SNAP: an Authentication Concept for the Galileo Open Service

Beatrice Motella and Davide Margaria
Navigation Technologies Research Area
Istituto Superiore Mario Boella
Torino, Italy
surname@ismb.it

Matteo Paonni
Directorate for Space, Security, and Migration
Joint Research Centre of the European Commission
Ispra, Italy
matteo.paonni@ec.europa.eu

***Abstract*— The design of a solution for the authentication of both navigation data bits and spreading code chips, referred to as *SNAP* and suitable for the evolution of the Galileo E1 OS signal, is presented in the paper. Though the technique is innovative and able to achieve predefined authentication performance, it exploits the structure of the legacy Galileo signal and the characteristics of the OS NMA.**

A detailed overview of the open choices for the design of signal components dedicated to authentication is provided, together with an analysis of signal parameters definition. A possible implementation option of the SNAP solution is also presented.

Keywords—Spreading code and navigation data based authentication proposal; Galileo; open service; navigation message authentication; spreading code authentication.

I. INTRODUCTION

Though the vulnerability of Global Navigation Satellite Systems (GNSSs) to Radio Frequency Interference (RFI) is a reason of concern since many years [1], the awareness of this kind of threats has been recently further strengthen, also thanks to the increased number of reported events of intentional interference [2].

The large availability of personal jammer devices, that can be easily purchased over the web and (illegally) used, makes them the primary source of intentional interference, both of malicious and uninformed nature. According to the classification given in [3], jamming broadcasted on or near GNSS frequencies is intended to damage GNSS users (malicious interference), but in the majority of the cases, there is no the intent to cause harm to third parties (uninformed interference) [4]. In addition, in the last years the attention went more to structured forms of interference, known as spoofing attacks, which have the goal of producing false information within the victim receiver. Not only the feasibility of such attacks have been widely demonstrated trough experiments and trials [5][6], but, more recently, their real implementation in the civil domain has been also documented [2].

Several anti-spoofing techniques have been proposed [7][8], based on a wide variety of different approaches. A first macro classification of such methods separates among those implemented at the receiver level and those directly applied at the signal level. The formers work on specific observables available along the receiver chain (e.g., antenna-aided techniques, methods based on the signal power monitoring or consistency check with other navigation sensors); while a second group consists of techniques that implement civil-signal authentication and cryptographic defense algorithms [9]. In this sense, a proper partition between system and receiver contribution to the robustness against spoofing represents a key aspect [10], in order to take advantage from both cryptographically secure features in the Signal In Space (SIS), and the implementation of non-cryptographic countermeasures.

In the design of the authentication solutions for a new generation of civil GNSS signals one can act on different signal's components. As described in [10], the Navigation Message Authentication (NMA) denotes the protection of the navigation message bits (i.e., the full data frame or a portion of it) and can be implemented by digitally signing the navigation data, thus keeping the navigation message unencrypted. Spreading Code Authentication (SCA) inserts, within the nominal (unencrypted) spreading code, unpredictable portions of chips, which are later verified through cryptographic functions.

Within this context, the work has been encouraged by the need of increasing the level of the SIS robustness, required in many applications. The paper proposes an authentication concept able to exploit some of the characteristics of the current Galileo Open Service (OS) signal [11] and those of the OS NMA [12], that will be transmitted starting from 2018 [13].

The paper firstly provides a detailed analysis on the open choices for the design of signal components dedicated to authentication, describing tangible benefits and considering possible limitations of each solutions. In addition, on the basis of the presented analysis, the paper presents the *Spreading code and Navigation data based Authentication Proposal* (SNAP), a solution to provide authentication both at the navigation data and spreading code levels and tailored to the evolution of the Galileo E1 OS signal. In detail, the proposed approach builds upon the structure and the characteristics of the OS NMA, thus being capable of increasing its spoofing robustness performance.

The remaining of the paper is organized as follows: section II provides an overview of the signal design aspects relevant for authentication at the spreading code level, while section III is dedicated to the description of the SNAP concept. After investigating the performance of the solution under different families of spoofing attacks (section IV), a trade-off analysis, addressed to the definition of the solution parameters, is presented in section V. Section VI drafts a possible implementation of the SNAP concept, referred to as *working point*. The conclusions of the work are summarized in section VII, that also highlights the open points and sketches some ideas for future activities.

II. Signal Design: a General Overview

In the design of the signal authentication solutions one can act on different signal's components. NMA and SCA techniques work at the navigation message and spreading code levels respectively, but in principle signal components dedicated to authentication can be also inserted at other levels, e.g., unpredictable features at subcarrier or carrier level, as variable pulse shaping, frequency hopping, etc.

The options for the design of an authentication solution based on spreading code are listed in Fig. 1. It presents a tree for each parameter that is considered relevant for the SCA authentication technique. The main possible parameters' alternatives, as identified by the authors, are listed as tree's leaves. In light orange are highlighted those ideas still draft, not consolidated in the literature and/or not demonstrated to be feasible. Though the discussion of the impact of each choice is out of the scope of the paper, for the sake of examples four of the most relevant aspects are analyzed hereafter: the *High-Level Authentication Concept*, the *Signal Component*, the *Relative Power Level*, and the *Distribution of Chips over Time*.

The choice of a specific *High-Level Authentication Concept* has an intrinsic impact on the achievable performance and on the implied complexity at both the system and user levels. Among the concepts available in the public literature, the following ones can be considered particularly relevant:

- *Spread Spectrum Security Codes (SSSC)* [14], especially the *Public-SCA* concept, similar to the *hidden markers* proposed in [15], is one of the simplest authentication approaches working at the spreading code level. The public-SCA is based on the idea of inserting bursts on unpredictable chips, called SSSC bursts, in the open spreading code sequence. The SSSC bursts are interleaved with a *time division* approach;
- *Signal Authentication Sequences (SAS)* is an approach similar to the previous one and described in [16] and [17];
- *Supersonic Codes* [18][19] is a sophisticated concept, suitable to implement a robust authentication technique, at the cost of a remarkable increase of the complexity. It exploits the Code Shift Keying (CSK) modulation principle for authentication purposes. The basic idea is to apply circular shifts to some or all the spreading code periods (i.e. partial or complete CSK) in order to encode specific data symbols;
- new emerging solutions can also be considered, together with techniques resulting from combinations or adaptations of existing ideas. Among others, it is worth mentioning the *Chips-Message Robust Authentication* (CHIMERA), recently proposed in [20] for the GPS L1C signal.

As for the *Signal Component*, the authentication schemes can basically be implemented in two ways: by modifying an existing signal component (e.g., E1-B, E1-C, E5a, E5b) or by introducing a new one (e.g., E1-D or non-SIS channel). A new component offers more flexibility in the design of the authentication scheme, thus reducing the constraints related to the backward compatibility. In addition, in the case existing signal components are used, possible performance degradations for non-participant users have to be considered. Alternative options can include the use a *non-SIS channel* with an architecture based on a remote authentication server [20].

The choices for the *Relative Power Level* of the authentication component basically are: same power as other open components, lower power level (e.g. 1/11 of the nominal power), or variable power (i.e. amplitude modulation). Low power level options tend to increase the robustness against some spoofing attacks (i.e., requiring high gain antennas), though they affect the authentication performance of participant receivers, mainly in terms of conventional metrics as the *Time Between Authentication* (TBA), the *Time To Alarm* (TTA), and the *Time To First Authenticated Fix* (TTFAF), due to a reduction on the effective Carrier to Noise ratio (C/N_0) of the received signal.

Last but not least, the *Distribution of Authentication Chips over Time* is another relevant aspect to be properly designed. Apart the case of having a full encrypted sequence of chips (that, strictly speaking, belongs to the category of Spreading Code Encryption – SCE – and thus is beyond the SCA options), there are basically two types of distribution of the chips over time: *time division* and *time hopping*.

In the former case, dedicated slots are allocated for the SCA bursts, while in latter a randomized pattern is used for the time allocation of cryptographically generated spreading code chips. The time division was first proposed in the SSSC concept [14], while the time hopping approach, also known as "puncturing", is adopted for example in CHIMERA [20], where both the punctured code chips (i.e. "marker frames") and their insertion pattern in the spreading code (i.e. "look-up table schedule") are cryptographically generated.

Considering not only a system perspective but also a receiver point of view, it must be highlighted that the time division and time hopping options present different advantages and specific drawbacks that can be summarized as follows:

- if an existing open signal is modified, the time division offers a lower backward compatibility, since systematic portions of the signal are unsuitable to be tracked by a legacy receiver. On the other hand, a punctured code (i.e. time hopping) can still be tracked,

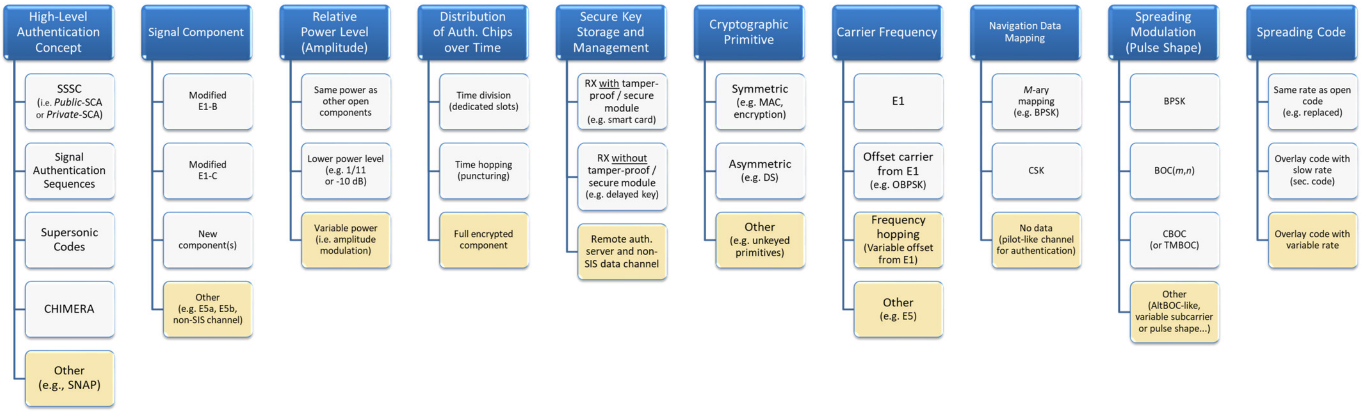
High-Level Authentication Concept
SSSC (i.e. *Public*-SCA or *Private*-SCA)
Signal Authentication Sequences
Supersonic Codes
CHIMERA
Other (e.g., SNAP)
Signal Component
Modified E1-B
Modified E1-C
New component(s)
Other (e.g. E5a, E5b, non-SIS channel)
Relative Power Level (Amplitude)
Same power as other open components
Lower power level (e.g. 1/11 or -10 dB)
Variable power (i.e. amplitude modulation)
Distribution of Auth. Chips over Time
Time division (dedicated slots)
Time hopping (puncturing)
Full encrypted component
Secure Key Storage and Management
RX with tamper-proof / secure module (e.g. smart card)
RX without tamper-proof / secure module (e.g. delayed key)
Remote auth. server and non-SIS data channel
Cryptographic Primitive
Symmetric (e.g. MAC, encryption)
Asymmetric (e.g. DS)
Other (e.g. unkeyed primitives)
Carrier Frequency
E1
Offset carrier from E1 (e.g. OBPSK)
Frequency hopping (Variable offset from E1)
Other (e.g. E5)
Navigation Data Mapping
M-ary mapping (e.g. BPSK)
CSK
No data (pilot-like channel for authentication)
Spreading Modulation (Pulse Shape)
BPSK
BOC(*m*,*n*)
CBOC (or TMBOC)
Other (AltBOC-like, variable subcarrier or pulse shape...)
Spreading Code
Same rate as open code (e.g. replaced)
Overlay code with slow rate (sec. code)
Overlay code with variable rate

Fig. 1. Overview of the main signal design aspects relevant for authentication at the spreading code level.

with an equivalent C/N_0 degradation proportional to the percentage of punctured chips [10] [20];

- a time hopping with an unpredictable insertion pattern significantly increases the complexity both at system level and receiver level. In fact, both the punctured code and the insertion pattern have to be cryptographically generated. In addition, the receiver is forced to store both open and encrypted code chips (i.e. several seconds of raw signal samples), while a deterministic time division approach allows to easily discard all the open code portions (e.g. 90% of the time), thus obtaining a significant memory saving;

- a deterministic time division shows a potentially higher vulnerability to "denial of service" attacks. In fact, if the position of the SCA bursts is known a priori, an attacker can possibly implement a *systematic jamming* of these signal portions [21]. Even if this kind of attack is easy to implement, it can be associated to a lower risk with respect to spoofing attacks, because it can be easily detected (with a failure on the authentication process) and it cannot force the victim receiver to compute a false position;

- the time hopping option can be considered as more vulnerable to specific spoofing attacks. In fact, considering the overall entropy of the punctured code chips, their distribution over long time windows (e.g. 3 minutes in the case of the "slow" channel of in CHIMERA [20]) can potentially reduce the effectiveness of the solution against specific spoofers.

Concerning the last item it must be noted that, for instance, the principle of the *State Modeling Attack* (proposed against NMA in [22]) can be generalized to work also at the spreading code level. In this case, the spoofer can try to transmit random guesses of the punctured code chips and then a-posteriori verifies the correctness of its guesses. To do so, it needs to estimate the correct code sequences with multiple high-gain directional antennas (one for each satellite in view, see section IV), or to be connected to a remote server with such resources. If the spoofer is capable to perform such verification with a limited latency (e.g. few microseconds), it can compensate possible errors in previous random guesses by increasing the transmission power of the following code chips. Aiming to maximize the success rate of the attack, it can also combine this basic idea with more sophisticated strategies, as those mentioned in [21], [22], or [23]. In this sense, a time division solution would concentrate the same entropy (at the spreading code level) in a shorter time slot, thus introducing more stringent constraints from a spoofer perspective (i.e. less time to guess/estimate/modify future code chips).

A proper design of the distribution of the chips over time requires a careful tradeoff between previous aspects and strongly depends on the specific application requirements and on the receiver constraints. Focusing on civil mass-market receivers, the time division approach offers significant savings in terms of memory and computational resources and a better robustness against specific attacks, thus it will be considered in the following discussion as a preferable option.

III. The SNAP Solution

After a brief description of the prefixed requirements of design (section III.A), the SNAP solution is presented hereafter (section III.B), together with a proposal for the possible time scheduling of the SCA bursts (section III.C).

A. Prefixed Requirements

In the design of the SNAP the main objective was twofold: provide authentication solution addressed to civil applications and receivers, and, at the same time, suitable for the evolution of the current Galileo E1 OS signal.

This means that the solution must:

- be able to raise the bar against spoofing, inserting cryptographic features at the spreading code level, hard to predict/ extract/ manipulate by attackers, thus forcing a potential spoofer to use multiple high-gain directional antennas;

- be easy to process and verify by participant receivers, e.g., mass-market devices without tamper resistant modules;

and, at the same time:

- reuse the OS NMA cryptographic data;

- offer an additional protection to the E1 OS signal against specific spoofing attacks;
- not make the E1 OS obsolete. In fact, the main goal is to provide a service capable of authenticating the current Galileo E1 OS signal, i.e.: E1-B/C. This does not necessary imply the modification of current OS components, as long as the solutions implemented through the use of other (new) signal components are capable to also authenticate the E1-B/C signals.

B. The high level concept

The design of the SNAP technique has been initially inspired from other solutions available in the state of the art, mainly focusing on the benefits of the following three approaches: the *public-SCA* concept [14], the *Supersonic Codes* [18][19], and the *Signature-Amortization* scheme. The former two are briefly described in the previous section, while the *Signature-Amortization* is an NMA concept, presented in [24], and alternative with respect to the OS NMA [25] [26].

The rationale behind the SNAP solution is to introduce segments of spreading code chips in the SIS that can be considered as *unpredictable* from a spoofer perspective, but can be easily a-posteriori verified by a participant receiver. The benefits of a CSK modulation scheme are also exploited in order to enhance the flexibility and the authentication performance of the scheme, as discussed later.

In detail, the SNAP foresees the use of two types of SCA bursts, inserted in the open pseudo-random noise (PRN) code sequence with a time division approach and at different rates, as illustrated for example in Fig. 2.

Slow SCA bursts (indicated in yellow as CSK_S in Fig. 2) allow for a robust a-posteriori verification with moderate latency (e.g., TBA of about 10 seconds). On the other hand, *fast SCA bursts* (indicated in green as CSK_F in Fig. 2) are inserted at higher rate in the PRN code sequence, being intended to reduce the TBA (e.g. to about 2 seconds) under a wide set of spoofing attacks.

Both slow and fast bursts should be cryptographically generated and a-posteriori verified by participant receivers. For the sake of simplicity but without losing of generality, the following presentation of the SNAP concept is based on two main assumptions:

- the inputs of the bursts generation are exclusively based on the re-use of the E1-B OS NMA data [25] [26], as the Timed Efficient Stream Loss-Tolerant Authentication (TESLA) key chain and Message Authentication Code (MAC) tags;
- the bursts are allocated on a generic signal component (i.e. a modified E1-B or E1-C, or a new component).

These initial assumptions are reasonable for a conservative scenario, aiming to a SNAP implementation with minimal changes and complexity increase at system/receiver level. Other relevant alternatives will be traded-off in section V.

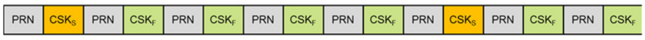

Fig. 2. Scheme for the insertion of *slow* and *fast* bursts in the PRN sequence.

The following presentation of the SNAP concept is based on the schematic sketched in Fig. 3. At the top, the figure illustrates the flow of the OS NMA data in the Galileo E1-B I/NAV message, whose basic principles are briefly recalled hereafter (for more details, please refer to [25] and [26]). At the bottom, Fig. 3 sketches the proposed concept, depicting both *slow* and *fast* bursts, together with their possible relation with OS NMA data.

The time scheduling and the relations between subsequent TESLA keys and MAC tags in a typical OS NMA configuration are also illustrated in Fig. 3, according to the specifications presented in [26]. Each I/NAV page (with a length of 2 seconds) is composed by two parts, denoted as even and odd pages parts and lasting 1 second each. Only odd page parts contain the field referred to as 'reserved 1', which has a length of 40 bits and is devoted to the transmission of NMA data. By processing such fields, the receiver is able to demodulate 40 bits every 2 seconds from each received SIS, for an equivalent data rate of 20 bits per second, dedicated to OS NMA service.

For the sake of simplicity, Fig. 3 assumes that the information (and the entropy) related to the 40 bits is uniformly spread in the time slot of each odd page part. This means that the reception of such bits will require the processing of 1 second of symbols every odd second of the Galileo System Time (GST). In fact, all the data bits in each I/NAV page part are convolutionally encoded, with rate ½, and interleaved with a block interleaver characterized by 30 columns and 8 rows [27]. This channel coding scheme makes a direct identification of each 'reserved 1' bit in the stream of SIS symbols a non-straightforward task [28]. Among the 40 bits of each 'reserved 1' field, 8 bits are dedicated to the transmission of the NMA headers and the signed root key (i.e. the HKROOT section), while the remaining 32 bits carry on the truncated MAC tags and the keys of the TESLA chain (i.e. the MACK section). According to [26], a typical NMA configuration allows transmitting a MACK section every 10 seconds. Each MACK section contains 3 MAC tags (and their related 'MAC-info' fields) and 1 key. In this example, each MAC tag consists of 10 bits, while each MAC-info field has a length of 16 bits. Thus, each MAC tag+info section has a total length of 26 bits. In a window of 10 seconds (i.e. from t_0 to $t_0 + 10$ s in Fig. 3), the 3 MAC tag+info sections are spread over about 4.4 seconds in the received satellite signal, as shown by the yellow boxes in Fig. 3. Similarly, a single TESLA key (e.g. k_{j+1} in Fig. 3) has a length of 82 bits and is spread over about 4.6 seconds.

Fig. 3 also shows that, at the time instant $\text{GST}_m = t_0$, a receiver has only received the current TESLA key k_j, together with the previous I/NAV message. At this point, it has to wait at least 9 seconds (i.e. until approximately $t_0 + 9$ s) in order to retrieve the necessary data to perform the NMA verification procedure. In fact, the receiver needs to correctly decode the 3 MAC tag+info sections and the next key k_{j+1}, then to verify the validity of k_{j+1} against previous k_j, and finally to re-

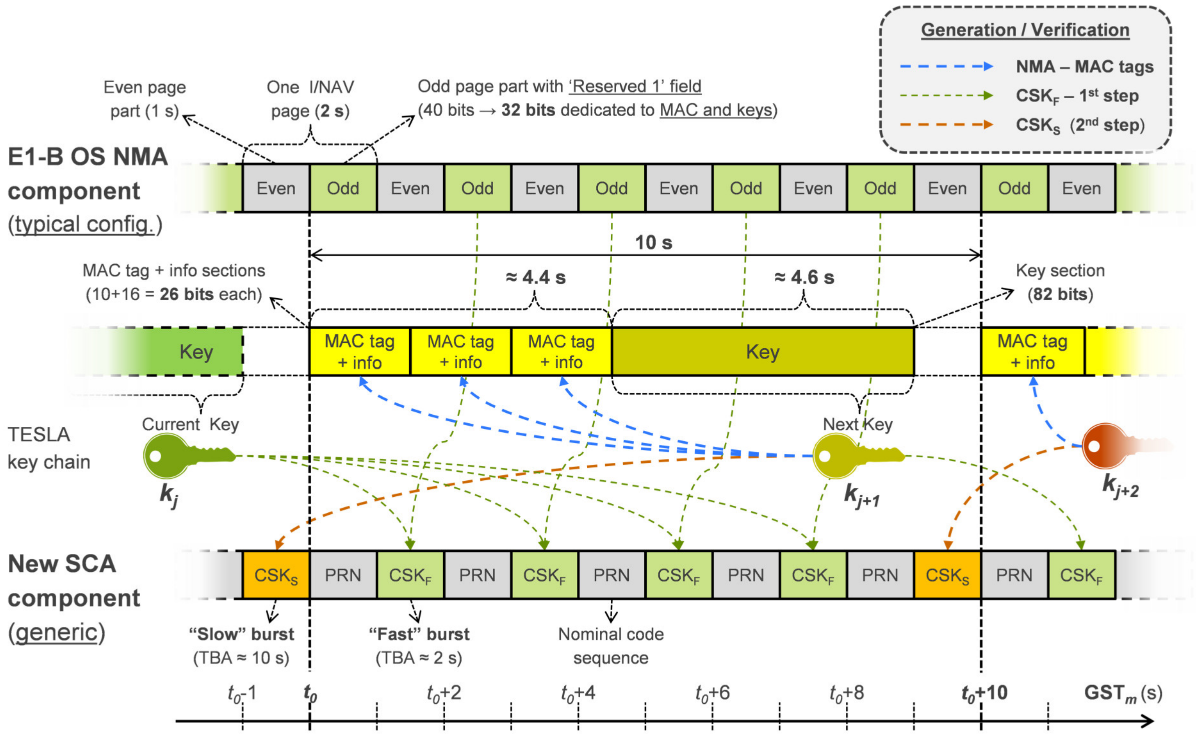

Fig. 3. Schematic of the SNAP concept: tentative time scheduling and relation between NMA bits and SCA bursts.

compute and check the 3 MAC tags with k_{j+1}, as shown by the blue arrows in the figure.

In the example, a burst (CSK$_S$ or CSK$_F$) with a length of 1 second is transmitted every 2 seconds, thus maintaining at least 1 second of nominal (predictable) code sequence, suitable to be potentially processed even by non-participant receivers (i.e., not capable/interested to process the authentication information). As anticipated in section II, the time distribution of the code chips and, thus, the length of each burst is a relevant design parameter. In principle, such time slot length can be increased up to the limit of encrypting the whole code sequence, without leaving any portion of nominal PRN sequence in the transmitted signal.

In principle, these bursts can be generated by the satellites reusing the OS NMA data (i.e. MACs and keys). The same information can be used also by a participant receiver in order to verify the received bursts, as discussed later.

An innovative aspect of the SNAP concept resides in the fact that, contrary to those authentication solutions in which the verification at the receiver is performed separately on each single channel, the SNAP is able to exploit the information received from all the in-view satellites, thus obtaining a solution suitable for a *two-steps authentication* procedure. More in detail, the fast bursts for all the satellite signals can be generated from the same cryptographically-generated spreading code chips, but a unique circular shift, that depends on the satellite identifier can be applied to each of them. In this way, the bursts received from different satellites at a given time instant consist of the same code chips sequence, just shifted in a different way for every satellite. As an example, the CSK$_F$ bursts for all the satellite signals can be generated from the same code chips, using a block or stream cipher initialized by a common cryptographic key computed as follows:

$$\text{crypto key}_m \propto \mathit{Hash}\{\, k_{j+1} \mid \text{GST}_m \,\} \qquad (1)$$

where $\mathit{Hash}\{\cdot\}$ represents a cryptographically secure hash function (i.e. a digest), k_{j+1} is the successive TESLA key, and GST_m is the current Galileo System Time. Note that the symbol ∝ is used for the sake of generality, aiming to cover also the case of a non-linear operation on the output of the hash function (e.g. a truncation).

Then, different CSK shifts can be applied to each burst, depending on the satellite identifier (Sat. ID), previous key k_j, and next NMA bits (next 'reserved 1' field, that will be disclosed 1 second later):

$$\text{shift}_m \propto \mathit{Hash}\,\{\,\text{Sat ID.} \,|\, k_j \,|\text{next 'reserved 1' field}\,\} \qquad (2)$$

Being the crypto key_m independent from the Sat. ID, the bursts received from different satellites at a given time instant GST_m consist of the same code chips sequence, just shifted in a different way for every satellite. In this way, the receiver can first cross-authenticate couples of satellite signals by applying a codeless CSK correlation between bursts from two satellites, properly shifted and aligned, i.e., *first step authentication*. It can then a-posteriori verify both *slow* and *fast* bursts, as soon as k_{j+1} will be disclosed i.e., *second step authentication*.

The concept for the working principle of the two steps authentication at the receiver level is schematically represented in Fig. 4. The receiver, before being able to verify the content

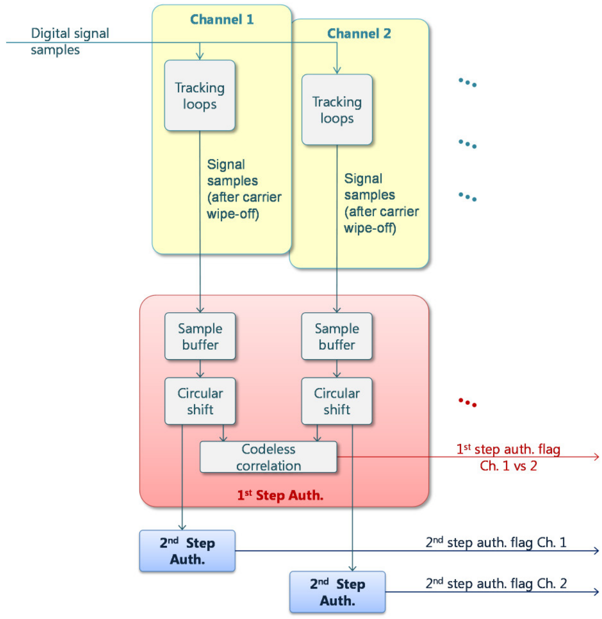

Fig. 4. Schematic concept for the 1st and 2nd step authentication at the receiver level.

of the fast bursts, can check the consistency among signals from different satellites. The bursts from couples of satellites can be stored in a buffer, properly shifted on the basis of the value of the last k_j and then correlated to verify their consistency. As obvious, a codeless correlation gives different performance respect to a classical correlation operation, and the technique might be suitable for proper working scenario conditions. At the same time, the 1st step authentication is able to considerably improve the authentication performance, reducing the TBA and TTA metrics (e.g. from 10 to 2 seconds) under different kinds of attack.

This two-step procedure is tailored to the CSK$_\text{F}$ bursts, while a simplified approach can be used for the generation and the verification of the CSK$_\text{S}$ bursts (e.g. similar to the Public-SCA concept in [14]). In this case, a different code burst can be transmitted by each satellite, for example simply using a different key k_{j+1} or by adding the Sat. ID as an additional input of the hash function in (1). In this way, at the receiver side, the information related to crypto keys and CSK shifts will be retrieved from the NMA bits, thus achieving a TBA of about 10 seconds. In principle, the authentication latency of some CSK$_\text{S}$ bursts can potentially be extended to several minutes, periodically introducing a delayed release of the TESLA key with some similarities to what done in the case of “slow MACs” in [26].

C. A tentative time schedule and verification procedure

Referring again to Fig. 3, we present here a first practical example with a high-level implementation of the SCA concept.

At the time t_0 a CSK$_\text{S}$ burst has just been received. At this point, in order to verify it, the receiver must wait to correctly decode the next key k_{j+1} from the OS NMA component. Thus, the authentication latency between the beginning of a CSK$_\text{S}$ burst (i.e. $t_0 - 1$ s) and the last bit of k_{j+1} (approximately $t_0 + 9$ s) is consistent with the target TBA of 10 seconds. Such time scheduling for the reception and the a-posteriori verification of CSK$_\text{S}$ bursts also repeats in following time windows, as shown by the orange arrows of Fig. 3.

As for the CSK$_\text{F}$ bursts, a different approach is adopted for their generation, enabling the two-steps authentication. At the time t_0, the key k_j has just been disclosed, while the next key k_{j+1} is still unpredictable from a spoofer point of view. For this reason, the code chips of the following 4 CSK$_\text{F}$ bursts can be generated by means of a (block or stream) cipher, initialized by a crypto key computed as described by equation (1), and circularly shifted by following equation (2).

At the time $t_0 + 2$ s a CSK$_\text{F}$ burst has just been received from each satellite in view. In order to perform the 1st step verification, the receiver must wait until the next 'reserved 1' field will be received in an odd I/NAV page part (i.e. until about $t_0 + 3$ s). At this point, the shift of each CSK$_\text{F}$ burst can be recomputed, and the burst shifted back. Since they have been generated starting from the same (yet unknown) code sequence, they can finally be used in a *codeless correlation* to check the consistency of bursts received from pairs of satellites, as shown in Fig. 4.

In this case, the authentication latency between the beginning of a fast burst (i.e. $t_0 + 1$ s) and the last 'reserved 1' bit needed for the codeless verification (approx. $t_0 + 3$ s) leads to a TBA of 2 seconds. Such 1st step verification procedure is repeated for the 3 subsequent CSK$_\text{F}$ bursts (i.e. from $t_0 + 3$ s to $t_0 + 8$ s), as shown by the green arrows of Fig. 3.

The next key k_{j+1} can be considered as completely disclosed only at time $t_0 + 9$ s, when the receiver is able to verify the CSK$_\text{S}$ burst previously received from $t_0 - 1$ s to t_0. In addition, at $t_0 + 9$ s, the receiver is able to implement the 2nd step verification also on each previous CSK$_\text{F}$ bursts, using k_{j+1}, to locally generate an exact replica of the bursts and verify them by means of a classical (non-codeless) correlation.

As a final remark, it is worth mentioning that the verification procedures for the CSK$_\text{S}$ bursts and the 2nd step check on the CSK$_\text{F}$ bursts mainly take advantage by the cryptographic robustness of the TESLA chain of keys. On the other hand, the 1st step verification of CSK$_\text{F}$ bursts is based on all the 'reserved 1' bits, thus using not only the TESLA keys, but also the MAC tags. In fact, the 3 MAC tags received approximately from t_0 to $t_0 + 4.4$ s can be considered unpredictable until the disclosure of k_{j+1}.

IV. ROBUSTNESS AGAINST DIFFERENT THREATS

When designing an authentication scheme, it is essential to define potential threats and attack scenarios the technique shall be able to cope with. Considering recent results from the scientific literature [9][10], different families of attacks have been identified as likely future threats for several applications based on GNSS. They are listed hereafter, sorted according to an increasing associated cost and complexity, i.e. from low-cost attacks, more likely to be implemented, to the most expensive ones, with limited feasibility on a large scale:

- *Meaconing-like attacks* are based on the reception and rebroadcasting of an entire block of radio frequency

(RF) spectrum containing an ensemble of received GNSS signals, without distinctions between different satellite signals. A basic attack belonging to this category can be easily implemented with a receiving antenna, connected to an amplifier and to a transmitting antenna (i.e. signal repeater). It is mostly applicable to a stationary victim only, being quite complex or impractical to move the meaconer in a plausible way in dynamic conditions.

- *Simplistic* or *Intermediate spoofing (without directional antennas)*: in this case the attacker is able to generate counterfeit GNSS signals, not necessarily reflecting any information on the current broadcast signals. It can be put in practice by using low cost hardware for receiving and replaying the GNSS signals, potentially including customized software-defined signal simulators/synthesizers in order to control and/or modify some of the signals parameters, or by using commercial hardware RF simulators, that are normally expensive and moderately complex to use. In the case of *intermediate* attacks, the spoofer can implement specific approaches to perform an *on the fly* estimation-and-replay of each symbol of the navigation message, i.e., *Security Code Estimation And Replay* (SCER) [29]. In addition, the spoofer can also take advantage of some kind of *prediction*, as in the case of the *Forward Estimation Attack* (FEA) or the *State Modeling Attack* (SMA) [22].
- *Sophisticated attacks with (multiple) high-gain directional antenna(s)*: these attacks are based on the use of one single antenna or multiple antennas with enough gain to directly estimate and spoof all the signal components from a single or multiple GNSS satellites. In detail, each antenna is pointed toward a satellite and, thanks to the directional gain of the antenna, it is possible to raise unknown or encrypted code chips beyond the noise floor, to directly read and rebroadcast them with limited latency. These attacks are clearly more expensive and complex, due to the high costs related to the high gain antennas and complexity associated to the attack setting-up.

Meaconing-like and *sophisticated* attacks with directional antennas can be considered as threats beyond of the scope of the SIS design and/or not relevant for directly assessing the robustness of NMA and SCA solutions, since they also require specific countermeasures at the receiver level.

On the contrary, for *Simplistic* or *Intermediate attacks*, the SNAP technique is able to increase the spoofing robustness of the basic OS NMA verification, by making impractical or detectable several specific types of attacks. The fact that the data message and spreading code solutions work interconnected to each other further protect the authenticated data bits, thus increasing the overall level of security, as also highlighted for example in [20].

V. Trade-off Analysis and Parameters Definition

As said, the implementation of an authentication concept is open to several choices and optimizations, mainly related to specific signal characteristics. It is worth remarking that the achievable performance of the proposed SCA concept strongly depends on specific implementation requirements and design choices.

After briefly recalling the used methodology in section V.A, section V.B describes trade-off analysis and section V.C summarizes the consequent recommendations.

A. Used Methodology: the Driving Criteria

The methodology used to trade-off all the available signal options takes into account both quantitative results from simulations (e.g. as those already presented by the authors in [10]) and qualitative analyses based on the maximization and harmonization of three main criteria. In details:

- first of all, the *authentication performance* has been taken into account, in order to assess the technique mainly in terms of two metrics, the TBA and the TTA;
- in addition, the criterion of *spoofing robustness* has been used to measure the level of resilience against a set of specific spoofing attacks, considered significant for applications based on the open services;
- the third criteria, referred to as *current signal valorization*, has been adopted to assess the level of reuse and valorization of the current Galileo OS signal and messages structures.

In the definition of the authentication method, the three criteria have not been considered separately, nor maximized independently from each other. On the contrary, the methodology we followed tried to balance all the criteria and find a solution, suitable for the evolution of the Galileo OS signal, and able to achieve competitive performance in terms of authentication requirement and spoofing robustness.

B. Trade-off analysis

On the basis of the three introduced criteria, this section analyses the trade-off for four specific signal characteristics: *1) Inputs for the SCA bursts generation*, *2) Signal component for the SCA bursts allocation*, *3) Type of allocation of SCA bursts*, and *4) Power level of each component carrying authentication features*.

1) Inputs for the SCA bursts generation.

In principle, the SCA bursts can be generated starting from different inputs. In fact, the needed cryptographic keys and/or seeds can be obtained by using:

- only OS NMA bits (TESLA keys and MACs);
- only bits transmitted on a new data signal component (e.g., CSK modulated);
- an intermediate solution, using bits from OS NMA plus few additional bits from a new component.

All the three solutions present good *authentication performance*. In the case the inputs for the SCA burst generation are only OS NMA bits, it is possible to obtain a reduced TBA (e.g. 2 s), and a TTA limited to 10 s (worst case).

In the case an additional data component is used, the authentication performance can be further enhanced. As for an example, longer keys (e.g., more than 82 bits) can be transmitted on the new component and/or a higher data rate can be achieved (e.g., by means of a CSK modulation). In this case, the 1st step authentication can potentially achieve TTA ≈ TBA ≈ 2 s, or even less.

As presented in section IV, the solution based on the use of OS NMA bits only is able to make impractical and/or detectable several *spoofing attacks*. The use of bits from a new data component might be able to provide *robustness* also against the specific attacks based on the transmission of arbitrary NMA bits, consistent with the CSK_F bursts. On the other hand, it has to be considered that the first option allows for the further protection of the NMA bits, used as SCA inputs.

Finally, the use of OS NMA bits (OS NMA only or intermediate solution) adds *value to the current signal design* and, in addition, NMA bits will be intrinsically protected by SCA.

2) Signal component for the SCA bursts allocation.

SCA bursts, both fast and slow, can be allocated on different signal components:

- on a modified version of the E1-B component;
- on a modified version of the E1-C component;
- on the new signal component E1-D.

Solutions based on the use of E1-B or C have to consider potential *performance* degradations for non-participant receivers. More in details, in the case of bursts inserted only on E1-B, data demodulation performance is typically affected (if the tracking is done on E1-C), while, in the case of bursts only on E1-C, the tracking performance is affected. On the opposite, solutions based on the new component E1-D present potential higher flexibility for implementing both NMA and SCA. In fact, they present less constraints on SCA burst length, the SCA latency is potentially lower (not constrained by E1-B OS NMA data rate bottleneck), and the values of TBA, TTA, and TTFAF can be potentially reduced.

In the case of authentication features only on E1-B (NMA and SCA on the same component), a reduced *spoofing robustness* can be experienced, since the spoofer is only forced to attack a single component. On the contrary, in case of E1-C or E1-D, authentication features on separated components might increase the spoofing robustness.

The higher values of *current signal valorization* can be obtained by using a modified E1-B or E1-C component, but with a possible impact on its backward compatibility.

3) Type of allocation of SCA bursts.

Fast and slow SCA bursts can be allocated:

- on a single component (joint allocation);
- on multiple signal components (separated allocation).

Solutions based on separated allocation of fast and slow bursts are able to optimize the burst scheduling, thus enhancing the *authentication performance*.

In the case the SCA bursts are split on multiple signal components, the spoofer might be forced to attack each of them, modifying and making consistent each authentication feature. For this reason the *spoofing robustness* is considered higher in this case.

The *level of valorization* depends on which are the signal components used to allocate the SCA bursts, rather than on the choice of using single or multiple components. Nonetheless, a solution based on multiple components would potentially reuse/valorize at least one existing signal component.

4) Power level of each component carrying authentication features.

For instance, the power levels of each signal component carrying authentication features can be:

- at nominal value (referred to as 0 dB);
- at -10 dB from nominal power.

As obvious, solutions at nominal power present higher *authentication performance*. For example, signal components at -10 dB are not suitable for the allocation of fast bursts, leading to an excessive increase on the required burst length in order to be compatible with the 1st step codeless check.

On the other hand, low power signals are generally more *robust to spoofing attacks*, since they force a higher cost/complexity at the attacker side. In the case of a SCA component at -10 dB, in fact, a spoofer that wants to read on-the-fly the values of the code chips needs an antenna with a gain of 10 dB higher.

Finally, signal components at -10 dB would better fit with the current *signal structure*. For example, in this case, there is the possibility to *reuse* one or more E1 OS components, as the BOC(6,1) portion of the CBOC modulated signals.

C. Recommendations

The analysis on the threats robustness together with that that on the parameters trade-off allows for the drafting of the following list of recommendations, thus helping in the definition of an implementation option (see section VI).

First of all, a SCA approach can offer *additional protection* to the OS NMA concept for Galileo E1-B. This is motivated by the *time binding* concept, recently mentioned in [20]. A joint NMA + SCA approach implies that authenticated navigation data and spreading code chips are bound together, thus forcing the attacker to contemporaneously spoof both of them. A joint approach, in fact, is able to detect specific attacks undetectable by NMA alone and/or improve the overall authentication performance (e.g., reducing the TBA and TTA from 10 to 2 s).

The *transmission of SCA bursts and the delayed release of crypto keys via NMA* must happen in *well separated time slots*. This is necessary to limit the feasibility of attacks exploiting *on the fly estimation* or *prediction* of NMA symbols, as for attacks like SCER, FEA, or SMA.

It is recommended to use the possible *additional capacity on a new data component* (e.g. E1-D) not to further reduce the TBA (by a faster key release), but to *further protect the E1-B OS NMA data*, for example allowing to detect and mitigate data demodulation errors on the I/NAV in case of degraded signal conditions (e.g. low C/N_0 in urban scenarios).

Specific design choices can also offer the possibility of a *multi-step synchronization and authentication procedure*, suitable to an *incremental adoption/exploitation* and to *relax the receiver synchronization requirements*.

Finally, it is recommended to adopt a solution based on *two signal components*, that offer different power levels and frequency diversity, thus enhancing the robustness against specific spoofing attacks. One component at nominal power level to assure high authentication performance, and one low power component to constrain possible spoofers to use high-gain directional antennas.

VI. A Possible Working Point

This section presents an example of implementation option, following referred to as *working point*, for the SNAP solution.

The schematic of the power spectral density (PSD) of the designed signal is represented in Fig. 5, together with the allocation of the authentication bursts.

The working point foresees the use of two new signals components (in addition to the existing E1-B and E1-C):

- E1-D: data channel at nominal power level, possibly located at an offset carrier with respect to the E1 carrier frequency. It can be used in time multiplexing, for alternatively carrying bursts and cryptographic data for the bursts verification;
- E1-E: low power component (e.g. -10 dB w.r.t. other signal components), possibly located around E1 carrier frequency. It is not compatible to data demodulation, thus potentially carrying a fully encrypted spreading code sequence (or open and encrypted codes time multiplexed).

As for the four signal characteristics traded-off in section V.B, the SNAP working point proposes to have:

1) Inputs for the SCA bursts generation:

Input data from E1-B OS NMA plus additional data from the E1-D data channel.

2) Signal component for the SCA bursts allocation:

Slow bursts on the E1-E component, and *fast* bursts on the E1-D component. A Binary Offset Carrier BOC(6,1) modulation can be used for the slow bursts, while an Offset Binary Phase Shift Keying (OBPSK) modulation can be adopted for the fast bursts.

3) Type of allocation of SCA bursts:

Fast and slow bursts are allocated on multiple signal components.

4) Power level of each component carrying authentication features:

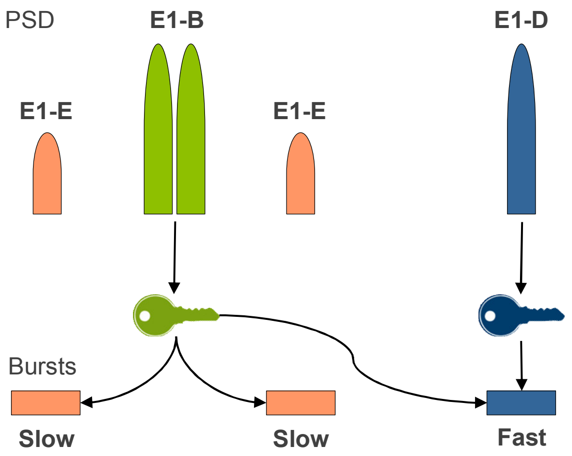

Fig. 5. Possible working point: schematic of the power spectral density of the signal components and allocation of the bursts.

The two signal components are at different power levels: slow bursts on E1-E at -10 dB, and fast bursts on E1-D at nominal power level. The first component is intended to constrain possible spoofers to use directional antennas with a very high-gain, while the second component is suitable to speed up the authentication performance by means of fast bursts.

The SNAP solution implemented in this way is capable to provide two levels of service, depending on possible receiver implementation profiles. In the case a low-complexity processing is required, the receiver can only use E1-B and E1-E components: the authentication information will be retrieved from the OS NMA bits of E1-B and used to verify the slow bursts on E1-E. Obviously, this profile will give limited authentication performance, leading to a TBA of the order of 10 seconds. On the other hand, a full complexity receiver, able to use E1-B, E1-E, and E1-D will provide a high performance service, with TBA of around 2 seconds.

In addition, SNAP is potentially compatible to alternative and intermediate implementation profiles, suitable to a medium complexity implementation. The receiver in fact can use E1-B plus E1-E to verify the slow burst and E1-D only to perform the 1st step authentication, with the codeless correlation on the fast bursts. This implementation profile will lead to an intermediate level of performance: the TBA is again of the order of 2 seconds, though a possible availability degradation has to be considered in low C/N_0 conditions.

VII. Conclusions and Future Work

The SNAP concept is an innovative solution for the authentication of both navigation data bits and spreading code chips, designed to be compatible with an evolution of the Galileo E1 OS signal. In principle, it can be potentially adapted also to other civil GNSS signals. The significance of the work mainly lies in three elements.

First, though the technique is innovative and able to achieve predefined authentication performance, it exploits the structure of the legacy Galileo signal and the characteristics of the OS

NMA, thus being capable of increasing the spoofing robustness.

Second, the work investigates the performance of the solution under different families of spoofing attacks. The advantages of the SNAP are described against solutions entirely based on the authentication of the navigation data components only, and those techniques that implement both NMA and SCA, but leave them independent from each other.

Third, the two-steps authentication concept allows the receiver to adapt its authentication verification process, depending on specific requirements and conditions. For example, receivers used for applications with low authentication requirements might decide to only verify the first step authentication, while more demanding users might implement the full two-step process, at a cost of an increased complexity within the receiver.

A. Open Points and Future Activities

The work presented in the paper gives several cues for future studies, concerning both the system and receiver levels. As for the formers, example of activities focused on the signal design are:

- the investigation of the use of an additional data channel (non-SIS) for authentication purposes, as also outlined in [20];
- the study of alternative/complementary authentication solutions, based on specific design parameters (e.g., variable waveform, frequency hopping, variable power level, etc.).

On the other hand, strategies for the verification of the authentication information at the user side can be further studied and optimized:

- as an example, specific approaches that exploit information available within the receiver (e.g., C/N_0) for the verification of authentication information should be investigated. Such algorithms would allow the receiver to output not only a binary information (i.e. authentic signal or not), but also an associated confidence interval;
- in addition, another important aspect to consolidate concerns the achievable performance of both 1st step and 2nd step authentication verification in terms of detection and false alarm probabilities (i.e. Receiver Operating Characteristic – ROC curves). Preliminary simulations have been already carried out in Additive White Gaussian Noise (AWGN) channel, confirming the feasibility of the SNAP concept and showing promising performance results. In any case, future activities should include the validation of the concept, also with extensive simulations covering more realistic channel models (e.g. presence of multipath, fading, and dynamic conditions);
- furthermore, additional studies should be conducted introducing refinements able to strengthen the solution availability also in degraded scenarios. For example, a simple detection strategy based on the mutual correlation among a set (instead of pairs) of satellites signals shall be investigated. This can limit the impact of false alarms on single channels, thus improving the overall performance.

Acknowledgment

This article will present some findings of a project funded by the European Commission under the Horizon 2020 Framework Program (Funding Reference No. 435/PP/GRO/RCH/15/8384).

In addition, the authors want to thank Airbus Defence and Space GmbH (project prime contractor) for their constructive comments, helpful to improve the analysis.

References

[1] Anon., “Vulnerability assessment of the transportation infrastructure relying on the global positioning system,” J.A. Volpe National Transportation Systems Center, 2001.

[2] M. Jones, “Spoofing in the Black Sea: What really happened?,” GPS Word website, October 2017. Available at: http://gpsworld.com/spoofing-in-the-black-sea-what-really-happened/

[3] S. Pullen, G.X. Gao, “GNSS Jamming in the name of Privacy,” Inside GNSS, Vol. 7, No. 2, March/April 2012.

[4] J. C. Grabowsky, “Personal privacy Jammers. Locating Jersey PPDs Jamming GBAS Safety-of-Life Signals,” GPS World, Vol. 23, No. 4, April 2012.

[5] M. L. Psiaki, T. E. Humphreys, and B. Stauffer, “Attackers can spoof navigation signals without our knowledge. Here's how to fight back GPS lies,” IEEE Spectrum, Vol. 53, Issue 8, August 2016.

[6] D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, “Drone Hack: Spoofing Attack Demonstration on a Civilian Unmanned Aerial Vehicle,” GPS World, August 2012.

[7] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, “GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques,” International Journal of Navigation and Observation, vol. 2012, Article ID 127072, 16 pages, 2012. Doi:10.1155/2012/127072.

[8] M. L. Psiaki and T. E. Humphreys, “GNSS Spoofing and Detection,” Proceedings of the IEEE, Vol. 104, No. 6, pp. 1258-1270, June 2016. Doi: 10.1109/JPROC.2016.2526658.

[9] F. Dovis, GNSS Interference Threats and Countermeasures. Artech House, Norwood, MA, Jan. 2015, ISBN 9781608078103.

[10] D. Margaria, B. Motella, M. Anghileri, J. J. Floch, I. Fernandez-Hernandez, and M. Paonni, “Signal Structure-Based Authentication for Civil GNSSs: Recent Solutions and Perspectives,” in IEEE Signal Processing Magazine, Vol. 34, No. 5, pp. 27-37, Sept. 2017. Doi: 10.1109/MSP.2017.2715898.

[11] European GNSS (Galileo) open service. Signal-in-space interface control document. OS SIS ICD, Issue 1.3, Dec. 2016.

[12] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simon, I. Rodríguez, and J. David Calle, “A Navigation Message Authentication Proposal for the Galileo Open Service,” NAVIGATION, Journal of The Institute of Navigation, Vol. 63, No. 1, Spring 2016, pp. 85-102.

[13] GSA website, “Assuring authentication for all”, Aug. 3, 2016. Available at: https://www.gsa.europa.eu/news/assuring-authentication-all

[14] L. Scott, “Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems,” Proc. of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003), Portland, OR, Sept. 2003, pp. 1543-1552.

[15] Kuhn, M. G., “An Asymmetric Security Mechanism for Navigation Signals,” Proc. of the 6th Information Hiding Workshop, 2004, pp. 239–252.

[16] O. Pozzobon, L. Canzian, M. Danieletto and A. D. Chiara, "Anti-spoofing and open GNSS signal authentication with signal authentication sequences," Proc. of 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC 2010), Noordwijk, Dec. 2010, pp. 1-6.

[17] O. Pozzobon, "Keeping the Spoofs Out – Signal Authentication Services for Future GNSS," Inside GNSS, May/June 2011, pp. 48-55.

[18] O. Pozzobon, G. Gamba, M. Canale, S. Fantinato, "Supersonic GNSS Authentication Codes," Proc. of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, Florida, Sept. 2014, pp. 2862-2869.

[19] O. Pozzobon, G. Gamba, M. Canale, S. Fantinato, "From Data Schemes to Supersonic Codes – GNSS Authentication for Modernized Signals," Inside GNSS, pp. 55-64, Jan./Feb. 2015.

[20] J. M. Anderson et al., "Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals," Proc. of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS + 2017), Portland, OR, Sept. 2017, pp. 2388-2416.

[21] J. T. Curran, M. Bavaro, P. Closas, and M. Navarro, "On the Threat of Systematic Jamming of GNSS," Proc. of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Portland, OR, Sept. 2016, pp. 313-321.

[22] J. T. Curran and C. O'Driscoll, "Message Authentication as an Anti-Spoofing Mechanism," Working Paper, June 2017. Available at: https://www.researchgate.net/publication/317950338_Message_Authentication_as_an_Anti-Spoofing_Mechanism

[23] G. Caparra, S. Ceccato, N. Laurenti, and J. Cramer, "Feasibility and Limitations of Self-Spoofing Attacks on GNSS Signals with Message Authentication," Proc. of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017), Portland, OR, September 2017, pp. 3968-3984.

[24] G. Caparra et al., "Design Drivers and New Trends for Navigation Message Authentication Schemes for GNSS Systems," Inside GNSS, pp. 64-73, Sept./Oct. 2016.

[25] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simon, I. Rodríguez, and J. David Calle, "A Navigation Message Authentication Proposal for the Galileo Open Service," Navigation, Journal of The Institute of Navigation, vol. 63, no. 1, 2016, pp. 85-102.

[26] European Commission, Galileo Navigation Message Authentication Specification for Signal-In-Space Testing. Ref. ARES(2016)6620281, v1.0, 25/11/2016.

[27] European Union, European GNSS (Galileo) Open Service Signal In Space Interface Control Document. OS SIS ICD. v. 1.3, Dec. 2016.

[28] I. Fernández-Hernández and G. Seco-Granados, "Galileo NMA signal unpredictability and anti-replay protection," 2016 International Conference on Localization and GNSS (ICL-GNSS), Barcelona, 2016, pp. 1-5. doi: 10.1109/ICL-GNSS.2016.7533686.

[29] T. E. Humphreys, "Detection Strategy for Cryptographic GNSS Anti-Spoofing," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073-1090, April 2013. doi: 10.1109/TAES.2013.6494400.