A Navigation Message Authentication Proposal for the Galileo Open Service

IGNACIO FERNÁNDEZ-HERNÁNDEZ
European Commission, Belgium

VINCENT RIJMEN
University of Leuven (KU Leuven), Belgium

GONZALO SECO-GRANADOS
Universitat Autònoma de Barcelona, Spain

JAVIER SIMON
European GNSS Agency, Czech Republic

IRMA RODRÍGUEZ and J. DAVID CALLE
GMV, Spain

ABSTRACT: *GNSS vulnerabilities have become evident in the last decade. Authentication of the GNSS signals and data can be an important building block contributing to mitigating these vulnerabilities. This paper presents a Navigation Message Authentication (NMA) scheme based on the Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol and a novel concept based on a single one-way chain for all senders and cross-authentication. The paper presents an NMA implementation in the Galileo Open Service (OS) navigation message that should provide similar navigation performance to data-authenticated users and standard non-authenticated users in terms of time to first fix, accuracy, and availability even in difficult reception conditions. The proposal also maintains a high level of signal unpredictability to help receivers protect against replay attacks. The scheme and implementation proposed yield significant improvements compared to the state of the art, offering the opportunity for Galileo to become the reference GNSS in civil navigation authentication.*

MOTIVATION AND CONTEXT

The need to mitigate GNSS vulnerabilities has become evident in recent decades [1], as highlighted by articles in this Journal over the past years [2, 3]. In particular, a recent survey of spoofing countermeasures [3] proposed the addition of cryptographic protection of the navigation message among other solutions such as inertial sensors, stable clocks, or antenna arrays. Navigation Message Authentication, or NMA, generally refers to the authentication of the navigation data broadcast by a GNSS [4]. Civil users receiving open signals can authenticate GNSS data through a digital signature [2], if their receivers have an authentic public key to validate the signature. Authentication can also follow hybrid (symmetric/asymmetric) approaches relying on time-delayed asymmetry, e.g., through the TESLA (Timed Efficient Stream Loss-tolerant Authentication) protocol [5], as proposed in this article. In addition to data authentication, previous literature proposes and characterizes detection methods of replay attacks based on the unpredictable bits or symbols of NMA [3, 6].

The Galileo signal design and message structure is suitable for introducing authentication, as it allows higher bitrates compared to other GNSS [7, 8] and, due to the Galileo safety-of-life service re-profiling a significant amount of bandwidth has been liberated for other uses.

The main problem targeted in this article is how to provide an NMA service that maximizes availability and robustness within the Galileo I/NAV E1-B Open Service signal, in a fully backward-compatible way with the current Signal In Space Interface Control Document (SIS ICD) [8], and with minimum changes to the deployed infrastructure.

In 2014, a proof-of-concept platform developed and tested a Galileo NMA solution with the real SIS, verifying its feasibility [9]. At the time of writing this article, new Galileo requirements to include NMA are being studied for incorporation

NAVIGATION: *Journal of The Institute of Navigation*
Vol. 63, No. 1, Spring 2016
Printed in the U.S.A.

in the Galileo technical baseline, which should allow provision of NMA before the end of this decade. While the work presented in this article has influenced this process, the final Galileo NMA implementation may still be subject to modifications in the years to come.

PERFORMANCE MEASURES

As a general principle, an optimally designed NMA solution should not degrade navigation accuracy or availability and should minimize the difference between Time To First Authenticated Fix (TTFAF, i.e., the time to calculate a first position based on data-authenticated satellites) and standard, non-authenticated Time To First Fix (TTFF). The impact of NMA on standard navigation performance indicators such as availability or accuracy is presented in [10] and [11]. There it is explained that the navigation performance and robustness against attacks are strongly dependent on Authentication Error Rate (AER) and Time Between Authentications (TBA). AER represents the probability of error of authenticating a satellite in the absence of attacks, but in the presence of disturbances in the transmission channel. It is therefore equivalent to packet error rate, where the packet comprises the navigation and authentication information required to perform the authentication verification.

$$AER = 1 - (1 - BER)^{NNA} \tag{1}$$

where *BER* is the bit error rate and *NNA* is the number of bits used in the authentication, which includes the plaintext navigation bits to be authenticated (*NN*), and the authentication bits (*NA*). TBA represents the time between two successive satellite authentications. When satellites are authenticated at different times (as proposed in [2] and later in this article), a distinction should be made between *all-in-view* TBA, i.e., the time between authentications of *any* satellite in view by a receiver, and *single-channel* TBA, i.e., the time between authentications of a given satellite. Unless specifically stated, this paper refers by default to the latter.

TBA influences the TTFAF and bounds the time during which a user can be spoofed without it being reported by the authentication system. For example, a TBA of 10 seconds implies that any attack altering the unpredictable cryptographic information transmitted by a satellite can be detected with a maximum delay of 10 s.

Together with AER and TBA, Maximum Predictable Time (MPT) and Unpredictable Symbol Ratio (USR) can characterize the level of protection against signal replay attacks. MPT represents the maximum time that the signal will be predictable. USR, in the context of this work, reflects the percentage of unpredictable symbols out of the total number of symbols over a given time period. USR gives an indicator of how long a receiver has to wait before having a reliable test statistic to protect against replay attacks.

Based on these performance measures, the problem treated in this work can be restated as how to design an NMA scheme that does not degrade performance and minimizes AER and TBA, while maximizing the probability of detecting a replayed signal by a receiver implementing the appropriate anti-replay checks.

PROPOSED AUTHENTICATION CONCEPT

This section presents the authentication concept on which the proposed implementation is based. It relies on two main principles:

- The authentication of data from some satellites by other satellites, or *cross-authentication*, as described in [10].
- The use of different keys from different satellites but from a single one-way chain shared by all satellites, through a Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol [5].

The TESLA Protocol Features

The main properties of the TESLA protocol make it very suitable for radionavigation authentication. These properties are [5] are as follows:

- Low computation overhead for the generation, and mainly the validation, of authentication information.
- Low communication overhead. This is a critical property for AER and TBA.
- Strong robustness to data loss, as is the case for GNSS receivers in environments with reduced visibility.
- Optimal for multicast (one-to-many) transmissions, as is the case for GNSS.

TESLA is based on the transmission of a Message Authentication Code (MAC) to authenticate the plaintext message and the delayed transmission of the key used to compute the MAC. This key belongs to a chain generated through a one-way function F. The chain starts with a random seed key K_n, which is secret, and ends with a root key K_0 that is public and certified as authentic:

$$K_0 = F^n(K_n) \tag{2}$$

where F^n means recursively applying n times the function F. A hash function (e.g., SHA-256 [12]) is a one-way function, so each element of the chain can be constructed by hashing the previous element. The idea of defining password chains, on which the TESLA protocol is based, was first described in [13]. In

addition to the obvious requirement that the cryptographic functions used must be cryptographically secure, two assumptions are required for TESLA to provide security: First, the receiver must be loosely synchronized with the transmitter. Second, another authentication system is required at start-up: The receiver is required to have an authentic root key.

The cryptographic strength of the chain generation mechanism is greatly increased by breaking the symmetry of the iterations. This can be done by adding additional information to the hashing process that is known to the receiver, e.g., a counter or a time tag. In addition, the key size can be reduced by truncating the hash output, in order to reduce communication overhead. In this article, the function F is proposed as follows:

$$F(K_m,\ GST_j) = trunc(len, hash(K_m \,||\, GST_j)) \tag{3}$$

where K_m is the key used as an input to the function, GST_j is the Galileo System Time associated with the beginning of the *authentication frame j* in which the key will be applied, *trunc* is the truncation function to the *len* most significant bit (MSB), *len* is the length of the key, *hash* is the hash function used, and || is the operator to concatenate K_m and GST_j into a single bit chain. An authentication frame here refers to the interval of time during which new authentication information (MACs and key) is transmitted. At the end of an authentication frame, the receiver can perform a new authentication check.

Adding GST_j to the hashing process protects against pre-computation attacks. In such attacks, an attacker would pre-compute and store long hash chains, and when one of the broadcast chain elements appears in the precomputed chains, the attacker would have knowledge of the following keys. GST_j is different for each iteration of the hashing process. By adding it to the key computation process, pre-computation attacks, which otherwise would seem to have the highest probability of success, are neutralized, since attackers are forced to specify in advance for which time slot they are performing the pre-computations. Other options to protect against pre-computation attacks are increasing the key size, with the consequent increase in communication overhead, or associating an unpredictable pattern with each chain and transmitting it just before the beginning of the chain entry into operation. Both options are described in more detail in [14].

GNSS authentication through TESLA would be performed in the following way:

- The receiver receives the navigation data and the MAC.
- The receiver later receives a key from which the MAC can be generated.
- The receiver authenticates the key with a previous key from the chain that is considered authentic, or the root key, by performing function F the required number of times.
- The receiver re-generates the MAC with the key and the data, which should coincide with the previously received MAC.

More details about TESLA protocol implementations for GNSS can be found in [4, 11, 15–17]. Now that the TESLA protocol and its use for GNSS have been generally described, the next sections will deal with the specificities of the proposed scheme.

TESLA with a Single Chain from Several Senders

In a standard TESLA approach, as presented in the consulted literature, each sender (satellite) uses a different one-way chain. If a receiver authenticates four satellites, it should receive, in addition to the data, four MACs and four keys, one from each satellite, where each key belongs to a different chain and needs a different root key to be validated. On the contrary, the proposed scheme uses a single one-way chain for all senders [18]. The main motivation for this choice is to drastically reduce AER: by allowing all satellites to be authenticated through the same chain, a user needs to receive only one key from any satellite, and four MACs, to authenticate all satellites. Combined with the high MAC truncation explained later, this dramatically reduces the number of bits required for calculating a PVT using data-authenticated satellites. In addition, a single chain will also help in the initialization, as only one root key is required for all satellites, reducing the time to first authenticated fix (TTFAF). The use of a single chain is especially useful when one or a few satellite signals are received in good condition while others are received at lower elevation angles or subjected to multipath or blockage and therefore demodulated with a much higher bit error rate, which may be the case in urban environments. Note that even if, due to shadowing or fading, no key is successfully demodulated from any satellite for a certain authentication frame, any key from the next authentication frame can be used to verify the previous ones.

TESLA with a Single Chain and Different Keys from Different Senders

One problem that arises when using a single one-way chain is that if the same key is used by all satellites, even if it is transmitted at the same reference time, it will be received at different times by users due to satellite clock offsets and, principally, different times of arrival. As a result, only the key arriving from one satellite is unpredictable, which

makes the signals from other satellites easier to replay. This problem can be overcome by transmitting different keys from different satellites, but still from the same chain. In this way, the keys would still be unpredictable while the MAC key is recoverable from any later key of the chain. Following this scheme, in a given authentication frame j, the key used for the MACs ($K_{j,MAC}$) can be derived from the key received from satellite i as follows:

$$K_{j,MAC} = F^{i-1}\left(K_{j,i},\ GST_j\right) \tag{4}$$

where F^{i-1} is the F function in Equation (3) applied $i-1$ times to $K_{j,i}$, the key received from satellite i in authentication frame j. In this case, the key $K_{j,1}$ would be transmitted by satellite 1 at authentication frame j and also used for the MAC computation, i.e., $K_{j,1} = K_{j,MAC}$. To avoid the reuse of the key, another chain element can be added, or an additional function can be used to generate the MAC keys from the transmitted keys as proposed in [5] and [15]. For the sake of simplicity, this is not part of the described solution, although it may be added to future schemes. In any case, the conclusions derived in this paper are not affected by this potential addition.

To validate the MAC key in authentication frame j with the MAC key from the previous frame $j-1$:

$$K_{j-1,MAC} = F^{S}\left(K_{j,MAC},\ GST_{j-1}\right) \tag{5}$$

where S is a constant representing the maximum number of satellites, i.e., the maximum number of times the function F is executed in one authentication frame. The main drawback of this approach compared to the use of one key per authentication frame is that it requires higher computational power, as the number of one-way operations per chain will be higher. For example, if $S=40$, the chain would become 40 times longer. Assessment of the additional CPU needs for this approach is covered in a later section.

Figure 1 and Figure 2 show how the keys of a certain chain are transmitted and used every time the receiver performs an authentication. The discontinuous lines of Figure 2 for the keys imply that, as explained above, the receiver (Rx) only needs to receive a key from any satellite to authenticate all MACs. As an example, if TBA is 10 seconds, every 10 seconds each satellite would send at least one MAC authenticating the satellite data with a key that, for the 10-second period of authentication frame j, is $K_{j,1}$. This key would also be transmitted from SVID (Space Vehicle ID) 1, while SVID 2 would transmit the previous key in the chain $K_{j,2}$, SVID 3 would transmit $K_{j,3}$, and so on. In this way, a spoofer would be unable to predict the key of one satellite from that of another. One could argue that if SVID 2 is closer than SVID 1 to the receiver, an attacker could predict $K_{j,1}$ from $K_{j,2}$. However, as the transmission of the keys is done in parallel, only the very few final bits of $K_{j,1}$ could be predicted by having a high number of bits of $K_{j,2}$, rendering such an attack rather impractical. User implementations must discard the last bits of the key in the anti-replay statistics.

TESLA and Cross-Authentication

In order to be available in this decade and with an affordable impact on the system infrastructure, the Galileo NMA must be generated on-ground and transmitted in real time to the satellites for broadcast. This ground-to-satellite continuous connection is also used by other satellite systems, such as SBAS, and other geostationary satellite-based services. It was predicted for the Galileo safety-of-life service, it is partly required for the regular Galileo navigation data update, and it is also required for the provision of high accuracy (PPP) through the future Galileo Commercial Service [9]. However, generating NMA on-ground means that only satellites connected to ground can deliver NMA in the Galileo first generation. This means no more than 20 out of 24 Galileo satellites. To overcome this limitation, the navigation data of the satellites not connected to ground at a given time can be authenticated by those connected to ground and transmitting authentication data: Several MACs can be associated with each key, allowing *cross-authentication* of some satellites by others. This feature opens the door to *cross-authenticating* data from other GNSS constellations.

Figure 3 shows how satellites 2 and 4 can authenticate their own navigation message (N2, N4) and cross-authenticate the surrounding (non-connected)

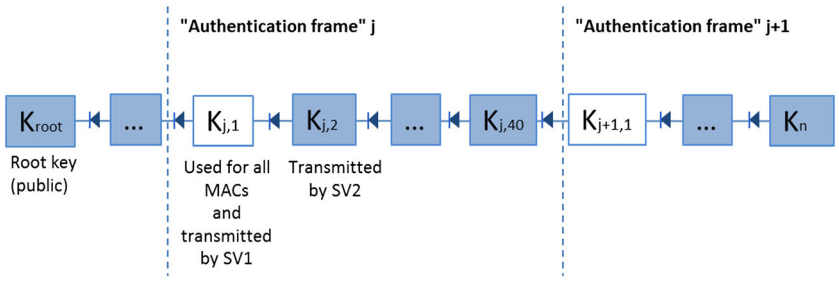

Fig. 1–TESLA one-way chain with different keys from different senders. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

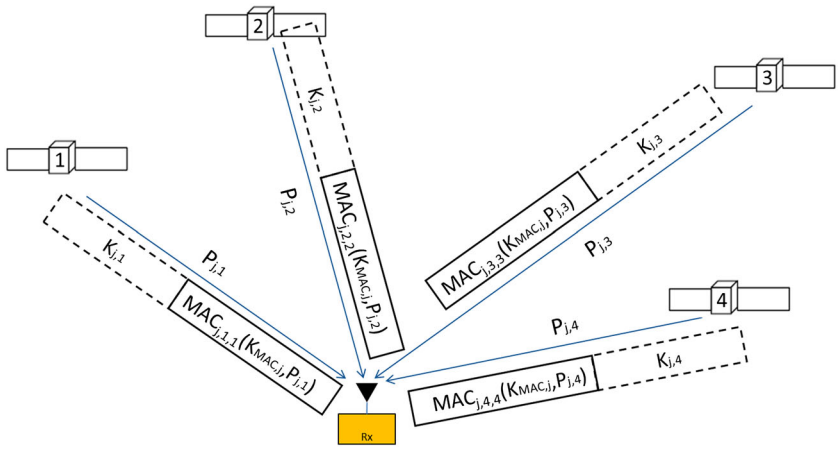

Fig. 2–TESLA single-chain approach with a different key transmitted by each sender. No cross-authentication. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

satellites 1, 3 and 5. This scheme may lead to duplications in the transmitted information (e.g., *MAC(N3,K_j)* in the figure) or to the reception of authentication for satellites that are out of view. However, it ensures the authentication of all the navigation information received by all users on Earth. Note, however, that if N1, N3, and N5 are predictable, which is generally the case, satellites 1, 3, and 5 would not be protected against replay attacks.

Taking into account the *cross-authentication* feature, generation of the MACs is performed as follows:

$$MAC_{j,i,l} = trunc\left(n,\ mac\left(K_{j,MAC},(i \,||\, l \,||\, CTR \,||\, P_{j,l})\right)\right) \tag{6}$$

where $MAC_{j,i,l}$ is the tag (or truncated MAC) transmitted at frame j by satellite i to authenticate the navigation of satellite l (note that i and l coincide when the satellite self-authenticates, as shown in Figure 2), *CTR* is a counter with the position of the tag in the transmission, and $P_{j,l}$ is the navigation data of satellite l at frame j that is authenticated. Note that, by using the *cross-authentication* approach, a satellite can transmit several MACs, for itself and the neighboring satellites. Examples of typical MAC functions used are HMAC-SHA-256, as standardized in [19], and CMAC-AES, standardized as Algorithm 5 in [20]. Note that the signal time is authenticated by ensuring the key authenticity with respect to the K-root: If a key of a certain subframe is authenticated using the TOW (Time Of Week) of that subframe, the TOW must be authentic too as otherwise the one-way process based on Equation (2) would not lead to the valid K-root. Notice also that, if l were not authenticated,

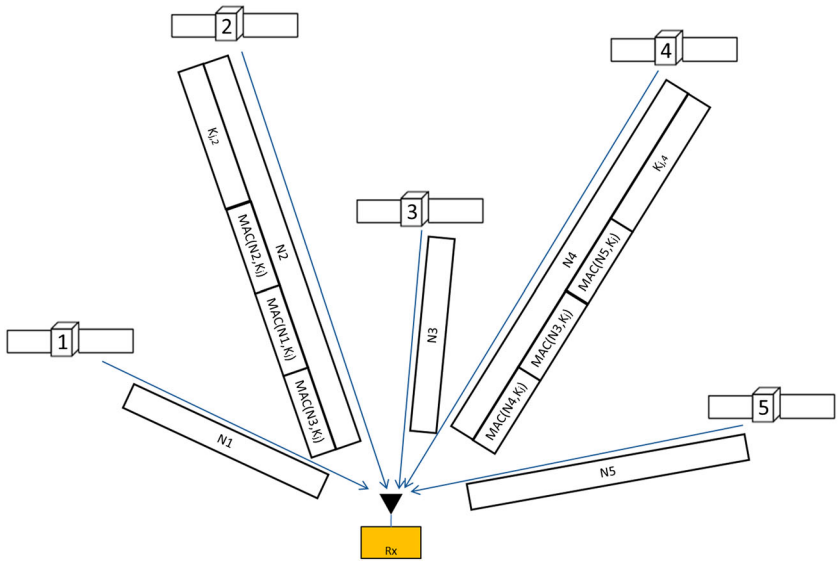

Fig. 3–TESLA single-chain approach with connected (2,4) and non-connected (1, 3, 5) satellites. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

given that all MACs in a MAC-K section are signed with the same key, an attacker could forge the signal and transmit the same navigation data from several satellites and later replay the first MAC.

IMPLEMENTATION TRADE-OFFS AND ANALYSES

This section presents some tradeoffs intended to justify the design decisions leading to the implementation presented throughout the rest of the article.

Digital Signatures versus Time-Delayed Asymmetry

To maximize the use of GNSS authentication, cryptographic key management should be simplified as much as possible. This means that asymmetric schemes, where the user receivers only need to possess a public key, are preferred to symmetric schemes, where the user needs to store a secret key in a security module within the receiver. Asymmetric schemes can be achieved mainly through two options:

- Digital signatures, such as RSA, DSA, or ECDSA [21]. In these schemes, the satellites transmit a digital signature of their navigation data, as described in [2].
- Delayed symmetric key delivery, such as TESLA.

The main advantage of authentication through digital signatures is that there are known methods and functions in the cryptographic standards that make them reliable for the cryptographic community [21]. The main disadvantage of authentication through digital signatures, compared to time-delayed asymmetric approaches, is the bandwidth required to transmit the authentication information. For example, in order to transmit a digital signature that guarantees a 112-bit to 128-bit security level, signatures on the order of 500 bits are required (e.g., 512 bits for standard DSA, 128-bit security, or 466 bits for 112-bit security through the Elliptic Curve Digital Signature Algorithm (ECDSA), as per [2]). Another disadvantage is the computational effort required for each authentication. The main advantage of time-delayed asymmetric schemes is the bandwidth reduction and the tolerance to data loss, as mentioned above. Their main disadvantages may be that they are not as standardized and accepted by the cryptographic community as digital signatures and they are potentially vulnerable to more threats, as they rely on a coarse time synchronization of the receiver with a time reference. In any case, to authenticate the SIS, the receiver must possess some information certified as correct independently from the SIS. This means that even for TESLA-based approaches, the receiver must possess a public key to authenticate the SIS. However, the frequency with which this public key is used, and the bandwidth associated with this process, can be very low, as in the proposed implementation.

Preliminary Assessment of the CPU Needs of a Single One-Way Chain

To understand the computational power required in a single one-way chain where each satellite transmits a different key, state-of-the-art SHA-2 implementations have been studied. It is claimed that around 11.5 processor cycles per byte are required [22]. As a rough estimation, a 1-GHz processor would need around 0.4 microseconds for a SHA-256 (i.e., 32 bytes) iteration. For the following assessment, and taking into account that the one-way function may also involve the concatenation of a time tag, and a truncation, 1 microsecond per iteration is considered. The CPU time required for a single-chain multiple-key TESLA approach, assuming 40 iterations per authentication (allowing coverage of 30+ satellites), would therefore be 40 microseconds for the whole set of satellites every authentication frame, as the operation is required only once for all satellites, which in absolute terms is very affordable for a standard low-end processor. Regarding the CPU needed to verify a certain key against an authenticated root key, which is, for example, 1 week older, and assuming 40 hash iterations for a 10-second authentication frame, 2,419,200 iterations would be required, i.e., around 2.5 seconds, which is highly affordable taking into account that this operation is very infrequent. Therefore, the CPU computing power required seems not to be a major driver. Note that, in a standard 1-chain-per-sender TESLA approach, the chain verification is required for each satellite, leading to lower but still comparable computing power needs.

Security Considerations of a Single One-Way Chain

By using only one chain for all satellites, if the chain is compromised (i.e., the seed key K_n is found), the whole system is compromised. However, as the one-way chain security depends on the choice of the hash primitive and hash bit output length, a cryptographically secure design choice for several chains is as secure for a one-way all-satellite chain. Instead of protecting several seed keys per chain, the system protects only one, but the existence of single versus multiple parallel chains does not change the system architecture, as the security measures of the system are similar.

To increase security, the hash function and the key length can be chosen with high margins. If, due to the higher criticality of compromising a single K_n,

higher security measures are required, the validity period of each chain could be shortened, or the key bit length increased, or other system-related measures could be introduced, while maintaining the advantages of the current concept. As shown later, the proposed implementation permits changes to the key and MAC lengths as well as the cryptographic functions in order to cope with future threats over the lifetime of the service.

Security Considerations on MAC Truncation and One-Way Function Truncation

For applications with large bandwidths compared to the output sizes of cryptographic algorithms, it is common to use MAC functions with output sizes of 80 or even 160 bits, as is the case for some HMAC functions [23]. However, meaningful levels of security are achieved with much shorter output sizes. Given the low throughput available in GNSS signals, the lengths of the key and the truncated MACs are very sensitive parameters. They influence the number of authentication bits (NA) and therefore affect AER and TBA (the latter assuming a fixed bandwidth), which are reflected in all other indicators. Therefore, their length should be reduced as much as possible while maintaining security at an acceptable level. As GNSS are one-way systems, an attacker has no control over the message that is authenticated (i.e., it cannot request a satellite to authenticate a given message) or the key used, and the MAC and key transmission occurs with a certain cadence controlled by the system specification. This yields some attacks impractical, permitting the use of very short MACs.

Assuming the MAC algorithm behaves as a lookup table with a message and a key as entries, and an n-bit random sequence as output, an attacker could only try to guess the MAC, which is very unlikely even for extremely truncated MACs to only a very few bits [10, 17]. For example, a MAC as short as 10 bits would be guessed with an average probability of 0.097% (one time out of 1024), rendering the 'MAC guessing' \ compared to a pure service denial by, e.g., jamming the signal. Some additional protections against this attack are as follows:

- A receiver can accumulate two or more MACs before accepting the data as authenticated. Given the very short TBAs and the possibility to cross-authenticate several satellites, this permits data authentication with higher probabilities with a delay of a few seconds, or even without any delay. For example, one can imagine a receiver that only uses a new Issue Of Data (IOD) when it is authenticated twice, reducing the 'MAC guessing' probability to $1/(1024)^2$ and navigating in the meantime with the previous IOD, which should minimally affect the navigation performance.
- Each MAC-Info section could be encrypted with the key delivered later, making the 'MAC guessing' attack more difficult and also adding unpredictability to the signals.

Security Considerations on Key Length

Regarding the symmetric key length used in the one-way chain, in accordance with [24] and [25], we consider a key length of 80 bits to be strong enough for chain durations of up to 1 year (at the time of writing). The inclusion of information that changes with every iteration to the input of the hash function, cf. Equation (3), works as a 'salt' and counters multiple-target attacks on the hash function, thereby significantly increasing the strength of the mechanism. If the key has to be guaranteed for, e.g., 20 or 30 years, 128-bit keys would be recommended [26]. While [26] proposes to restrict the use of 80-bit keys to legacy applications, we point out that [27] allows the continued use of 80-bit keys when each key encrypts less than 2^{20} blocks of plaintext, which is the case here. Furthermore, the use of 80-bit keys is in accordance with the still widespread use of 160-bit hash functions like SHA-1 and RIPEMD-160, and the current implementation allows longer keys in future chains (up to 128 bits).

To accommodate keys as short as 80 bits in a standard one-way function such as SHA-256 [12] or SHA-3 [28], the one-way function needs to truncate the output of the hash function to the length of the key for every iteration in the chain, as shown in Equation (3).

Root Key Authentication and Public Key Management

In addition to the SIS information, the system shall provide a public key through other means than the SIS, allowing the verification of the root key by a digital signature. The authenticity of the public key must be ensured, e.g., by allowing access to the receivers to a trusted public source containing the public key(s) or the fingerprints of the public keys. An external certification authority may also be used.

A public-private key pair may be valid for several years, but possibly not for long enough to cover the entire lifetime of the system. Moreover, public–private keys may need to be revoked. Also, the public keys cannot be published much before their validity period, as otherwise the paired private key could be attacked by the knowledge of the public key, even if no information has been yet signed to it. Therefore, the system must be able to provide new public keys to the users. To provide a new public key, different approaches may be followed. Either the system does over-the-air rekeying for the public key [17] or the receiver connects to a network to obtain it. Otherwise, publishing keys with a certain frequency and validity period of, e.g., 1 year and 5 years, respectively, and sending digital

signatures using all of the keys would allow receivers not to have to download the keys for some years. All approaches have pros and cons regarding autonomy and management of key revocation, but they prove that public-private key management schemes grant a high autonomy to NMA receivers.

T_0 (GST_0 sync.) (s)	E1-B Content						E1-B Page	E1B Sub frame ID
0	Spare Word (2/2)	Res	SAR	Spare	CRC	Res	Odd	N-1
1	Word 2 (1/2)						Even	N
2	Word 2 (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
3	Word 4 (1/2)						Even	N
4	Word 4 (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
5	Word 6 (1/2)						Even	N
6	Word 6 (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
7	Word 7 or 9 (1/2)*						Even	N
8	Word 7 or 9 (2/2)*	Res	SAR	Spare	CRC	Res	Odd	N
9	Word 8 or 10 (1/2)*						Even	N
10	Word 8 or 10 (2/2)*	Res	SAR	Spare	CRC	Res	Odd	N
11	Reserved (1/2)						Even	N
12	Reserved	2/2)	SAR	Spare	CRC	Res	Odd	N
13	Reserved (1/2)						Even	N
14	Reserved	2/2)	SAR	Spare	CRC	Res	Odd	N
15	Reserved (1/2)						Even	N
16	Reserved	2/2)	SAR	Spare	CRC	Res	Odd	N
17	Reserved (1/2)						Even	N
18	Reserved	2/2)	SAR	Spare	CRC	Res	Odd	N
19	Reserved (1/2)						Even	N
20	Reserved	2/2)	SAR	Spare	CRC	Res	Odd	N
21	Word 1 (1/2)						Even	N
22	Word 1 (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
23	Word 3 (1/2)						Even	N
24	Word 3 (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
25	Word 5 (1/2)						Even	N
26	Word 5 (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
27	Spare Word (1/2)						Even	N
28	Spare Word (2/2)	Res	SAR	Spare	CRC	Res	Odd	N
29	Spare Word (1/2)						Even	N
30	Spare Word (2/2)	Res	SAR	Spare	CRC	Res	Odd	N

Fig. 4–Galileo E1B I/NAV message structure. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

IMPLEMENTATION EXAMPLE: NMA IN GALILEO E1-B I/NAV

This section presents an implementation of the proposed NMA concept in the Galileo I/NAV message structure, which is based on the transmission of a full navigation frame every 750 seconds. The I/NAV frame is composed of 30-second subframes, which always transmit all the required positioning parameters except the almanac. Every subframe is divided into 15 2-second pages, each of which contains one word and some other fields, as shown in Figure 4. A total of 125 bps can be transmitted in the E1-B I/NAV message. These bits are convolutionally encoded at 250 sps and interleaved. The E1-B symbol stream is modulated through a Direct Sequence Spread Spectrum (DSSS) technique with 4092-chip pseudorandom codes at 1.023 Mcps, and with a subcarrier at 1.023 MHz to achieve a BOC(1,1) modulation. Thus, every 4-ms symbol is modulated within a single 4-ms code. Details about the Galileo I/NAV signals and message are in the Galileo OS SIS ICD [8].

We propose to use the field 'Reserved 1' in the ICD [8] to transmit NMA information. This field provides a bandwidth of 40 bits every other second. The reason to use this field, as opposed to other spare fields in the I/NAV message, is that it can be filled in and transmitted to the satellites with minimal impact on the core mission navigation and control tasks, as explained before and presented in more detail in [18]. Using this field also scatters the NMA bits across the navigation message, reducing MPT.

Figure 4 presents the Galileo E1-B I/NAV message structure and highlights the position of the 'Reserved 1' field. Note that only the same amount of symbols as unpredictable bits before encoding are considered unpredictable (i.e., a maximum of 40 symbols in the current case [29]). This must be taken into consideration when assessing the MPT and USR parameters.

The 40 bits-per-2-second bandwidth yields 20 bps for a total of 600 bits every I/NAV subframe, after which, in nominal conditions, the I/NAV words are repeated. This 30-second subframe structure has also been taken as a reference for NMA, to facilitate synchronization between the reference time, the authenticated navigation data, and the authentication data.

While a thorough description of each header and field that compose the NMA transmission structure is outside of the scope of this paper, the main data blocks of the proposed implementation are presented in Figure 5 and explained later.

The top row of Figure 5 shows the subframe time, which goes from 0 to 30 seconds. The second row shows the page order, from 1 to 15. For every page, 40 bits are available for NMA. The SIS authentication information is based on two main sections transmitted in parallel:

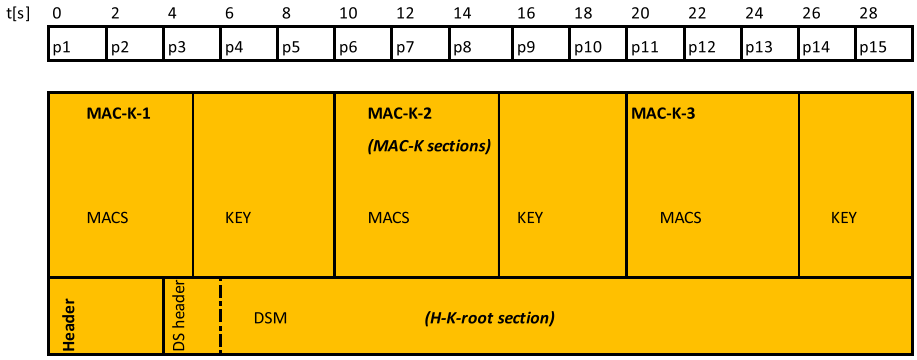

Fig. 5–Galileo NMA proposal within the I/NAV message structure. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

- 'H-K-root' section, with the global header and a digitally signed root key.
- 'MAC-K' sections, with the MACs and associated delayed key. Three 10-second MAC-K sections are shown.

The authentication service will mainly be based on the MAC-K section, which occupies 32 out of the 40 bits per word, leaving 8 bits per word for the H-K-root section. This implies a total of 120 bits per subframe for the H-K-root section and 480 bits per subframe for the MAC-K sections. The MAC-K authentication implements a TESLA authentication scheme, as described above. To authenticate the keys used in the MAC-K section, a root key (K-root) will be continuously digitally signed and sent in parallel in the H-K-root section. Reading the root key is required only when the user needs a new root key, which should happen very infrequently. Separating H-K-root and MAC-K sections maintains a constant level of unpredictability and allows more flexibility in the solution design.

H-K-root Section

H-K-root will be transmitted synchronously with the I/NAV subframe. This means that in each 30-second subframe, a full H-K-root block of 120 bits can be transmitted, as depicted in Figure 5. The H-K-root section is composed of a global Header and the K-root digital signature and message (DSM) information. The DSM contains the root key (K-root), plus some descriptive fields that include the associated time and the digital signature information.

The type of digital signatures used is still under consideration and is not the main scope of this paper. Schemes such as Elliptic-Curve Nyberg-Rueppel, for message recovery, or Schnorr-DSA, as described in [30] and [31], respectively, are plausible options. To add flexibility to the design, a DSM will authenticate not only the K-root and its reference time but also other specific parameters of the chain, such as the key size (which can be increased if needed), the MAC size, the one-way function (e.g., SHA-256 or SHA-3), and the MAC function (e.g., HMAC-SHA-256) etc. In this way, the NMA solution gains in flexibility while maintaining its format and specification, which is a highly desirable parameter for a service that may be in operation for several decades.

MAC-K Section

The section transmitted in parallel to the H-K-root contains the truncated MACs and keys used for authentication. With a key length of 80–128 bits and a truncated MAC length of 10–20 bits, two or three keys can be sent every subframe, one every 15 or 10 seconds, respectively, with their associated MACs. The MAC-K section is composed of the following:

- A **MAC section**, which in turn is composed of the **MAC** and a **MAC-Info** section, giving information about the MAC.
- The **key**, which will follow the one-chain multiple-key scheme mentioned above.

Figure 6 presents the structure of the MAC-K sections. The different fields are grouped by type: all MAC and MAC-Info sections of a given MAC-K section are put in a column, the associated key is put in the next column, and so on. The order in which the information is received by a receiver is given by reading each row from left to right, and then from top to bottom.

MACs are generated according to Equation (6). The MAC-Info section is transmitted contiguously with the MAC and is composed of the following:

- A Satellite Vehicle ID field (SVID), to allow for cross-authentication of surrounding satellites, either from the same constellation or from others. An 8-bit field as per Figure 6 allows for up to 255 satellite IDs.
- An 'Authentication Data & Key Delay' field (ADKD), which can fit into 4 bits and is defined as shown in Figure 7.
- A truncated issue of data field (IOD), which identifies the data to be signed. It may relate to an IOD from the navigation message or another convention may be used, depending on the ADKD case.

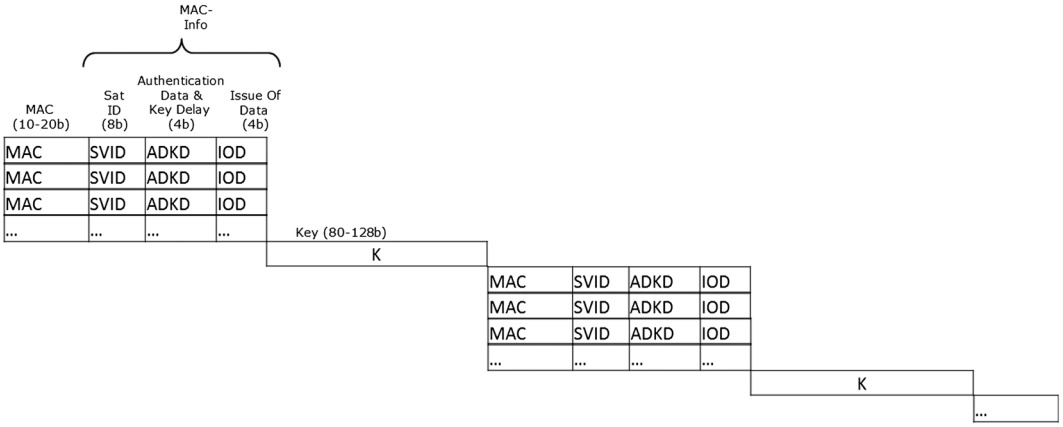

Fig. 6–MAC-K section structure

Value	0	1	2	3	4	5	6	7
Definition	Eph & Clk	Iono+	Subframe	Gal F/NAV eph & clk.	GPS L1C eph & clk	SBAS-related TBC	Rsvd	Rsvd
	8	9	10	11	12	13	14	15
	Rsvd	Rsvd	Rsvd	SLOW-MAC Eph&clk 1 subframe delay	SLOW-MAC Eph&clk 2 subframe delay	SLOW-MAC Eph&clk 3 subframes delay	SLOW-MAC Eph&clk 4 subframes delay	SLOW-MAC Eph&clk 5 subframes delay

Fig. 7–Authentication data and key delay

While more ADKD values can be further refined, this definition is representative of the fact that each transmitted MAC could sign different types of information not only from different satellites but also from different signals from the same satellite, and for different sets of information. For example, in order to reduce AER by authenticating fewer bits, most of the transmitted MACs can sign only the ephemeris and clock data (ADKD = 0), and only a few MACs can sign the ionospheric information (ADKD = 1). The meaning of the most relevant values of the ADKD field to understand and characterize the concept is described here:

- '0', *Eph & Clk*: the MAC authenticates the ephemeris and clock data bits transmitted in the E1 I/NAV message, Words 1 to 4 as per [8].
- '1', *Iono+*: the MAC authenticates data bits in the I/NAV Word 5: ionospheric correction, BGDs, etc.
- '2', *Subframe*: the MAC authenticates the bits of a full subframe, including all data words. It does not include other reserved fields, the SAR data or the CRC.
- '11' to '15', 'slow-MACs': To relax the TESLA loose time synchronization requirement in the receiver, the concept of 'slow MACs' is added. A 'slow MAC' is a MAC generated with a key that will be broadcast some subframes later. For example, if the ADKD field value is 15, it means that the receiver will get the key associated to that MAC exactly with a five-subframe delay with respect to the time it would have received it in normal conditions. If a receiver is switched on and its clock error can be up to several seconds before GNSS signal acquisition and synchronization, an attacker could spoof the data. The spoofing attack would consist of receiving and rebroadcasting a valid key with a navigation message including a system time reference coherently delayed with the rebroadcast key, wrong navigation, and correct MACs computed with the already known key. Following the example, a receiver with a 5-ppm clock stability could accumulate a 10-second error after 23 days without GNSS, which is a plausible scenario. Transmitting 'slow MACs' allows detection of such attacks: with a five-subframe delay, the same clock would take around 370 days to accumulate such an offset.

One advantage of associating an IOD with a set of authentication data is that it ensures that authentication latency has very little impact on NMA performance. This is a relevant insight for the design of an NMA system, understanding latency as the time between navigation data being received on ground and the time the authentication is received on ground. The main reason for this is that a user who is already tracking data-authenticated signals and calculating authenticated PVT can continue

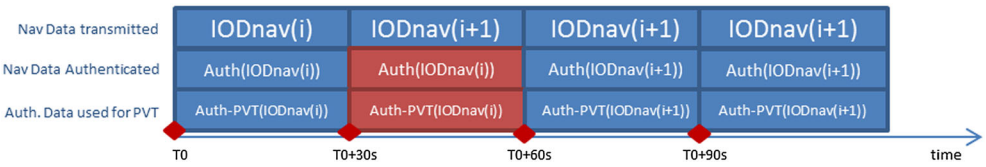

Fig. 8–Data-authenticated PVT as a function of IODnav of a given satellite during IODnav transition. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

navigating using authenticated data. Thus, when a new IODnav is transmitted by the system but not yet authenticated by the NMA generation module, the user receivers can still use the previous IOD until the next subframe in which the new IOD is authenticated. This will cause no or minimal degradation in the navigation performance during 30 seconds, as illustrated in Figure 8.

An authentication latency of 30 seconds would only impact the case in which a user receives a subframe containing *IODnav(i + 1)* and *Auth (IODnav(i))*, such as between T0 + 30s and T0 + 60s in Figure 8. For TTFAF, we can assume that the satellites whose IODs are being updated at a given time are a minority and that the user will still have enough satellites in view whose IOD is not updated, therefore causing no degradation. However, in a worst-case scenario, in which the updated IODnav satellites are necessary for the PVT and therefore for the first fix, the user might need to wait for another subframe (30 seconds) to receive the new authenticated IODnav. To overcome this, the NMA system module could receive a new IODnav from E1 and E5b (which both transmit the same I/NAV message, but in different order) and, assuming a latency of 12 seconds, would still have time to transmit it to the user within the subframe.

Bandwidth Allocation Analysis and Comparison with Other NMA Proposals

This section first presents a bandwidth allocation analysis between MAC lengths and key lengths. Based on this analysis, we select the preferred options for the implementation of the authentication solution to avoid having unused bits in the MAC-K sections. In the current example, the following input parameters have been considered:

- MAC-K sections total bandwidth: 480 bits.
- Number of MAC-K sections per subframe: 3.
- Total length per MAC-K section: 160 bits.
- MAC header length (including IOD, SVID, ADKD): 16 bits.

With these constraints, Table 1 shows the combinations that yield all 160 bits used, and the achievable number of MACs every 10 seconds. We can observe that by keeping a truncated MAC length of 10 bits, and a key length of 82 bits or less, 3 MACs per MAC-K section can be transmitted, for a total of 9 MACs, i.e., 9 data-authenticated satellites per channel, every 30 seconds.

Table 1—Number of Possible MACs for a Given MAC Length and Key Length Combination Using the Whole Bandwidth

Key size [bits]	MAC size [bits]	Number of MACs
82	10	3
88	20	2
90	19	2
92	18	2
94	17	2
96	16	2
98	15	2
100	14	2
102	13	2
104	12	2
106	11	2
108	10	2

The full use of the 20 bps of the I/NAV E1-B 'Reserved 1' field allows very low TBAs and a high cross-authentication redundancy, both of which increase robustness and performance. However, in absolute terms, 20 bps could seem high compared to other NMA solutions, so a more detailed comparison has been performed. AER, at least under an AWGN channel, is not affected by bitrate. TBA is affected by bitrate as presented in Figure 9. Figure 9 assumes that 200 bits between authentications are required, which corresponds to 160 bits for the MAC-K section, and 40 bits for the H-K-root section transmitted in parallel.

An assessment of the bandwidth allocated in other NMA solutions proposed for the GPS constellation [2, 17] has been performed and is summarized here. These solutions define NMA authentication methods on top of the GPS CNAV messages. CNAV does not specify a fully static message structure but defines the minimum and maximum rates for each message type and also allows us to define new message types that can be sent in spare blocks. The rates required by the system are particularly demanding for the ephemeris and clock data and more relaxed for other information. Each of these messages may contain several unallocated bits. These bits together with the possibility of defining new messages to complete the spare blocks of the CNAV allow definition of many different message sequences for future services. In summary, these GPS NMA

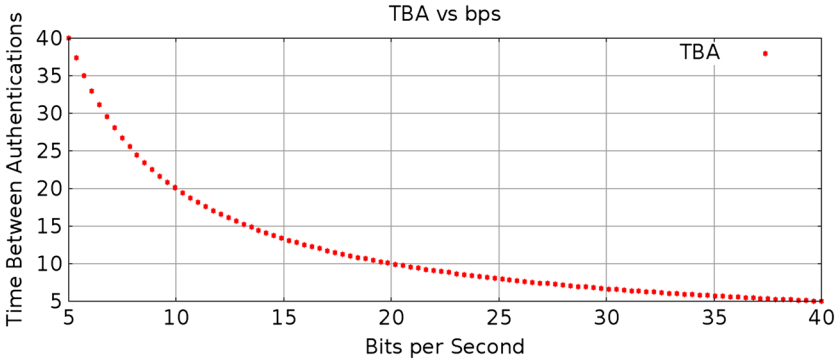

Fig. 9–Time between authentications [s] versus bandwidth [bps], assuming 200 bits required between authentications. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

solutions define a bit allocation range between 7% and 25% of the total CNAV bandwidth, for TBAs of more than 5 min down to around 100 seconds in L2-CNAV. The difference in TBA of one order of magnitude or more compared to the proposed solution is mainly due to the different bitrate (125 bps in Galileo I/NAV versus 25 bps in GPS L2-CNAV) and the reduction of authentication bits in the proposed solution. This indicates that the solution herein proposed for Galileo E1B, which occupies 16.6% of bandwidth, is in the same range as other NMA solutions in the literature, in relative terms. Taking into account that the Galileo I/NAV message structure still has 7 out of 15 spare words (for a total of 840 bits per subframe) and other spare fields, and that increasing the NMA bandwidth increases TBA and therefore TTFAF and robustness, the full use of the 20 bps of the 'Reserved 1' field seems a reasonable design choice.

Time Staggering of Authentication Events

The introduction of a time offset for subsets of satellites to stagger the authentication verification times can improve authentication robustness, as proposed in [2]. In a fully synchronous scheme with a TBA of 10 seconds, all satellite information is authenticated every 10 seconds. However, receivers subject to an attack during these 10 seconds will only detect it when the authentication verification is performed. If the transmission of authentication information is offset for some satellites, different subsets of satellites would be authenticated at different times, leading to a lower all-in-view TBA. The Key-to-MAC allocation proposed in this scheme must allow minimization of TBA at user level while not compromising other parameters, and avoiding the disclosure of the keys before or during the period when the associated MACs are transmitted. Even if less than the four satellites required to compute a position are authenticated at every authentication event, a receiver can make use of the pseudorange bounds for increasing its robustness, in combination with other previously authenticated ranges. The main features of the proposed offsetting example are as follows:

- MAC-K transmission is divided in two groups:
  - SVID 1 to SVID 15 transmit MAC-K sections allowing authentication at seconds 0, 10, and 20 of the subframe.
  - SVID 16 to SVID 30 transmit delayed MAC-K sections allowing authentication at seconds 4, 14, and 24.
- MACs transmitted by SVID 1 to SVID 15 use the key transmitted from SVID 1 ($K_{j,1}$). This key can also be recovered from a key from any satellite SVID 2 to SVID 15 transmitted at the same time, or any key transmitted later.
- MACs transmitted by SVID 16 to SVID 30 use the key transmitted from SVID 16 ($K_{j,16}$). This key can also be recovered from a key from any satellite SVID 16 to SVID 30 transmitted at the same time, or any key transmitted later.

Figure 10 presents this offsetting in MAC and key transmission scheme.

The proposed scheme authenticates half of the satellites at 0, 10, and 20 seconds, and the other half at 4, 14, and 24 seconds, leading to an *all-in-view* TBA between 4 and 6 seconds, with an average of 5 seconds. As shown in the figure, the proposed scheme allows the simultaneous transmission of keys and MACs without compromising the system, as the transmitted MACs always use a key that cannot be computed from the keys under transmission.

PERFORMANCE COMPARISON

This section characterizes the proposed solution in terms of AER, MPT, and USR, and assesses the impact of adding NMA on accuracy, availability, and time to fix. Summarizing, the characterized implementation is based on the following parameters:

- One MAC-K section every 10 seconds, including one key and three MAC-K sections.

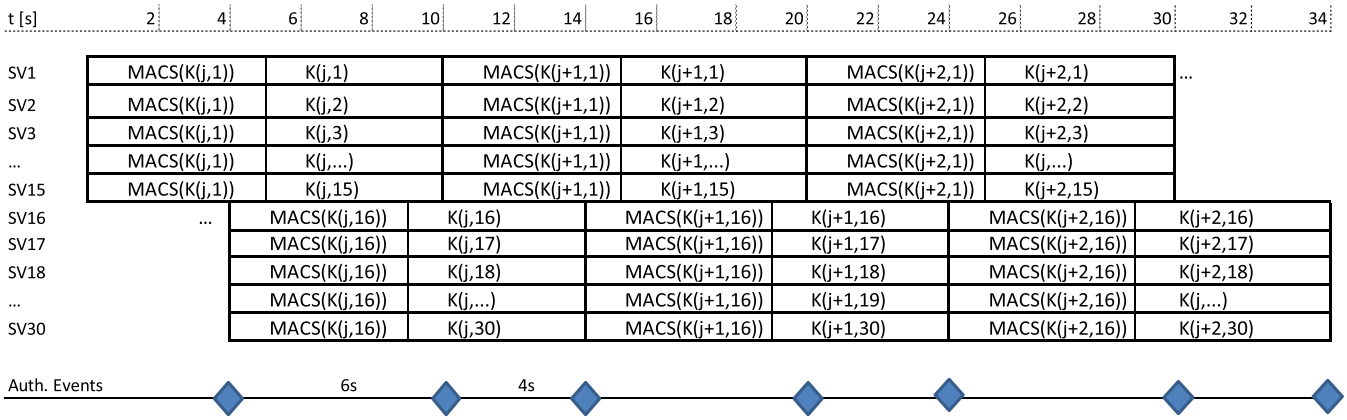

Fig. 10–Key-to-MAC allocation and authentication offsetting. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

- Key length of 82 bits.
- MAC length of 10 bits.
- MAC-Info length of 16 bits.

The characterization below excludes the processing of the H-K-root. We assumed that the receiver has an authentic K-root.

Authentication Error Rate: As shown in Equation (1), AER depends on the bit error rate (BER) and on the number of navigation and authentication bits (NNA) to be demodulated. BER, in turn, can be bounded by Equation (7) [32], assuming a static receiver, an AWGN channel and stable PLL tracking:

$$BER \leq \frac{1}{2} \cdot \left(36D^{10} + 21D^{12} + 1404D^{14} + 11633D^{16}\right) \quad (7)$$

where

$$D = e^{-\frac{C/N_0}{2Rb}} \quad (8)$$

and Rb is the number of bits per second. The BER in (7), from which the AER is calculated, represents the average bit error rate at a given C/N_0, after soft decision Viterbi decoding of the Galileo I/NAV symbols [8]. This is the same expression used for the modernized GPS signals and SBAS signals using the same convolutional code [32]. AER versus C/N_0 is shown in Figure 11 for three cases, where ADKD are '0', '1,' and '2.'

Figure 11 shows that, under these assumptions, very low AER values are obtained even at low C/N_0 values. For example, an AER of 1% is obtained with a C/N_0 between 25 dBHz and 26 dBHz for all cases. It also shows that below 24 dBHz NMA is barely usable. The authentication performance is thus as good as expected and in line with the I/NAV demodulation performance. A receiver able to successfully demodulate the navigation data should also be able to authenticate this data. A receiver in an environment subject to fading may have problems performing authentication. However, it still can have a high NMA availability by receiving

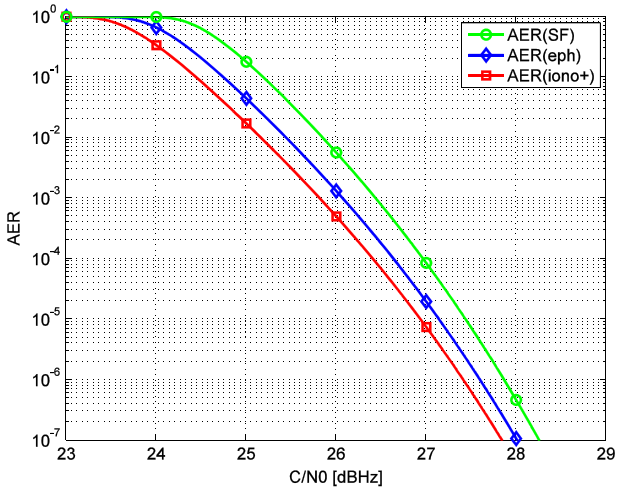

Fig. 11–AER versus C/N_0 for I/NAV authentication, ADKD = 2 (SF), ADKD = 0 (eph), ADKD = 1 (iono+). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

the keys and MACs from the connected satellites with best visibility conditions.

Figure 12 shows the different error rates for decoding just the navigation data and for decoding the navigation plus authentication, assuming an AWGN channel. FER-1SV represents the frame error rate (FER) of decoding only the ephemeris and clock of one satellite (words 1–4, NN = 506 bits). AER-1SV represents the authentication error rate of decoding the same bits plus the authentication (NNA = 608 bits in total). Similar reasoning applies to the 4SV and 4SVI cases. 4SV implies decoding and authenticating the ephemeris and clock data, plus TOW, for four satellites (NN = 2044 bits; NNA = 2230 bits) and 4SVI implies decoding the same data as in the 4SV case plus an additional word (Word 5) with the ionospheric and BGD corrections (NN = 2129 bits; NNA = 2341 bits). Figure 12 shows that there is very little difference in the error rate for the receiver, with or without authentication, at a given C/N_0.

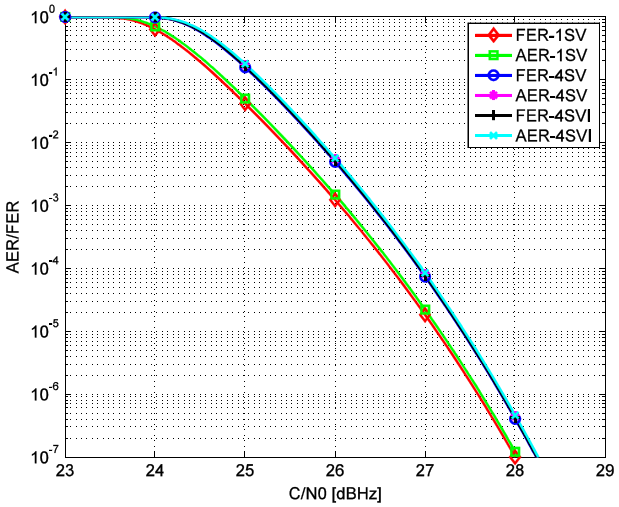

Fig. 12–Error rates with and without authentication for 1 satellite, 4 satellites and 4 satellites and 'iono + '. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

In absolute terms, and assuming that a Galileo signal processed with a receiver with a standard noise figure and in good visibility conditions will be received with a C/N_0 of around 45 dBHz, it can be concluded that, even with a power loss of 18 dB (from 45 dBHz to 27 dBHz), data authentication can be performed at a very low error rate (AER for four satellites is below 10^{-4}). Therefore, taking into account that all the data required for navigation is authenticated, the degradation of adding the proposed NMA implementation to accuracy and availability of the standard navigation performance will be minimal. Also, given that TBA is 10 seconds and therefore significantly lower than the time to get a full subframe (30 seconds), and taking into account the slight degradation in error rates, the difference between TTFF and TTFAF will be negligible. Only when the receiver needs a new K-root at start-up, which is foreseen to occur very seldom, or when the receiver synchronization is so loose that it needs to process a 'slow MAC', will TTFAF be higher.

For comparison purposes, Figure 13 presents the 'four satellite NA-AER' for a given BER. It represents the probability that four satellites are not correctly authenticated under a noisy channel (AWGN in this case), versus a given BER, for different authentication solutions, including the one under study. In order to highlight the differences between the authentication solutions, only the authentication bits (i.e., NA as opposed to NNA) have been considered. The TESLA solutions assume a 16-bit MAC-info section per MAC, allowing cross-authentication. The solutions analyzed are as follows:

1. AER-NA-DS-112: 466-bit digital signature, one per satellite (112 security bits; 1864 bits needed)
2. AER-NA-STD-TESLA-112: Standard TESLA approach, with 112-bit keys and 10-bit truncated MACs (224 security bits; 552 authentication bits needed).
3. AER-NA-1C-TESLA-112: Current concept, with 10-bit MACs and 112-bit keys (112 security bits; 216 authentication bits needed).
4. AER-NA-1C-TESLA-82: Proposed implemen tation, with 10-bit MACs and 82-bit keys (82 security bits; 186 authentication bits needed).

Cases 1, 2, and 3 use the same number of security bits but illustrate the improvement from digital signatures to TESLA, and from multiple-chain TESLA to single-chain TESLA. Case 4 represents the current proposal. One can see that between the standard digital signature approach and the current proposal there is an improvement of around one order of magnitude at low BERs.

Authentication Error Rate Including Multiple Satellites and Multiple MAC and Key Channels

The data authentication performance of the proposed solution also depends on having multiple channels to receive redundant authentication information. This section quantifies AER in this case. In the analyzed scenario, only natural impairments degrade the capability of decoding the authentication (MACs, keys) and navigation information.

Let us calculate the probability of authenticating a satellite over a given time interval T. In order to authenticate a satellite, we need the correct reception of the navigation data (assuming it is not already available), the key, and the MAC. Therefore, the probability of successfully authenticating satellite l can be expressed as

$$\begin{aligned} P_{S_l} &= P_{M_l} P_K P_{N_l} \\ &= \left(1 - \overline{P}_{M_l}\right)\left(1 - \overline{P}_K\right)\left(1 - \overline{P}_{N_l}\right) \end{aligned} \tag{9}$$

where P_{M_l}, P_K, and P_{N_l} are the probabilities of successfully decoding a MAC for satellite l, the key, and satellite l's navigation during T, respectively. Let us start by calculating the probability of *not* successfully decoding a MAC of a given satellite l:

$$\overline{P}_{M_l} = M_{FER_l}^{NMl \cdot C} \tag{10}$$

where M_{FER_l} is the frame error rate of the MAC information packet for satellite l, NMl is the number of MACs that satellite l received per channel during T, and C is the number of channels. The number of channels in this context is the number of connected satellites. In other words, the probability of not getting a MAC for satellite l is the probability of not receiving any of the MACs for this satellite that were transmitted during T from *any* of the C connected satellites (or channels).

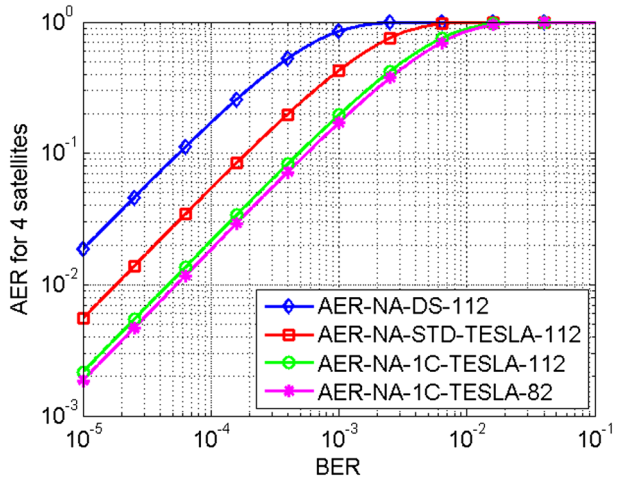

Fig. 13–Four-satellite AER versus BER. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

We can now calculate the probability of *not* successfully decoding *any* key usable to authenticate satellite l's MAC during T as follows:

$$\overline{P}_K = \left[\frac{1}{N_{MK}} \sum_{i=1}^{N_{MK}} (K_{FER})^i\right]^C \tag{11}$$

where N_{MK} is the number of MAC-K sections in T, K_{FER} is the frame error rate of the key packet (i.e., the key, as no further information is transmitted), and C is the number of channels. The formula is explained with an example: Let us suppose that T is 30 seconds (1 subframe), and N_{MK} is three, each MAC-K with one key. If satellite l's MAC was received in the first MAC-K section, then it can be authenticated with any of the three keys that will come during T. The probability of not receiving successfully any valid key is therefore $(K_{FER})^3$. If satellite l's MAC was received in the second MAC-K section, it is $(K_{FER})^2$, and if it were in the first section, it is K_{FER}. As we do not know *a priori* to which section the MAC belongs, all sections are treated with the same probability $1/N_{MK}$, or 0.33. The sum of all cases is what Equation (11) provides.

Finally, we need to account for the error rate in the navigation reception:

$$\overline{P}_{N_l} = N_{FER_l} \tag{12}$$

where N_{FER_l} is the navigation frame error rate for satellite l. We assume that it is transmitted once every T, and only from each satellite. Otherwise, a power factor would be applied as for the case of the MAC and the key.

The plots in Figure 14 present some results for the current implementation proposal in cases where there are two satellites transmitting NMA ($C=2$, left) and when there is one satellite transmitting NMA ($C=1$, right). For simplicity, the same BER is assumed in the calculations, even if the information may come from different sources.

Figure 14 shows that adding authentication in the way proposed does not degrade navigation performance if there are two satellites transmitting authentication information, and the degradation is minimal in the case of a single channel.

Maximum Predictable Time: This parameter depends on how the encoding and interleaving process of the Galileo I/NAV message encodes the unpredictable bits into symbols (some of which will be predictable and some not) and spreads them across the transmitted message. It turns out that all of the unpredictable symbols of every 2-second word are transmitted in a period of 0.4 to 0.5 seconds, leaving the remaining 1.6 to 1.5 seconds fully predictable. As a reference, an MPT of 1.6 seconds will be taken. This is based on the assumption that every 40-bit 'Reserved 1' field contains unpredictable information. Since every 'Reserved 1' field contains 32 bits of a MAC-K section, and the longest predictable interval of a MAC-K section is the 16 bits of the MAC-Info field, all the 'Reserved 1' fields will contain some unpredictable symbols. Test statistics to use the unpredictable symbols to protect against replay attacks are presented in [3, 6, 29].

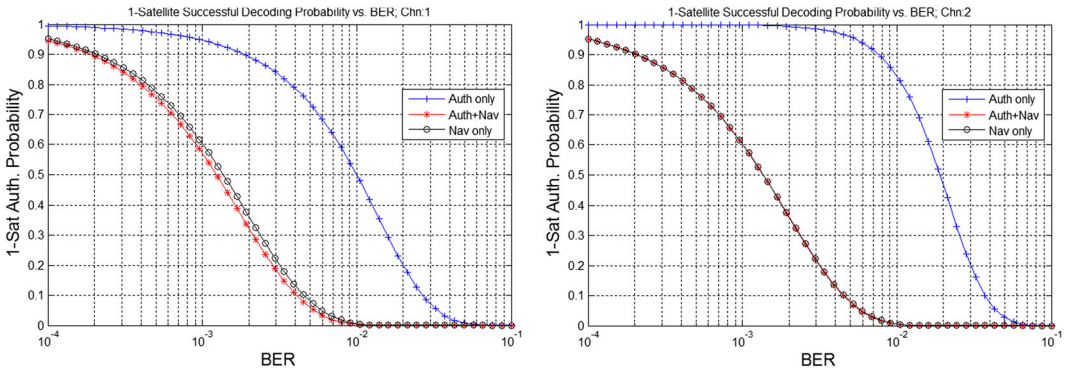

Fig. 14–One satellite ('1-Sat') successful authentication probability when only the MAC & key is required (cross), and when the MAC & key & navigation is required (diamond), vs. the probability of successful reception of navigation information (circle). Left: 2 channels; Right: 1 channel. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

Unpredictable Symbol Ratio: This parameter is calculated under the assumption that all symbols are predictable except the MAC and the key bits, excluding some last bits of the key to avoid attacks whereby the last key bits are deduced and rebroadcast. The number of unpredictable symbols can be determined as follows:

$$USR = \frac{\left(K_{len} - K_{pred} + N_{MAC}\, MAC_{len}\right) N_{MACK}}{N_{SF}} \tag{13}$$

where K_{len} is the key length (82 bits), K_{pred} is the number of key bits considered predictable, N_{MAC} is the number of MACs in a MAC-K section (three in our implementation), MAC_{len} is truncated MAC length (10 bits), N_{MACK} is the number of MAC-K sections in a subframe (three in our implementation), and N_{SF} is the total number of symbols in a given time period in which the message structure is repeated (I/NAV subframe). A K_{pred} of 20 bits has been used. That gives a total of 276 unpredictable symbols per subframe out of a total of 7500 symbols, i.e., a USR of 3.68%. That means that on average, there are 9.2 unpredictable symbols per second from which a receiver can compute an anti-replay test statistic.

FUTURE WORK

The AALECS (Authentic and Accurate Location Experimentation with the Commercial Service) project [33] has prototyped the presented authentication concept and has started testing in the last months. Guidelines to understand detailed NMA implementation aspects in GNSS receivers are also under development. They should include defining the level of protection provided that some satellites in the position solution may be not authenticated at all; only data-authenticated; data and TOA-authenticated, where TOA authentication has an associated confidence level based on a test statistic; and all these, combined with other trust measures such as AGC, J/N detectors, trusted clocks, inertial sensors, or antenna arrays. Another area of future work is the detailed definition of infrastructure and processes for public key management between the user receivers and the Galileo system.

The level of protection of NMA against replay attacks in highly degraded environments is also currently under study. For the future, NMA can be combined with spreading-code authentication, for example, in combination with Galileo E6 encrypted codes (assuming inter-signal biases can be estimated or are not relevant for the target authentication application). The current NMA structure could be strengthened in future Galileo satellite payloads. For example, it could be used in combination with next generation signals *watermarked* at code level, as proposed in [34] with a TESLA key transmitted a few seconds later [11].

CONCLUSIONS

The Galileo program is studying the provision of an open navigation message authentication (NMA) service in the years to come, in order to contribute to the mitigation of GNSS vulnerabilities and provide a differentiator with respect to other GNSS. Different applications could benefit from NMA to protect against certain spoofing attacks, used in isolation or in conjunction with inertial sensors, trusted clocks, or antenna arrays.

This article presents a concrete NMA implementation for the Galileo Open Service. It is based on the standard TESLA protocol modified in order to use a single chain of keys for all satellites, to increase robustness to data loss. The scheme is designed to allow *cross-authentication* of neighboring satellites by a given satellite, improving availability in difficult visibility conditions and overcoming current limitations in the Galileo system.

We propose to divide the NMA structure into two main sections: The H-K-root section to transmit a header and a signed root key needed to initialize the authentication process, and the MAC-K sections to transmit the MACs and keys used regularly for authentication. The concept follows a flexible approach whereby the SIS can inform the receiver

Table 2—Performance Characterization of the Proposed Authentication Solution

Indicator	Result	Comments
Availability	No degradation	Same performance as standard navigation with Galileo I/NAV
Accuracy	No degradation	Same performance as standard navigation with Galileo I/NAV
TTFAF	No degradation	Except when no K-root available or 'slow MAC' required
TBA	10 seconds	5 seconds all-in-view TBA if authentication events are offset
MPT	~1.6 seconds	Every 2 seconds, 40 authentication bits, out of which some are unpredictable, are transmitted
USR	3.68% (of total) 23% (of auth.)	Assuming 276 unpredictable symbols every 30 seconds. 3.68% of total I/NAV symbols (7500 per subframe) 23% of authentication symbols (1200 per subframe)
AER	See Figure 11, Figure 12, Figure 14, Figure 13	No or minimal degradation with respect to standard navigation

about the MAC and key sizes for new chains, allowing maintenance of high system robustness during the system lifetime even in the case of computational and cryptanalysis progresses.

The proposed solution is then characterized, mainly in terms of TBA and AER, which in turn affect availability, accuracy, and TTFAF (time to first authenticated fix). The results show that the proposed implementation does not degrade the performance of a data-authenticated user with respect to a standard user.

The paper also characterizes NMA unpredictability as a relevant design parameter to be maximized. Anti-replay protection is taken into account in the design by the parameters MPT and USR. Table 2 summarizes the performance of the NMA proposal for Galileo I/NAV.

Based on the presented results, and notwithstanding any improvements that may be incorporated in the future, we can conclude that Galileo, through its I/NAV E1-B signal, can provide a highly available and robust NMA service.

ACKNOWLEDGMENTS

The authors would like to especially thank P. Walker and A. Kerns for their insightful comments, the Journal anonymous reviewers for their reviews, H. Tork, the other AALECS team members involved in authentication (C. Holmes, E. Carbonell, O. Pozzobon, S. Fantinato, M. Canale), C. Wullems, R. Ioannides, T. Humphreys, F. Diani, J. Marechal, and the GNSS team at JRC-IPSC, A6 Unit.

DISCLAIMER

The material in this paper does not represent any official view of the EU or its Member States. The solutions proposed are subject to modifications and will not necessarily be included in future Galileo operational services.

REFERENCES

1. John A. Volpe National Transportation Systems Center, "Vulnerability Assessment of the Transport Infrastructure Relying on the Global Positioning System," U.S. DoT, 2001.
2. Wesson, K., Rothlisberger, M., and Humphreys, T., "Practical Cryptographic Civil GPS Signal Authentication," *NAVIGATION*, *Vol*. 59, *No*. 3, *Fall* 2012, pp. 177–193.
3. Günther, C., "A Survey of Spoofing and Counter-Measures," *NAVIGATION*, Vol. 61, No. 3, Fall 2014, pp. 159–177.
4. Wullems, C., Pozzobon, O., and Kubik, K., "Signal Authentication and Integrity Schemes for Next Generation Global Navigation Satellite Systems," *Proceedings of the European Navigation Conference*, 2005.
5. Perrig, A., Canetti, R., Tygar, J. D., and Song, D., "The TESLA Broadcast Authentication Protocol," *CryptoBytes*, *Vol*. 5, *No*. 2, Summer/Fall 2002, pp. 2–13.
6. Humphreys, T., "Detection Strategy for Cryptographic GNSS Anti-Spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, *Vol*. 49, *No*. 2, 2013, pp. 1073–1090.
7. The US Government, "GPS Interface Specification IS-GPS-200," 2014.
8. European Union, "OSSISICD: Open Service Signal In Space Interface Control Document, Issue 1.1," September 2010.
9. Calle, J. D., Carbonell, E., Rodríguez, I., Tobías, G., Göhler, E., Pozzobon, O., Cannale, M., and Fernández, I., "Galileo Commercial Service from the Early Definition to the Early Proof-Of-Concept," *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation* (*ION GNSS+ 2014*), Tampa, FL, September 2014, pp. 392–405.
10. Fernández-Hernández, I., "Authentication: Design Parameters and Service Concepts," *Proceedings of the European Navigation Conference*, 2014.
11. Fernandez-Hernandez, I., Rijmen, V., Seco-Granados, G., Simón, J., and Rodríguez, I., "Design Drivers, Solutions and Robustness Assessment of Navigation Message Authentication for the Galileo Open Service," *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation* (*ION GNSS+ 2014*), Tampa, FL, September 2014, pp. 2810–2827.
12. National Institute of Standards and Technology, "FIPS PUB 180-4: Secure Hash Standard (SHS)," 2012.
13. Lamport, L., "Password Authentication with Insecure Communication," *Communications of the ACM*, Vol. 24, No. 11, November 1981, pp. 770–772.
14. Walker, P., Rijmen, V., Fernandez-Hernandez, I., Simón, J., Calle, D., Pozzobon, O., and Seco-Granados, G., "Galileo Open Service Authentication: A Complete Service Design and Provision Analysis," *Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation* (*ION GNSS+ 2015*), Tampa, FL, September 2015, pp. 3383–3396.
15. Lo, S. and Enge, P., "Authenticating Aviation Augmentation System Broadcasts," *Proceedings of IEEE/ION PLANS 2010*, Indian Wells, CA, May 2010, pp. 708–717.
16. Curran, J. T., Paonni, M., and Bishop, J., "Securing the Open-Service: A Candidate Navigation Message Authentication Scheme for Galileo E1 OS," *European Navigation Conference ENC 2014*, Rotterdam, 2014.
17. Kerns, A. J., Wessons, K., and Humphreys, T., "A Blueprint for Civil GPS Navigation Message Authentication," *Proceedings of IEEE/ION PLANS 2014*, Monterey, CA, May 2014, pp. 262–269.
18. Fernández-Hernández, I. "Method and system to optimise the authentication of radionavigation signals," WO/2015/154981, Filed by the European Union. 8 Apr. 2014.
19. National Institute of Standards and Technology, "FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (HMAC)," 2008.

20. International Organization for Standardization, "ISO/IEC 9797-1:2011: Information Technology - Security Techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms Using a Block Cipher," 2011.
21. Menezes, A., Van Oorschot, P., and Vanstone, S., *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, USA, 1996.
22. Guilford, J., Yap, K., and Gopal, V., "Fast SHA-256 Implementations on Intel Architecture Processors," *IA Architects*, 2012. Available: http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/sha-256-implementations-paper.pdf.
23. Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed-Hashing for Message Authentication," *Network Working Group*, United States, 1997.
24. Lenstra, A. K., "*Key Lengths, Contribution to The Handbook of Information Security*," Lucent Technologies and Technische Universiteit Eindhoven, 1 North Gate Road, Mendham, NJ 07945-3104, U.S.A., 2004.
25. ECRYPT, "Yearly Report on Algorithms and Keysizes," 2012. Available: http://cordis.europa.eu/docs/projects/cnect/6/216676/080/deliverables/002-DSPA20.pdf.
26. ENISA, "Algorithms, Key Sizes and Parameters Report - Recommendations," 2013.
27. National Institute of Standards and Technology, "SP 800-131A: Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths," 2011.
28. National Institute of Standards and Technology, "DRAFT FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," 2014.
29. Fernández-Hernández, I., "Snapshot and Authentication Techniques For Satellite Navigation," Faculty of Engineering and Science, Aalborg University, June 2015.
30. International Organization for Standardization, "ISO/IEC 9796-3. Information Technology – Security Techniques – Digital Signatures Giving Message Recovery – Part 3: Discrete Logarithm Based Mechanisms," 2006.
31. International Organization for Standardization, "ISO/IEC 14888-3: Information Technology - Security Techniques - Digital Signatures With Appendix - Part 3: Discrete Logarithm Based Mechanisms - Amendment 1," 2009.
32. Kaplan, E. and Hegarty, C., *Understanding GPS Principles and Applications*, Artech House, Inc., Boston & London, 2006.
33. Rodríguez, I., Tobías, G., Calle, D., Martín, J., Pozzobon, O., Canale, M., Maharaj, D., Walker, P., Göhler, E., Toor, P., and Fernández-Hernández, I., "Preparing for the Galileo Commercial Service – Proof of Concept and Demonstrator Development," *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation* (*ION GNSS+2014*), Tampa, FL, September 2014, pp. 3399–3410.
34. Scott, L., "Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems," *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, Portland, OR, September 2003, pp. 1543–1552.