, preceded by zeros if needed. For example, file `GSCH01_23101000000_144_06_01.BGD` means that GSCH (GSC High Accuracy Data Generator) is providing BGD estimations according to interface version '01', from 11/4/2023 (day 101) 00:00:00 for 1440 minutes (24 hours) with a periodicity of 60 minutes. The file version is '01'. The BGD filename shall be consistent with the BGD header parameters. BGD files are composed by a file header and a body section. The BGD file header parameters are defined as per *TABLE 6*. The BGD file body contains the BGDs as specified in the BGD file header and shown in *TABLE 7*. A receiver processing the Galileo I/NAV message can determine the satellite time offset to be applied to the E6-C measurement from satellite *k* at *t*, or $\Delta t^k_{E6C}(t)$, as follows:

$$\Delta t^k_{E6C}(t) = \Delta t^k_{INAV}(t) - BGD^k(t_i) \quad (1)$$

where $\Delta t^k_{INAV}(t)$ is the clock correction from the Galileo I/NAV broadcast message for satellite *k* at *t* (EU, 2023b). $\Delta t^k_{INAV}(t)$ corresponds to $\Delta t_{SV}(X)$ in eqs. (12) and (13) (EU, 2023b), using the I/NAV parameters in *TABLE 64* of same reference. $BGD^k(t_i)$ is the BGD provided from the BGD file for the satellite *k* and time t_i, where t_i is the latest BGD record prior to *t*. BGD is estimated as a constant value and represents the satellite group delay between the E6-C signal component and the I/NAV E1/E5b ionosphere-free combination.

FIGURE 4 – E6-C biases as estimated by the Galileo HAS (currently unauthenticated), 20/9/2024 to 25/9/2024. Source: EUSPA RTM prototype

FIGURE 4 shows the E6-C biases as estimated by the Galileo HAS engine for five days. While they are very stable and only cm-to-dm-level variations are expected, the BGD file includes a BGD estimated accuracy term which, for satellite *k* at *t*, or $\sigma^k_{BGD}(t)$, is calculated as follows:

$$\sigma^k_{BGD}(t) = \sigma^k_{BGD}(t_0) + \delta^k_{1,BGD} \cdot (t - t_0) + \delta^k_{2,BGD} \cdot (t - t_0)^2 \quad (2)$$

where $\sigma^k_{BGD}(t_0)$ is the BGD Accuracy for the same satellite at the beginning of the file t_0, and $\delta^k_{1,BGD}$ and $\delta^k_{2,BGD}$ are the linear and quadratic degradation factors, as per TABLE 13.

Status and Log Data

From the SAS root endpoint, the SLOG files are stored from path `./SLOG/XXXX/YY/`, where the string `'XXXX'` denotes the origin of the file and the string `'YY'` defines the year. Users shall consult the latest SLOG file before the autonomy period. A SLOG file is generated every day. Events affecting SAS Status (SASS) must appear in the latest SLOG file. SLOG filenames are expressed in alphanumeric characters according to the following convention: `XXXXII_YYDDD.LOG`, where `XXXX` is an identifier of the file provider, II is the interface version, `YYDDD` is the day of the file, where `YY` are the last two digits of the year, `DDD` is the day of the year. An SLOG file is an ASCII file composed of entries, in the following format:

```
YYYY_DDD_HH:MM:SS <MID> <…MESSAGE…>
```

where YYYY_DDD_HH:MM:SS defines the time (4-digit year, day of the year, hours, minutes and seconds, referred to GST) of the message entry, <MID> is a four-digit Message ID string, and <…MESSAGE…> contains a variable-length string. SAS predefined messages are defined in TABLE 8. SASS shall be reported using message MID=0001 at the beginning of each SLOG file at HH:MM:SS = 00:00:00. In nominal operation conditions, each SLOG file will only contain a single line with a MID=0001 reporting operational status. Other messages may be defined in future versions of this specification, including any other information of relevance for SAS users. These future versions may include SAS-specific outages or changes on a per-satellite basis that may affect SAS, such as e.g. E6-C encryption status, in addition to other signal-in-space-based measures to report that information.

Secure Time Synchronization

The SAS server will also provide a secure time source, allowing receiver synchronization through NTS (Network Time Security) for NTP (Network Time Protocol) (Franke, Sibold, Teichel, Dansarie, & Sundblad, September 2020).

RECEIVER CRYPTOGRAPHIC OPERATIONS

Once the OSNMA key K_{j+D_K} belonging to subframe $j + D_K$ is received and verified by the OSNMA keychain, the RECS decryption key K'_{j+D_K} is generated as follows, where $\mathrm{SHA256}()$ is the hash function SHA-256 as per (NIST, 2012):

$$K'_{j+D_K} = \mathrm{SHA256}(K_{j+D_K}) \tag{3}$$

Once K'_{j+D_K} is obtained, $\mathrm{RECS}_{j,i}$ is decrypted into $\mathrm{ECS}_{j,i}$ as follows (the satellite superindex is omitted for simplicity but without loss of generality):

$$\mathrm{ECS}_{j,i} = \mathrm{AES256}^{-1}_{CBC}(K'_{j+D_K}, \mathrm{RECS}_{j,i}, \mathrm{IV}) \tag{4}$$

where $\mathrm{AES256}^{-1}_{CBC}(a, b, \mathrm{IV})$ is the inverse cipher AES, configured for 256-bit keys (AES-256 as per (NIST, 2001)) in CBC mode (NIST, 2001b), to decrypt a plaintext *b* with 256-bit key *a*, and with an initialization vector IV. Note that $\mathrm{RECS}_{j,i}$ is a multiple of 128 bits, as required by the AES implementation. The IV is determined as follows:

$$\mathrm{IV} = trunc(128, \mathrm{SHA256}(P_j)) \tag{5}$$

where $trunc(n, p)$ is the truncation function that retains the *n* MSBs of the input *p*, and P_j is a 128-bit plaintext generated as follows:

$$P_j = (GST_{SF,j+D_K} \,||\, \mathrm{RAND} \,||\, p1) \tag{6}$$

where (X || Y) concatenates bitset X to bitset Y, with X at the MSB; $GST_{SF,j+D_K}$ is the 32-bit GST as per (EU, 2023b) of the start of the E1-B I/NAV subframe containing the key, *minus one second* (one second earlier than the start of the subframe containing the key.), i.e. following the definition of GST_{SF} as per (EU, 2023c); RAND is the randomization flag of the RECS, as per TABLE 3 (i.e. 8-bit uint). P_j is padded by $p1$, an array of 88 zeros.

OSNMA TESLA keys are transmitted over the last seconds of the I/NAV subframe. In order to avoid RECS determination before full OSNMA key disclosure, which might be theoretically possible for $D_K = 0$, the I/NAV and RECS subframes are shifted by an *OSNMA key broadcast margin* offset, as depicted in FIGURE 5. This way, the last $\mathrm{RECS}_{j,i}$ of I/NAV subframe *j* are encrypted with K_{j+D_K+1} instead of K_{j+D_K}. RECS within the 200 ms RECS period starting at or after a RECS subframe transition, i.e. at or beyond the red dotted line in FIGURE 5, shall be encrypted with the next OSNMA key. The key broadcast margin offset value is currently TBD and will be fixed by design soon.

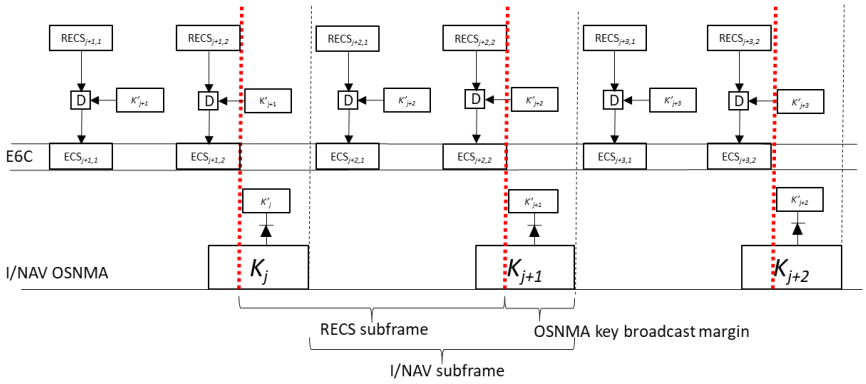


FIGURE 5 – RECS decryption process ('D' in the figure) for $\boldsymbol{D_K = 1}$

When RAND=1, the RECS random delays $\Delta\tau_{j,i}^{k}$ are generated from K'_{j+D_K} for each satellite *k* using AES to generate a sufficiently large ciphertext as follows:

$$\left(C_{j,1}^{k},..,C_{j,10}^{k}\right) = \mathrm{AES256}_{OFB[1:10]}(K'_{j+D_K}, P_j, trunc(128, \mathrm{SHA256}(k)) \quad (7)$$

$$C_{j,1}^{k} = \left[B_{j,1}^{k},\ldots,B_{j,16}^{k}\right]; C_{j,2} = \left[B_{j,17}^{k},\ldots,B_{j,32}^{k}\right]; C_{j,3} = \left[B_{j,33}^{k},\ldots,B_{j,48}^{k}\right];\ldots C_{j,4} = \left[B_{j,144}^{k},\ldots,B_{j,160}^{k}\right]; \quad (8)$$

where $\left(C_{j,1}^{k},..,C_{j,10}^{k}\right)$ consists of 10 128-bit (16-byte) blocks necessary to define the random delays for the 150 RECS in a subframe; $\mathrm{AES256}_{OFB[1:10]}$ is the AES cipher in OFB mode as per (NIST, 2001b) (s6.4), for 10 iterations, P_j is defined as per eq. (5); and the last term of (7) serves as AES Initialisation Vector. Then, $\Delta\tau_{j,i}^{k}$ are calculated from each of the first 150 bytes $B_{j,1..150}^{k}$ as follows:

$$\Delta\tau_{j,i}^{k} = \mathrm{KDI} + 3 \cdot \left(B_{j,i}^{k} \bmod (3)\right); i \in [1,..,150] \quad (9)$$

where mod is the *modulo* operator. $\Delta\tau_{j,i}^{k}$ provides a randomized slot between 0 and 8, as per FIGURE 3. In order to convert to milliseconds with respect to the RECS period start, or $\Delta\tau_{j,i}^{k}[ms] = 16 \cdot \Delta\tau_{j,i}^{k}$.

SAS RECEIVER ARCHITECTURE AND TEST RESULTS

The previous sections have described the SAS up to ECS decryption and synchronization. The E6-C measurement generation and PVT authentication depend on the receiver architecture. This process will be described in the future Galileo SAS SDD (Service Definition Document) or other Galileo documentation, when the service is officially launched. For the moment, we refer to mechanisms that have been developed and tested in previous work and highlight the main challenges.

E6-C Signal Detection and Measurement Generation

The SAS receiver needs to generate the correct E6-C ECS replica and synchronize it with the snapshot for each satellite. This requires estimating the E6-C code phase (or *sequence* phase) in time and Doppler frequency. If the receiver needs to perform a full search, the process can be very time consuming and degrade signal detection (Winkel, Fernandez-Hernandez, & O'Driscoll, 2024) (Terris-Gallego R. , Fernandez-Hernandez, López-Salcedo, & Seco-Granados, 2022). Therefore, a SAS receiver should use the existing time/frequency estimations from other signals to reduce computational complexity, still pending confirmation of their authenticity with the E6-C correlation. The most obvious candidates are the E1-B/E1-C and the E6-B signal components. E1-B is required for OSNMA in nominal operation unless the receiver obtains the TESLA keys through another channel. In this process, the receiver needs to apply pre-calibrated receiver HW biases and adjust the ionosphere estimated delay, which is higher for E6 than for E1, and deal with a few-meter uncertainty window from the different measurement errors (Fernandez-Hernandez, et al., 2023). HAS's E6-B is also a good candidate, given that time, frequency and phase offsets are expected to be practically zero in this case (Terris-Gallego R. , Fernandez-Hernandez, López-Salcedo, & Seco-Granados, 2024).

The E6-C measurement accuracy depends on the RECS length and the measurement generation process. While just-acquired signals are usually less accurate than tracked ones, snapshot architectures allow to achieve similar accuracies as those of tracking through several techniques starting from very simple ones, such as interpolation (Borre, Fernandez-Hernandez, López-Salcedo, & Bhuiyan, 2022).

We have analyzed the accuracy of E6-C measurements in prototype SAS receivers, as shown in FIGURE 6. The top left shows the frequency distribution of E6-C signal detection from the snapshot under the following conditions: C/N_0 = 35 dBHz, moderate multipath, high ionosphere activity, high dynamics (~9g), E1-E6 receiver-satellite biases known and 16-ms RECS correlated (Winkel, Fernandez-Hernandez, & O'Driscoll, 2023). False alert probability (P_{fa}) was fixed at 10^{-7} leading to a theoretical detection probability P_d = 99.9995% without errors other than signal noise, and an obtained probability of 99.95% including multipath and ionosphere. The figure also shows results of signal present (H1) and absent (H0) hypotheses. The top right plot shows an example of the correlation results of multiple correlations, at different sample bins. The bottom left plot shows the SAS prototype developed by UAB in the PAULA project, and composed of two OCXO-synchronized bladeRF SDRs for E1 and E6 respectively, and the bottom right plot shows the range errors (including an uncalibrated bias of about 2m) between both E1-B and E6-C measurements, for 4-ms RECS in open sky, static conditions (Terris-Gallego, López-Salcedo, Seco-Granados, & Fernandez-Hernandez, Sept. 2023), showing an E6-C measurement accuracy comparable to that with continuous tracking. Other receiver architectures have been explored in the NACSET and PAULA research projects (Cancela, et al., 2019), and the ESA-funded Nautilus platform (O'Driscoll & Caparra, 2021).

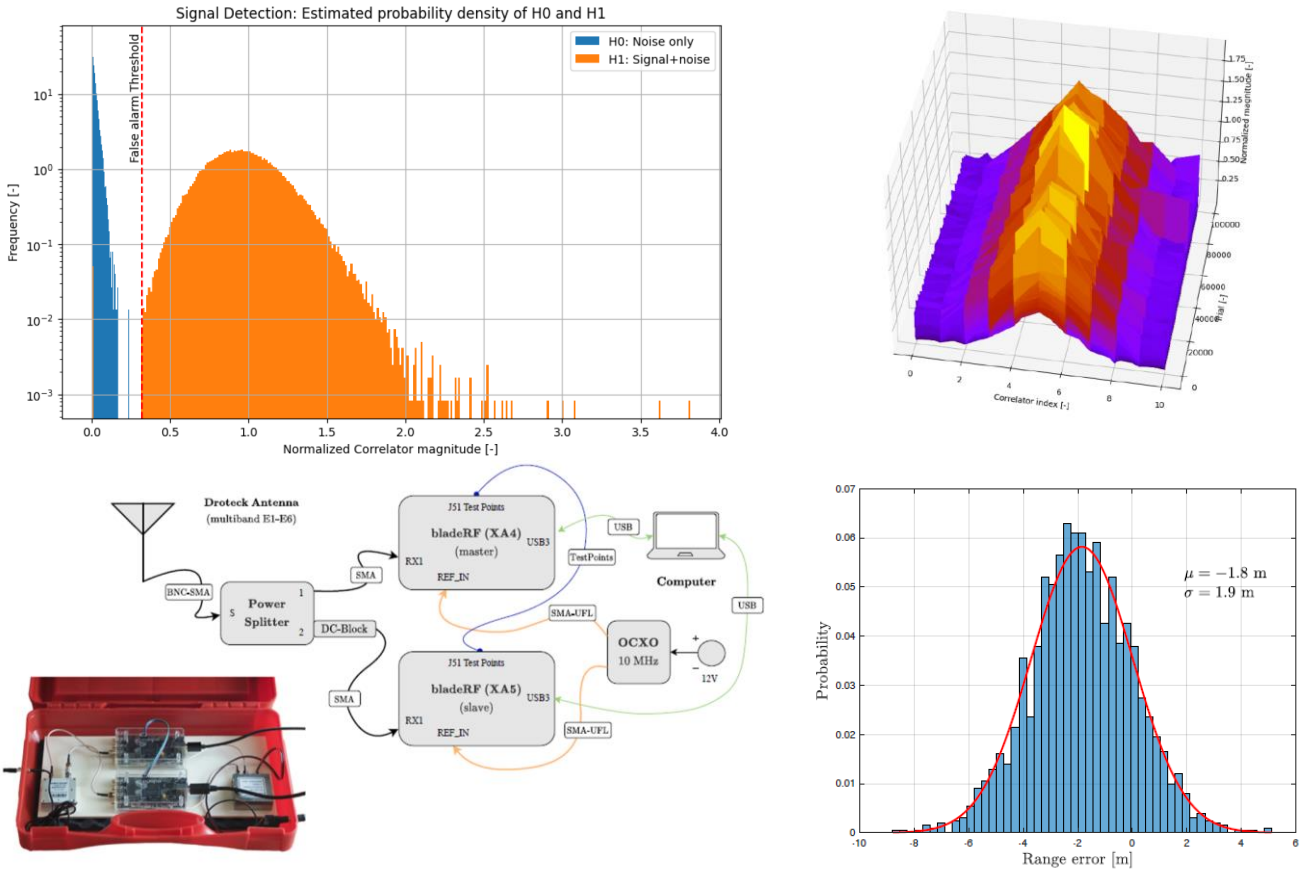


FIGURE 6 – SAS simulation and field testing. Top left: signal detection histogram. Top right: E6-C correlations over time (Winkel, Fernandez-Hernandez, & O'Driscoll, 2023). Bottom left: UAB SAS first prototype architecture. Bottom right: E1-E6 measurement comparison

Authenticated PVT Calculation

Once the SAS receiver has E6-C measurements and authentic data, the problem of authenticating the resulting position (or PVT, Position, Velocity and Time), and to which degree, remains open. It is known that spreading code encrypted/authenticated signals can be *meaconed*, leading to false positions. However, as meaconed signals arrive in delay, a SAS receiver can perform

Vestigial Signal Search (VSS) to detect, and even mitigate, selective meaconing (Winkel, Fernandez-Hernandez, & O'Driscoll, 2024). A spoofer/meaconer raising the signal power to conceal the authentic signals may be detected by the AGC, if checked. Another limitation of SAS (and asymmetric signal authentication in general) is the authentication delay, and the fact that the receiver may need to *coast* between authentications, extrapolating the level of confidence from the last E6-C-based position. FIGURE 7 presents an overview of a possible SAS receiver logic, where the E6-C SAS navigation can be combined with AGC/CN0/VSS and other checks (Akos, 2012) (Wesson, Shepard, Bhatti, & Humphreys, 2011) in an authentication module to achieve a good level of trust.

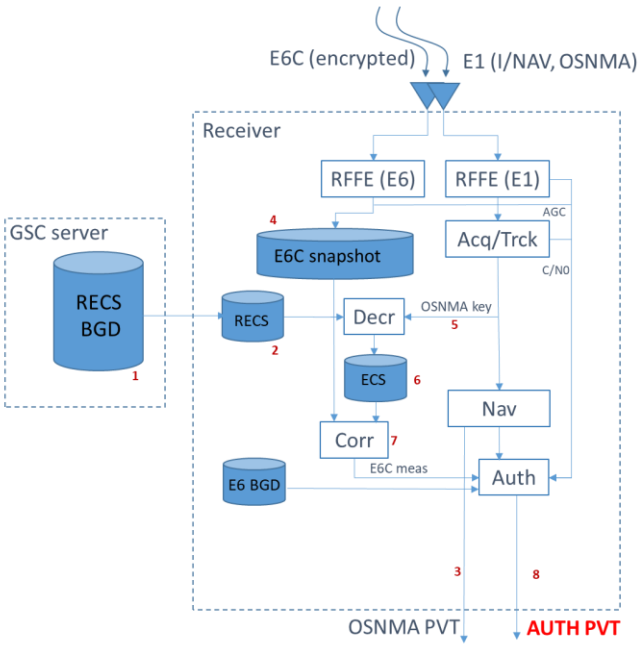


FIGURE 7 – High level receiver architecture for authenticated PVT, including RECS/BGD upload to (1) and download from (2) the server, OSNMA E1 navigation (3), E6-C snapshot recording (4), ECS decryption with OSNMA (5,6), E6-C correlation (7), and use in authenticating engine in combination with AGC, CN0 and potentially other checks (consistency, VSS) for an authenticated PVT (8)

RECS Storage Requirements

Another additional capability required for a SAS receiver is the RECS storage. Also, the size of the snapshots can be important (specially in case of using VSS) (Terris-Gallego R. , Fernandez-Hernandez, López-Salcedo, & Seco-Granados, 2024). The RECS storage requirement depends on the receiver autonomy period, the RECS desired time between authentications (RTBA), the number of satellites stored, and the RECS length (Nchip). For example, a receiver requiring a 2-hour autonomy with a 30-second RTBA, RECS for 4 satellites, and 8-ms correlations would require 4.91 Mbytes of space, which seems affordable. Some other examples are shown in TABLE 1. At the moment, the maximum autonomy foreseen for SAS receivers, i.e., the period for which RECS will be published in advance, is one week.

TABLE 1: RECS storage required for different use cases

Nsats	RTBA [s]	Nchip	Corr. time [ms]	Autonomy [h]	Total storage [Mbytes]
4	30	41690	8.008	2	**4.91**
24	30	5120	1.001	12	**22.6**
10	30	5120	1.001	1	**0.78**
24	120	41690	8.008	120	**451.35**

CONCLUSION AND NEXT STEPS

Galileo SAS is a global and free signal authentication service provided by Galileo 1st Generation. The *Galileo SAS* term intends to replace previous signal authentication concepts and analyses, referred to as CS Authentication, CAS, SCAS or, more recently, ACAS.

Galileo SAS is built on existing infrastructure and signals (E1-B and E6-C). The concept is based on re-encrypting portions of the to-be-encrypted E6-C signal component with future OSNMA keys and publishing them in a Galileo server. SAS receivers do not require private keys or a continuous assistance channel. Instead, they only need to connect at the beginning of the operation period. In exchange, SAS adds a latency of some seconds and requires some Mbytes of storage in the receiver.

This paper has described extensively the SAS ground interface, including the RECS download process and format, at the core of the concept, the BGD and SLOG files, as well as all cryptographic operations. The paper also presents a summary of results obtained so far, including simulated and real and conditions. As other spreading code authentication schemes, SAS can be combined with other anti-spoofing measures, at RF (AGC), signal processing (C/N_0, VSS) or measurement level, to improve spoofing detection and mitigation techniques.

As already announced in EU Decision 2024/1882 this summer, Galileo intends to provide a SAS early capability by the end of 2024. This early capability will be based on a prototype RECS/BGD server and the E6-C encryption of the L3 satellites in eccentric orbits and will be followed by the E6-C encryption of the rest of the constellation in the first quarter of 2025. An operational service launch is expected by 2026. SAS will evolve in Galileo 2nd Generation with ad-hoc designed signals for better performance and autonomy.

DISCLAIMER

The content of this article does not necessarily reflect the official position the authors' organizations. Responsibility for the information and views set out in this article lies entirely with the authors. Readers may use the content of this paper at their own risk, and future SAS technical publications, recommendations or specifications may differ from the content presented in this work.

ACKNOWLEDGMENTS

The authors would like to thank the S. Cancela and the NACSET, PAULA and MMARIO teams for their work in the development and testing of SAS, and C. Hernandez for the EUSPA RTM tool.

REFERENCES

Akos, D. M. (2012). Who's Afraid of the Spoofer? GPS/GNSS Spoofing Detection via Automatic Gain Control (AGC). *Navigation, 59(4), 59(4)*(https://doi.org/10.1002/navi.19), 281–290.

Anderson, J., Lo, S., Neish, A., & Walter, T. (2023). Authentication of Satellite-Based Augmentation Systems with Over-the-Air Rekeying Schemes. *NAVIGATION: Journal of the Institute of Navigation, 70*(3).

Borre, K., Fernandez-Hernandez, I., López-Salcedo, J. A., & Bhuiyan, M. Z. (2022). *GNSS Software Receivers.* Cambridge University Press.

Cancela, S., Navarro, J., Calle, D., Reithmaier, T., Chiara, A. D., Broi, G. D., . . . Simón, J. (2019). Field Testing of GNSS User Protection Techniques. *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019).* Miami, FL, pp. 1824-1840. https://doi.org/10.33012/2019.17087.

EC. (2017). Commission Implementing Decision (EU) 2017/224 of 8 February 2017 (CS Implemeting Act).

EC. (2018). Commission Implementing Decision (EU) 2018/321 of 2 March 2018 (amending Implementing Decision (EU) 2017/224).

EC. (2024). Commission Implementing Decision (EU) 2024/1882 amending Commission Implementing Decision (EU) 2017/224 [...] as regards the free provision of a signal authentication service.

EU. (2017). *Patent No. EP3349044A1.*

EU. (2019). *E6-B/C Signal-In-Space Technical Note.*

EU. (2022). *Galileo Open Service Navigation Message Authentication (OSNMA) Signal-In-Space Interface Control Document (SIS ICD), Issue 1.0.* EUSPA.

EU. (2023). *Galileo High Accuracy Service - Service Definition Document (HAS SDD) v1.0.* EUSPA.

EU. (2023b, September). *Galileo Open Service Signal-In-Space Interface Control Document (OS SIS ICD), Issue 2.1.*

EU. (2023c). *Galileo Open Service Navigation Message Authentication (OSNMA) SIS ICD Issue 1.1.*

Fernandez-Hernandez, I., Chamorro-Moreno, A., Cancela-Diaz, S., Calle-Calle, J. D., Zoccorato, P., Blonski, D., . . . Mozo, A. (2022). Galileo High Accuracy Service: Initial Definition and Performance. *GPS Solutions, 65*(26).

Fernandez-Hernandez, I., Simón, J., Blasi, R., Payne, C., Miquel, T., & Boyero, J. P. (2014). The Galileo commercial service: current status and prospects. *European navigation conference ENC2014.* Rotterdam.

Fernandez-Hernandez, I., Winkel, J., O'Driscoll, C., Cancela, S., Terris-Gallego, R., & López-Salcedo, J. A. (2023). Semi-Assisted Signal Authentication for Galileo: Proof of Concept and Results. *IEEE Transactions on Aerospace and Electronic Systems*(DOI. No. 10.1109/TAES.2023.3243587).

Franke, D., Sibold, D., Teichel, K., Dansarie, M., & Sundblad, R. (September 2020). *RFC 8915 - Network Time Security for the Network Time Protocol.* Internet Engineering Task Force (IETF) - ISSN: 2070-1721 DOI 10.17487/RFC8915, <https://www.rfc-editor.org/info/rfc8915>.

NIST. (2001). *Federal Information Processing Standards Publication 197 (FIPS-197) Specification for the ADVANCED ENCRYPTION STANDARD (AES).* NIST.

NIST. (2001b). *NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation.* NIST.

NIST. (2012). *FIPS PUB 180-4: Secure Hash Standard (SHS).*

O'Driscoll, C., & Caparra, G. (2021). Nautilus: An Embedded Navigation Authentication Testbed. *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021).* St. Louis, Missouri, September 2021, pp. 3698-3710. https://doi.org/10.33012/2021.17976.

Psiaki, M. L., & Humphreys, T. E. (2016). GNSS spoofing and detection. *Proceedings of the IEEE, 104(6)*, 1258-1270.

Scott, L. (2003). Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems. *ION GPS*.

Terris-Gallego, R., Fernandez-Hernandez, I., López-Salcedo, J. A., & Seco-Granados, G. (2022). Guidelines for Galileo assisted commercial authentication service implementation. *International Conference on Localization and GNSS (ICL-GNSS) (pp. 01-07). IEEE.*

Terris-Gallego, R., Fernandez-Hernandez, I., López-Salcedo, J. A., & Seco-Granados, G. (2024). Efficient Detection of Galileo ACAS Sequences using E6-B. *European Navigation Conference ENC2024; MDPI Engineering Proceedings.* Noordwijk, NL.

Terris-Gallego, R., López-Salcedo, J. A., Seco-Granados, G., & Fernandez-Hernandez, I. (Sept. 2023). Preliminary Evaluation of Galileo ACAS Using Existing E1-E6 Open Signals and a Low-Cost SDR Platform. *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023).* Denver, Colorado, pp. 1388-1402. https://doi.org/10.33012/2023.19319.

Wesson, K. D., Shepard, D. P., Bhatti, J. A., & Humphreys, T. E. (2011). An Evaluation of the Vestigial Signal Defense for Civil GPS Anti-Spoofing. . *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011).* 177-193.

Winkel, J., Fernandez-Hernandez, I., & O'Driscoll, C. (2024). Combining Galileo's Assisted Commercial Authentication Service (ACAS) with Vestigial Signal Search for Good Protection. *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation (pp. 1225-1234).*

Winkel, J., Fernandez-Hernandez, I., & O'Driscoll, C. (2023). Implementation Considerations for ACAS and Simulation Results. *IEEE Transactions on Aerospace and Electronic Systems*(DOI 10.1109/TAES.2024.3402196).

ANNEX

This Annex provides the tables detailing the RECS, BGD and SLOG file specification, including the relevant parameters referred in the main text.

TABLE 2: RECS query parameters

Parameter	Definition	Type[1]	Format
IntV	RECS interface version. It indicates the interface version to interpret the file content and format. Future versions of the specification will increment this value. Current value: '01'. Values '00', '02'-'99': reserved.	ASCII	II
t_{START}	Start time of the autonomy period, expressed in GST, where YY are the last two digits of the year, DDD is the day of the year, starting at 001 (1st Jan), HH is the hour of the day, MM is the minute, and SSS is the *tenths of second*, and a multiple of 0.2s.	ASCII	YYDDDHHMMSSS
TDur	Duration of the autonomy period, expressed in seconds.	ASCII	dddddd
RTBA	RECS Time Between Authentications (TBA), expressed in *tenths of second* and a multiple of 0.2s.	ASCII	TTTTT
SVIDs	List with one or more entries of requested satellites, identified as SVID as per (EU, 2022), where ss ∈ ['01',..,'36'].	ASCII	ss,ss,…,ss
KDI	Key Delay Indicator (KDI) as per TABLE 9.	ASCII	K
RAND	Randomization flag (applied: R='1'; not applied: R='0').	ASCII	R
NChip	ECS Sequence length in chips as per TABLE 10	ASCII	N

TABLE 3: Unitary RECS File Header Parameters and Correspondence with the Filename

Parameter	Definition	Type	Bits	Scale	Unit
Provider	4-byte string to be filled in by the file provider, corresponding to 'XXXX' in the filename.	ASCII	32	-	-
IntV	Version of the interface corresponding to 'II' in the filename. Values 1:version ; 0, 2-99: Reserved; 100-255: Unused.	uint	8	-	-
Reserved	Reserved bits.	-	4	-	-
t_{START},WN	WN of the GST second in which the RECS file starts as per (EU, 2023b) and linked to t_{START},ATOW. It shall be consistent with 'YYDDDHHMMSSS' in the filename.	uint	12	1	week
t_{START}, ATOW	SAS TOW of the RECS file in GST in tenths of a second, based on (EU, 2023b) but with 10x higher resolution. It shall be consistent with 'YYDDDHHMMSSS' in the filename. It expresses and shall be consistent with the start time of the 200ms period where the RECS is found.	uint	24	0.1	s
SVID	Satellite ID as per (EU, 2023b), corresponding to 'ss'. Values 1-36: SVID; 0,37-99: Reserved. 100-255: Unused.	uint	8	-	-
KDI	KDI, as per TABLE 9, corresponding to 'K' in the filename.	uint	8	-	-
RAND	Randomization, corresponding to 'R' in the filename. Values 0: no randomization; 1: randomization; 2-9: reserved; 10-255: unused.	uint	8	-	-

[1] The parameter type convention for all tables is 'ASCII': text char/string; 'uint': unsigned integer (of the length specified in the 'Bits' field); 'int': signed integer using two's complement, and where the sign bit (+ or -) occupies the MSB; '-': not applicable.

File Version	File Version, corresponding to 'VV' in the filename. It starts at 1 and increases if a new version of a RECS file for the same period and with the same SVID-KDI-RAND configuration is created. Values: 1-20: File Version; 21-99: Reserved; 100-255: Unused.	uint	8	-	-
Header length	Length of the header, in bytes[2].	uint	16	1	bytes

TABLE 4: Unitary RECS File Body

Parameter	Definition	Type	Bits	Scale	Unit
$\mathrm{RECS}^{\mathrm{SVID}}$	For SVID, RECS of the file.	-	81920	-	-

TABLE 5: Logic to Signal Level Assignment (EU, 2022)

Logic Level	Signal Level
0	+1
1	-1

TABLE 6: BGD File Header

Parameter	Definition	Type	Bits	Scale	Unit
Provider	4-byte string to be filled in by the file provider, corresponding to 'XXXX' in the filename.	ASCII	32	-	-
IntV	Version of the interface corresponding to 'II' in the filename. Values 1:version ; 0, 2-99: Reserved; 100-255: Unused.	uint	8	-	-
t_{START},TOW	TOW (EU, 2023b) of the GST second in which the BGD file starts. It shall be consistent with YYDDDHHMMSS in the filename, and coincide with an I/NAV subframe start.	uint	20	1	s
t_{START},WN	WN (EU, 2023b) of the GST second in which the BGD file starts. It shall be consistent with YYDDDHHMMSS in the filename, and coincide with an I/NAV subframe start.	uint	12	1	week
L	File length, in 10-minute units, for which BGD are provided in the file, corresponding to 'LLL' in the filename. Note that *L* defines the time for the *last* BGD provided, whose validity period can be extended beyond the file duration. Values 1-999: *L* in 10-minute units; 0,1000-65535: Reserved.	uint	16	10	min
τ_{BGD}	BGD Period, in 10-minute units, corresponding to 'PP' in the filename. Values 1-99: Period in 10-minute units; 0,100-255: Reserved.	uint	8	10	min
Reserved	Reserved bits	-	8	-	-
BVI	BGD Validity Interval. Every BGDs in the file is valid from the start of its BGD period and during BVI.	uint	4	-	-
SVID mask	Mask with 36 bits defining for which SVID BGDs are provided, starting with SVID1 in the first bit. '1' means SVID BGD provided; '0' means SVID BGD not provided.	-	36	-	-

[2] If the spec evolves in the future and the header is extended, a file reader following the initial spec can skip the new bytes and start reading the body. The file evolutions shall be defined so that new body content is placed after the content as per the current spec.

$\boldsymbol{\sigma_{BGD}}$	Array of 36 4-bit values with BGD Accuracies $\sigma_{BGD}^{k=1..36}$ of the *first* BGD estimation in the file, for SVID1 to SVID36[3], and before applying any degradation factor. Every σ_{BGD}^{k} expresses the standard deviation of the over-bounding Gaussian distribution of the BGD error.	-	144	-	-
$\boldsymbol{\delta_{BGD}}$	Array of 36 8-bit values of BGD degradation factors, for every SVID (SVID1 to SVID36). For a given satellite k = 1,…,36, the first 4 bits provide the linear degradation factor $\delta_{1,BGD}^{k}$, and the second 4 bits provide the quadratic degradation factor $\delta_{2,BGD}^{k}$.	-	288	-	-
File Version	File Version, corresponding to 'VV' in the filename. It starts at 1 and increases if a new version of a BGD file for the same period and periodicity is created. Values 1-20: File Version; 21-99: Reserved; 100-255: Unused.	uint	8	-	-
Header length	Length of the header, in bytes.	uint	16	1	bytes

TABLE 7: BGD File Body

Parameter	Definition	Type	Bits	Scale	Unit	Range
$BGD^{1}(t_0)$	BGD of the 1st satellite in SVID mask, 1st batch (t_0 = *Start Time*).	int	16	0.02	m	-655.36, 655.34
…	…	…	…	…	…	…
$BGD^{K}(t_0)$	BGD of the last satellite in the SVID mask (*K*), 1st batch (t_0).	int	16	0.02	m	-655.36, 655.34
$BGD^{1}(t_0 + \tau_{BGD})$	BGD of the first satellite, 2nd batch ($t_1 = t_0 + \tau_{BGD}$)	int	16	0.02	m	-655.36, 655.34
…	…	…	…	…	…	…
$BGD^{1}(t_0 + L)$	BGD of the first satellite, last batch (t_0+ *L*).	int	16	0.02	m	-655.36, 655.34
…	…	…	…	…	…	…
	BGD of the last satellite, last batch (t_0+ *L*).	int	16	0.02	m	-655.36, 655.34

TABLE 8: SLOG Messages to handle SASS

MID	Message	Description
0001	Current SASS=[OPER /TEST/DONT_USE]	SAS Status (SASS) information message.

[3] BGD accuracy and degradation are defined only once per file.

TABLE 9: Key Delay (D_K) Indicator (KDI)

Value	I/NAV Subframe Offset D_K (seconds)
0	0 (0s)
1	1 (30s)
2	11 (330s)

TABLE 10: NChip

Value	Chip Length	Duration
0	5120	1.001 ms
1	10240	2.002 ms
2	20480	4.004 ms
3	40960	8.008 ms
4	81920[4]	16.016 ms
5-9	Reserved	-

TABLE 11: Broadcast Group Delay Validity Interval

Value	BGD Validity
0	30 min
1	1 h
2	2 h
3	4 h
4	1 day
5-14	Reserved
15	Validity not provided

TABLE 12: Broadcast Group Delay Accuracy[5]

Value	BGD Accuracy
0..49	0 cm to 49 cm with 1 cm resolution
50..74	50 cm to 98 cm with 2 cm resolution
75..99	1 m to 1.96 m with 4cm resolution
100..254	Reserved

TABLE 13: BGD Degradation Factor

Bits 0-3	BGD Linear Degradation Factor δ_1 [m/s]	Bits 4-7	BGD Quadratic Degradation Factor δ_2 [m/s²]
0	0	0	0
1	10^{-5}	1	10^{-5}
2	10^{-4}	2	10^{-4}
3	10^{-3}	3	10^{-3}

TABLE 14: SAS Status

Value	Definition	Semantic
TEST	Test Mode	SAS service testing activities ongoing. Nominal performance may not be met.
OPER	Operational Mode	SAS is expected to provide nominal performance.
DONT_USE	Don't Use	Users shall stop using SAS and discard previously received SAS information.

[4] The last 80 chips are set to zero.

[5] This table uses the same values of Galileo SISA field (EU, 2023b). By providing an estimation of the accuracy the SAS user can use this estimation for its measurement or position authentication test hypothesis.