Received: 2 March 2021 | Revised: 10 September 2021 | Accepted: 27 December 2021

DOI: 10.33012/navi.519

ORIGINAL ARTICLE

Mapping Bit to Symbol Unpredictability with Application to Galileo Open Service Navigation Message Authentication

Cillian O'Driscoll[1] | **Ignacio Fernández-Hernández**[2]

[1] Independent Consultant, Cork, Ireland

[2] European Commission, Brussels, Belgium

Correspondence
Cillian O'Driscoll, 37 Lake Lawn, Well Rd, Cork T12 YX2A, Ireland.
Email: cillian@ieee.org.
Tel: +353 85 184 1523

Abstract

This paper investigates the distribution of unpredictable symbols in the open service navigation message authentication (OSNMA) scheme, which introduces cryptographic elements into the Galileo I/NAV message. Prior work has described the forward estimation attack (FEA; Curran & O'Driscoll, 2016), that takes advantage of the forward error correction (FEC) employed by the Galileo E1 OS to ensure that a spoofed receiver correctly decodes the I/NAV message, even if it has been generated with errors in some symbols. In order to defend against such an attack, the receiver can re-encode the navigation message into symbols and compare the symbol error rates for those symbols that are predictable and those that are not. In order to perform this, it is first necessary to know which symbols are unpredictable. This paper presents in detail how this can be achieved, including the impact of the cyclic redundancy check (CRC) on symbol unpredictability.

Keywords

authentication, GNSS, OSNMA

1 | INTRODUCTION

Out of the various GNSS anti-spoofing methods proposed in the literature (Psiaki & Humphreys, 2016), *navigation message authentication* (NMA) is a method by which navigation satellites transmit cryptographic data allowing a receiver to authenticate navigation messages transmitted by a GNSS and ensure that the data does not come from another source. NMA has been implemented for the Galileo Open Service through a delayed release symmetric key scheme based on the Timed Efficient Stream Loss-Tolerant Authentication (TESLA) protocol (Perrig et al., 2002), with some adaptations allowing its implementation in GNSS, and in particular in Galileo (Fernandez-Hernandez et al., 2016).

A secondary benefit of NMA is that, if the cryptographic data varies regularly, the navigation messages, which now include cryptographic data, are not predictable, preventing a spoofer from, for example, generating a spoofed navigation message today and broadcasting it tomorrow. If a defense mechanism is implemented in a receiver based on this concept, the receiver must discern which parts, or *bits*, of the message are unpredictable, and which are not. However, this is sometimes not evident. This is particularly the case if the message is convolutionally encoded and has some sort of checksum as the dependencies between the information bits,

checksum bits, and encoded symbols do not make it obvious how to determine the unpredictable symbols.

The first part of this paper is devoted to determining unpredictable symbols after the cyclic redundancy check (CRC) and forward error correction (FEC) encoding and interleaving, based on a stream of predictable and unpredictable bits particularized for the Galileo I/NAV messages (European Union, 2021). Our work is based on Fernández-Hernández and Seco-Granados (2016), which proposed a general scheme to determine unpredictable symbols, but just by checking the number of equations (symbols) and unknowns (bits) for each newly received symbol (therefore without the full mathematical formulation) and omitting the analysis of the CRC. In addition, Cancela et al. (2019) performed a similar analysis and considered CRC bits to be unpredictable, assuming that they depended on the unpredictable bits, but without any mathematical proof. This paper presents, for the first time to the knowledge of the authors, how determining which symbols are unpredictable given the knowledge of unpredictable bits can be achieved for both the convolutional encoding and the CRC, proposing a simple implementation method in the receiver. This method is particularized for Galileo I/NAV and open service navigation message authentication (OSNMA). As a result, for a certain OSNMA chain configuration, the receiver can know a priori the mask of unpredictable symbols for every I/NAV subframe.

In the second part of the paper, the unpredictable symbol mask is used to protect against a signal replay attack, by which the adversary tries to forge the pseudorange measurement while leaving the navigation data unchanged. In order to implement an anti-spoofing defense based on the unpredictable symbol mask, the receiver needs to re-encode the navigation data once it has been successfully decoded. This re-encoding enables the receiver to compare the symbol error rates for those symbols that were known a priori and those that were not. Given the information for which symbols are predictable and which are not, it is then straightforward to design and implement a replay detection mechanism.

The proposed defense mechanism is tested against a forward estimation attack (FEA; Curran & O'Driscoll, 2016). This attack takes advantage of the FEC employed by the Galileo E1 OS in order to ensure that a spoofed receiver correctly decodes the I/NAV message, even if it has been generated with some errors in unpredictable symbols. The critical observation is that the FEC is designed to remove symbol level errors, thereby ensuring the correct decoding of the navigation message. At the same time, this attack does not break the NMA scheme, in that it does not make the receiver vulnerable to spoofed navigation messages, but rather makes it more likely that the receiver will decode the correct message, even if a spoofed message is broadcast. The consequence of a successful FEA attack is that, not only can a signal be replayed, but it can also be advanced in time, which gives an even higher potential advantage to the adversary.

After this introduction, the Galileo I/NAV message structure, including OSNMA, checksum, and coding scheme, are recalled. Next, the symbol unpredictability ignoring the CRC is determined. This is followed by the unpredictable symbol determination also including the CRC. Later, an attack model is described based on the FEA and simulated over a randomly generated bit stream based on a realistic OSNMA configuration. The results of the simulations are presented and the paper finalizes with the conclusions of analysis.

2 | I/NAV MESSAGE STRUCTURE AND UNPREDICTABLE BITS

While the proposed method can be used for other messages, we take the Galileo I/NAV and OSNMA as the reference. The I/NAV message is broadcast in

one-second page intervals. Each page part consists of 250 symbols at a rate of 250 symbols/second. These 250 symbols are further divided into a preamble of 10 symbols, followed by 240 data symbols. These 240 data symbols are derived from 114 bits of navigation data plus six tail bits that are rate ½ encoded using a convolutional coding scheme of constraint length seven. The page parts come in pairs—an even page part followed by an odd page part—to form a two-second page, and 15 such pages are grouped together to form a 30-second subframe. Every odd page part includes a 40-bit OSNMA field (currently named *Reserved 1*), and a 24-bit CRC. After the CRC generation, the symbols output by the convolutional encoder are re-ordered using an 8x30 interleaver matrix that is written one column at a time and read out one row at a time. Further details on the CRC, encoder, and interleaver are illustrated in Figure 1 and presented in the Galileo *Signal-In-Space* (SIS) *Interface Control Document* (ICD; European Union, 2021).

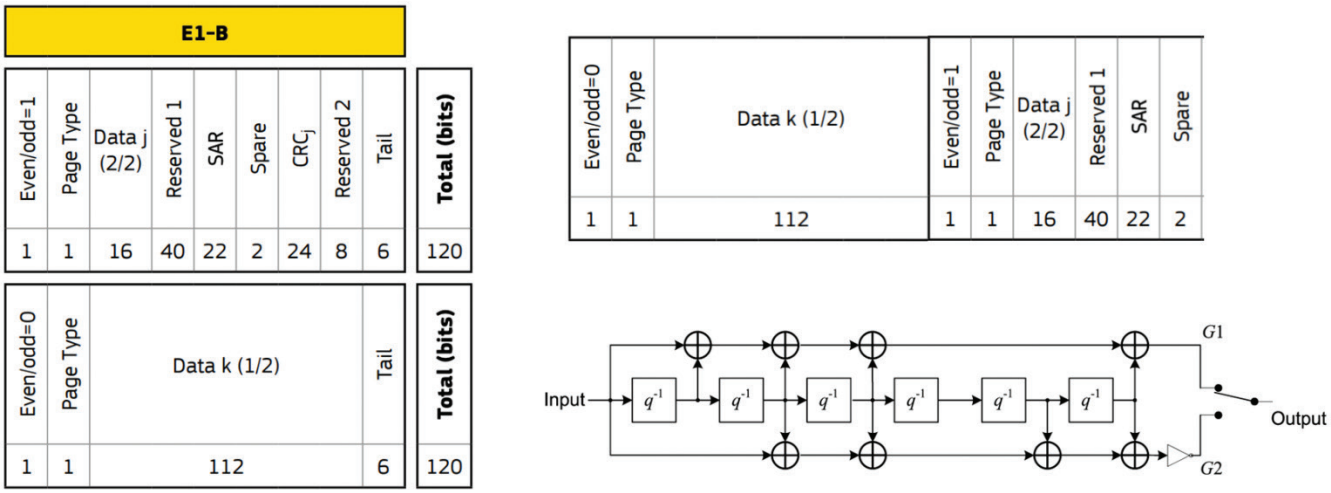


FIGURE 1 Left: I/NAV page structure, including the 40-bit OSNMA field as *Reserved 1* and the 24-bit CRC; Top-Right: Fields that are checked by the CRC (including *Reserved 1*); Bottom-Right: I/NAV FEC scheme

2.1 | OSNMA Field Structure

The OSNMA data field consists of 40 bits (bits 19 through 58 of the odd I/NAV page parts), split into an eight-bit header and root key (HKROOT) section and 32-bit MACK (Message Authentication Code–MAC–and Key) data. The HKROOT section is slowly varying and so, as a conservative bound, we assume that these bits are known a priori, as we do for the rest of the I/NAV message, with the exception of the CRC. The MACK section contains a mix of predictable and unpredictable data depending on the particular configuration of the NMA scheme in force. Therefore, we consider only a maximum of 32 unpredictable bits every odd page part.

The MACK data is arranged in one, two, or three MACK blocks per subframe. There are 480 MACK bits per subframe (15x32 bits), and so either 480, 240, or 160 bits per MACK block. Each block includes a number of MACs (consisting of unpredictable bits), followed by a MAC information section (consisting of predictable bits) and a key. The key bits are considered predictable in our analysis, since key information is shared amongst satellites, so some key information related to the current satellite signal may already have been received from another satellite. This is a conservative assumption.

Therefore, the exact distribution of predictable and unpredictable bits throughout a subframe and within an odd I/NAV page part is a function of the specific OSNMA configuration (i.e., the key and MAC sizes in bits and the number of MACK blocks per subframe). In the following analysis, we consider a sample configuration consisting of the following parameters: Key size (KS) = 96 bits; MAC size (MS) = 32 bits; NMACK (number of MAC & key blocks per subframe) = 2; Number of MACs per MACK block = 3. However, the technique applied will be re-useable for any configuration. The layout of the MACK blocks in this configuration over the 15 pages in a subframe is shown in Figure 2. For further details on the OSNMA SIS specification see European Commission (2018).

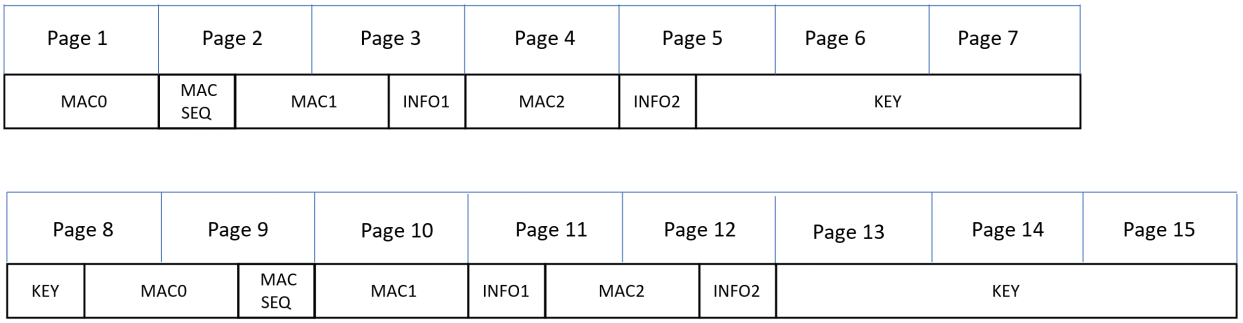


FIGURE 2 Layout of the MACK blocks in one subframe for the chosen OSNMA configuration; only the MAC*N* blocks ($N \in \{0, 1, 2\}$) are considered unpredictable. Note that different OSNMA configurations yield different arrangements of the unpredictable bits within the OSNMA data field.

Note from the figure that, for the OSNMA configuration considered, there are only four possible cases for the 32 MACK bits in the OSNMA field in the odd page parts:

1. All 32 MACK bits are unpredictable.
2. Only the first 16 MACK bits are unpredictable.
3. Only the last 16 MACK bits are unpredictable.
4. All 32 MACK bits are unpredictable.

In the following analysis, we propose a technique for determining which symbols can be considered unpredictable, given only the information on which of the original data bits are unpredictable. First, we introduce some nomenclature.

2.2 | Nomenclature

In the rest of the paper, we will use the following nomenclature for the message symbols encoding the message bits:

- *Known symbols*: symbols generated only from bits that are known a priori (e.g., information that is slowly varying or constant)
- *A-priori unknown symbols*: symbols that depend on at least one bit that is considered unknown, either because it is part of the OSNMA unpredictable information or part of the CRC. The a-priori unknown symbols are divided as follows:
  - *Predictable symbols*: a-priori unknown symbols that, based on the already transmitted symbols, can be determined before transmission

- *Unpredictable symbols*: a-priori unknown symbols that, based on the already transmitted symbols, cannot be determined before transmission

For example, if we take the case in which all MACK 32 bits are unknown in a given page, this would reflect in bits 27, 28…58 for OSNMA, and 83, 84…106, for the CRC. The related *a-priori unknown* symbols would be 53, 54…128, and 165, 166…224, respectively. While the rest, 1…52 and 129…240 depend on known bits, and are, therefore, *known symbols*. The next sections study which of the a-priori unknown symbols are *predictable* and which are *unpredictable.*

3 | UNPREDICTABLE SYMBOL DETERMINATION IGNORING THE CRC

To simplify the analysis, we begin by disregarding the impact of the CRC on symbol predictability and, instead, focusing entirely on those symbols directly affected by the OSNMA *Reserved 1* field (bits 19 to 58 of the odd I/NAV page parts).

3.1 | Methodology

The following analysis is based on the work of Fernández-Hernández and Seco-Granados (2016), where we establish a set of equations for each symbol received. We observe that the convolutional encoding can be written as an affine transformation as follows[1]:

$$\boldsymbol{s} = \boldsymbol{H}_E \boldsymbol{d} + \boldsymbol{c} \tag{1}$$

where $\boldsymbol{s} = [s_1, s_2, \ldots, s_{240}]^T$ is the set of symbols generated and expressed as a vector, $\boldsymbol{d} = [d_1, d_2, \ldots, d_{120}]^T$ is the set of input data bits, $\boldsymbol{H}_E$ is the encoding matrix, and $\boldsymbol{c}$ is a constant with the latter two given by:

$$\boldsymbol{H}_E = \begin{bmatrix} \ddots & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & 0 & 0 & 1 & 1 & 1 & 1 & & \\ & & 1 & 1 & 0 & 1 & 1 & 0 & 1 & & \\ & & & 1 & 0 & 0 & 1 & 1 & 1 & 1 & \\ & & & 1 & 1 & 0 & 1 & 1 & 0 & 1 & \\ & & & & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ & & & & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & \ddots \end{bmatrix} \tag{2}$$

$$\boldsymbol{c} = \begin{bmatrix} 0 & 1 & 0 & 1 & \cdots & 0 & 1 \end{bmatrix}^T$$

[1]Note that a very similar expression could be used, e.g., for the GPS L1 CNAV-2 message, where $\boldsymbol{H}_E$ corresponds to the Low-Density Parity Check (LDPC) encoding matrix and $\boldsymbol{c} = \boldsymbol{0}$.

The (1,0) stream shown in $\boldsymbol{H}_E$ is based on the encoding polynomials shown in Figure 1, and the vector $\boldsymbol{c}$ reflects the inversion of G2, also shown in Figure 1. If we split the data vector $\boldsymbol{d}$ into two components: $\boldsymbol{k}$ as the vector of known bits (zeros at all unknown bit locations) and $\boldsymbol{u}$ as the vector of unknown bits (zeros at all known bit locations), then we can write:

$$\begin{aligned} \boldsymbol{d} &= \boldsymbol{k} + \boldsymbol{u} \\ \boldsymbol{s} &= \boldsymbol{H}_E\boldsymbol{k} + \boldsymbol{c} + \boldsymbol{H}_E\boldsymbol{u} \\ &= \boldsymbol{s}_{\boldsymbol{k}} + \boldsymbol{s}_{\boldsymbol{u}} \end{aligned} \tag{3}$$

Where $\boldsymbol{s}_{\boldsymbol{k}}$ is the vector generated from the known bits and $\boldsymbol{s}_{\boldsymbol{u}}$ is the vector of symbols generated from unknown bits:

$$\begin{aligned} \boldsymbol{s}_{\boldsymbol{k}} &= \boldsymbol{H}_E\boldsymbol{k} + \boldsymbol{c} \\ \boldsymbol{s}_{\boldsymbol{u}} &= \boldsymbol{H}_E\boldsymbol{u} = \boldsymbol{s} - \boldsymbol{s}_{\boldsymbol{k}} = \boldsymbol{s} + \boldsymbol{s}_{\boldsymbol{k}} \end{aligned} \tag{4}$$

Where we replace subtraction with addition as they are equivalent modulo 2. As each symbol s_i is received, the attacker can compute $s_{u,i} = s_i + s_{k,i}$, and this adds a new equation relating the unknown bits $\boldsymbol{u}$ to the already received symbols. Each new symbol corresponds to a row in the matrix $\boldsymbol{H}_E$, say the i-th symbol received yields the j-th row of $\boldsymbol{H}_E$, where:

$$j = 1 + 8\,\mathrm{mod}(i-1,30) + \left\lfloor \frac{i-1}{30} \right\rfloor \tag{5}$$

which reflects the effect of interleaving. Let $\tilde{\boldsymbol{H}}_{E,i}$ denote the matrix constructed from the rows of $\boldsymbol{H}_E$ given by the first i unknown symbols, and consisting of only those columns that correspond to the unknown bits in $\boldsymbol{u}$. Then the number of linearly independent rows in $\tilde{\boldsymbol{H}}_{E,i}$ is given by its matrix rank, which can be easily computed. When a new symbol is received, the updated matrix is obtained, and its rank is computed. If the rank so computed is the same as the rank given the previous symbol, then the new row is a linear combination of rows of $\tilde{\boldsymbol{H}}_{E,i-1}$, and so the symbol s_i is *predictable* given symbols up to and including s_{i-1}. Also, let N_u denote the number of unknown bits (and so equal to the number of columns in $\tilde{\boldsymbol{H}}_{E,i}$), then a sufficient and necessary condition for determining all elements of $\boldsymbol{u}$ is:

$$\mathrm{rank}\,\tilde{\boldsymbol{H}}_{E,i} = N_u \tag{6}$$

The following algorithm can therefore be used to determine the predictable symbols of the I/NAV odd page part under OSNMA (ignoring the CRC):

1. Input: list of all a-priori unknown symbols (in order received), list of unpredictable bits, the number of symbols in advance to predict A
2. Initialize: set the modified encoding matrix $\tilde{\boldsymbol{H}}_{E,0}$ to be the empty matrix, and its rank $r_0 = 0$
3. For each a-priori unknown symbol, s_i, in the order received:
   a. Compute j, the index of the symbol prior to interleaving
   b. Compute the j-th row of $\boldsymbol{H}_E$ and extract only those columns that depend on the unknown bits $\tilde{\boldsymbol{r}}_j$
   c. Concatenate this row to the matrix $\tilde{\boldsymbol{H}}_{E,i-A}$ to obtain:

$$\tilde{\boldsymbol{H}}_{E,i} = \begin{bmatrix} \tilde{\boldsymbol{H}}_{E,i-1} \\ \tilde{\boldsymbol{r}}_j \end{bmatrix} \tag{7}$$

d. Calculate the rank of $\tilde{\boldsymbol{H}}_{E,i-A+1}$: r_i
   a. If $r_i = r_{i-1}$, then add s_i to the list of predictable symbols
   b. If $r_i = N_u$, then add all remaining symbols (not including s_i) to the list of predictable symbols and terminate. The remaining a-priori unknown symbols are the unpredictable symbols.

In the following sections, we apply this methodology to three OSNMA field cases based on the aforementioned configuration (KS = 96, MS = 32, NMACK = 2). One can see that, in this configuration, there are three possible cases which are presented hereafter.

3.2 | Case 1: 32 Unpredictable Bits in OSNMA Field

In this case, bits 27 to 58 of the odd page part are unknown, and these bits affect symbols 53 to 128 inclusive. The CRC bits are left out of the analysis for the moment and will be covered in the next section. Running the algorithm for determining the unpredictable symbols for an advance of one symbol, we obtain the results illustrated in Figure 3, where white symbols are *known symbols* (CRC symbols are not incorporated in the analysis yet, so they are still considered known), green symbols are (a-priori unknown) *unpredictable symbols*, and yellow symbols are (a-priori unknown) *predictable* symbols.

In other words, at a point in time before any symbols of the page part are transmitted, none of the colored symbols are known. As the symbols are received, those symbols colored green remain unknown until such a point in time as they are received, while those colored yellow are known at least one symbol in advance of being received. In the figure, the symbols are received in order from left to right starting from the top-left corner, and starting on the left-hand column in each row, much like reading lines of text.

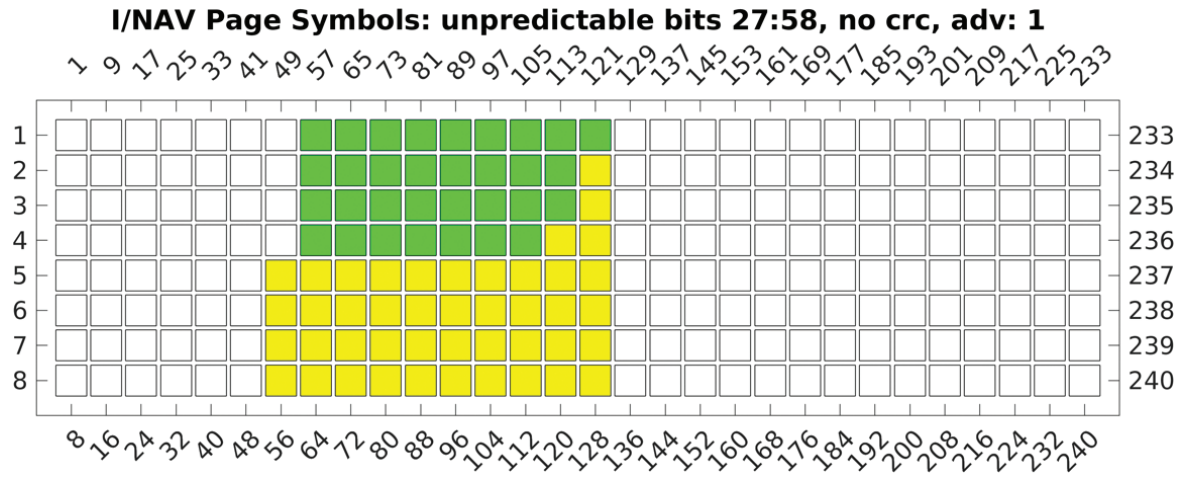


FIGURE 3 Layout of the predictable and unpredictable symbols for Case 1, without accounting for the CRC; predictable symbols are shown in yellow, while unpredictable symbols are in green

Note that this result differs from the work of Fernández-Hernández and Seco Granados (2016) in that the symbols s_{122} and s_{123} are here shown to be predictable. The difference is due to the fact that Fernández-Hernández and Seco-Granados (2016) assumed linear independence amongst the equations. Note also that, in accordance with their work, there are precisely 32 unpredictable OSNMA symbols.

3.3 | Case 2: 16 Unpredictable Bits in OSNMA Field (First 8 Bits Known, Next 16 bits Unknown, Last 16 Bits Known)

In this case, bits 27 to 42 of the odd page part are unknown, and these bits affect symbols 53 to 96 inclusive (*a-priori unknown symbols* 165 to 224). Again, running the symbol prediction algorithm with an advance of one symbol leads to the results in Figure 4.

Again, in this case, there are two predictable symbols within the first 16 OSNMA-dependent symbols, so a naïve approach would mis-identify these as unpredictable. There are exactly as many unpredictable OSNMA-dependent symbols as there are unknown bits, and again we have ignored the CRC.

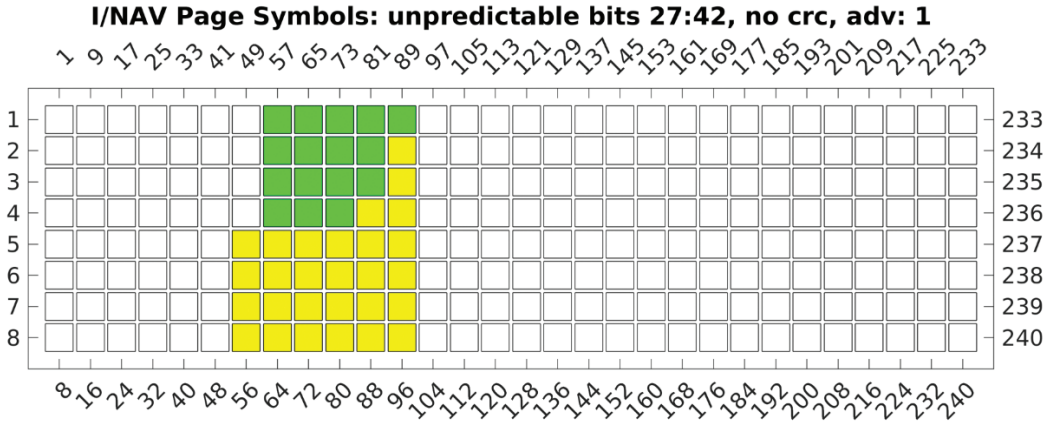


FIGURE 4 Layout of the predictable and unpredictable symbols for Case 2; predictable symbols are highlighted in yellow. Note that symbols 90 and 91 are predictable. CRC is excluded.

3.4 | Case 3: 16 Unpredictable Bits in OSNMA Field (First 24 Bits Known, Last 16 Bits Unknown)

In this case, bits 43 to 58 of the odd page part are unknown, and these bits affect symbols 85 to 128 inclusive. Running the symbol prediction algorithm yields the results in Figure 5, which are consistent with the other two cases.

4 | UNPREDICTABLE SYMBOLS INCLUDING THE CRC

While the results of the previous section show how to account for the possible linear dependence between unknown symbols, they ignore the impact of the CRC. In this section, we show how the CRC can be incorporated into the symbol prediction algorithm to give the receiver designer the correct view of which symbols are, in fact, unpredictable.

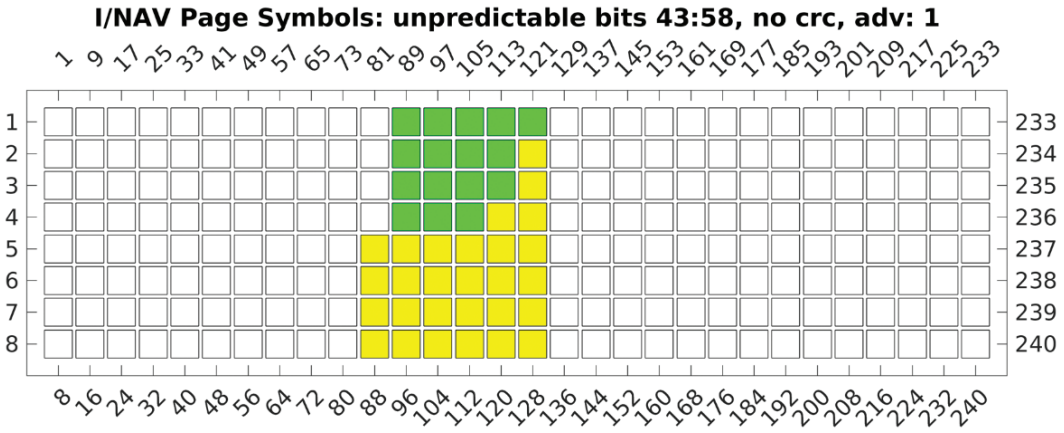


FIGURE 5 Layout of the predictable and unpredictable symbols for Case 3; predictable symbols are highlighted in yellow. Note that symbols 122 and 123 are predictable. CRC is excluded.

4.1 | Methodology

The methodology employed to account for the CRC relies on the following observations:

1. The CRC can be expressed as a linear mapping from the input bits to CRC bits.
2. This linear mapping can be computed symbolically with a simple algorithm.

The first observation is well known, though, it is not obvious. The second observation appears to be new and will be shown below. Let us write the vector of data bits as follows:

$$\boldsymbol{d} = \left[\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}, \boldsymbol{r}, \boldsymbol{c}\right]^T \tag{8}$$

where $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are known bits, $\boldsymbol{t}$ is the vector of unknown OSNMA bits, and $\boldsymbol{r}$ is the vector of CRC bits. Then, following the methodology of the preceding section, we write:

$$\boldsymbol{k} = \left[\boldsymbol{a}, \boldsymbol{0}, \boldsymbol{b}, \boldsymbol{r}_k, \boldsymbol{c}\right]^T \tag{9}$$

Where here $\boldsymbol{r}_k$ is the CRC computed from the known bits of the navigation message with the other bits set to zero. Similarly, we write:

$$\boldsymbol{u} = \left[\boldsymbol{0}, \boldsymbol{t}, \boldsymbol{0}, \boldsymbol{r}_u, \boldsymbol{0}\right]^T \tag{10}$$

where $\boldsymbol{r}_u$ is the CRC computed from only the unknown bits with the other bits set to zero. Now, due to the linearity of the CRC, we have:

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{r}_k + \boldsymbol{r}_u \\ \therefore \boldsymbol{d} &= \boldsymbol{k} + \boldsymbol{u} \end{aligned} \tag{11}$$

So, we can apply the same methodology as in the previous section, provided that we can find a suitable matrix mapping from $\boldsymbol{u}$ to $\boldsymbol{s}_u$. Now, the computation of the CRC can be expressed as a long division operation, where the numerator $N(x)$ and

denominator $G(x)$ are both expressed as polynomials over GF(2). For the Galileo I/NAV CRC, the numerator is formed as follows (European Union, 2021):

$$N(x) = m(x)x^{24} \tag{12}$$

where $m(x)$ is the message for which the CRC is to be computed. This can be written as:

$$m(x) = e(x)x^{64} + t(x)x^{24} + l(x) \tag{13}$$

where $e(x)$ is a polynomial of degree 131, representing the known 132 bits of the CRC message prior to the OSNMA bits, $t(x)$ is a degree-39 polynomial representing OSNMA bits, and $l(x)$ is a degree-23 polynomial representing the known 24 bits of the CRC message occurring after the OSNMA bits. Following the methodology outlined in the appendix, we can write the CRC for the unknown bits as follows:

$$\boldsymbol{r_u} = \boldsymbol{H}_\rho \boldsymbol{t} \tag{14}$$

and so, we can write:

$$\boldsymbol{u} = \left[\boldsymbol{0}, \boldsymbol{I}, \boldsymbol{0}, \boldsymbol{H}_\rho, \boldsymbol{0}\right]^T \boldsymbol{t} \tag{15}$$

Now that we have a simple expression for $\boldsymbol{u}$, we can re-use the methodology above and compute:

$$\boldsymbol{s_u} = \boldsymbol{H}_E \boldsymbol{u} = \boldsymbol{H}_E \left[\boldsymbol{0}, \boldsymbol{I}, \boldsymbol{0}, \boldsymbol{H}_\rho, \boldsymbol{0}\right]^T \boldsymbol{t} \tag{16}$$

In the following sections we use this approach to determine the unpredictable symbols for the OSNMA cases shown in Figure 2.

4.2 | Case 1: 32 Bits Unknown

The results of this analysis are shown below:

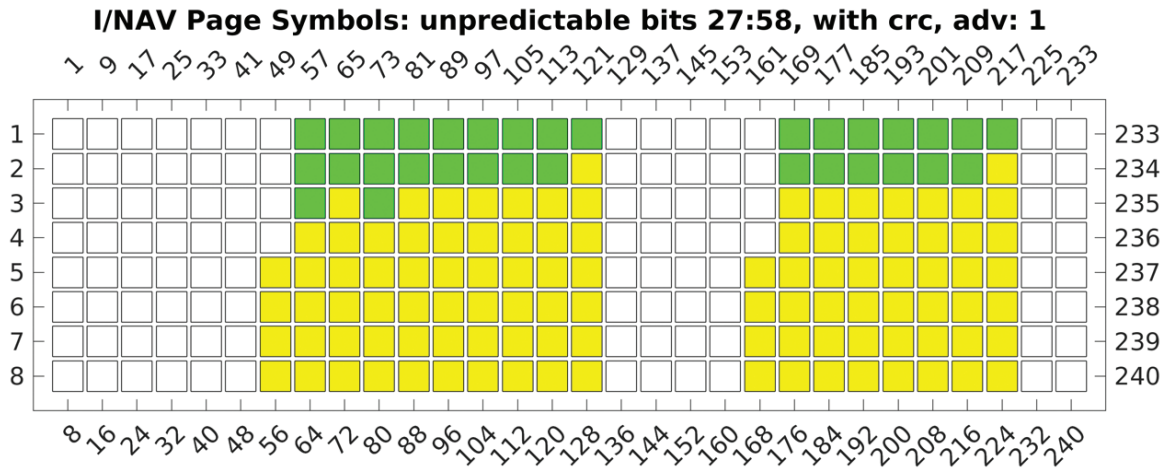


FIGURE 6 Location of unpredictable symbols for I/NAV Case 1 when accounting for the CRC; known symbols are white, unpredictable symbols are green, and predictable symbols are yellow.

Note that symbols 67, 122, and 218 are predictable, and that 32 unpredictable symbols must be received before the attacker has full knowledge of all the unknown data bits. Contrasting this with Figure 3, we see the symbols become predictable much earlier than without accounting for the CRC, and symbol 75 is the last unpredictable symbol received. On the other hand, most of the early CRC symbols are unpredictable, for the same total of 32 bits.

4.3 | Case 2: 16 Unpredictable Bits in OSNMA Field (First 8 Bits Known, Next 16 Bits Unknown, Last 16 Bits Known)

The results for Case 2 are shown below. Note again that only 16 unknown symbols are received before the attacker can potentially have full knowledge of all the unknown bits. In this case, there are three linear dependencies in the first 19 received symbols.

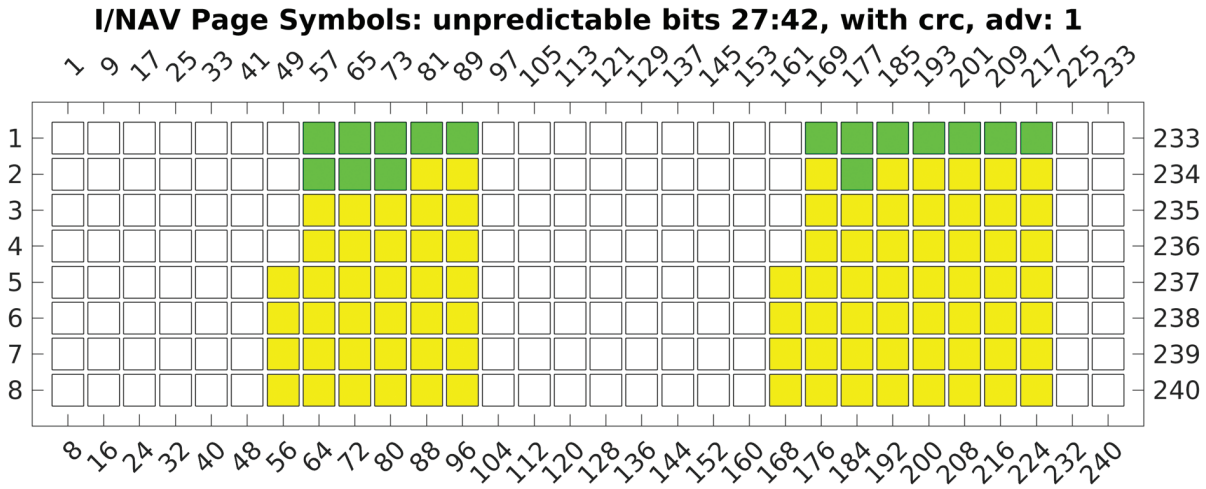


FIGURE 7 Location of unpredictable symbols for I/NAV Case 2 when accounting for the CRC; known symbols are white, unpredictable symbols are green, and predictable symbols are yellow.

4.4 | Case 3: 16 Unpredictable Bits in OSNMA Field (First 24 bits Known, Last 16 Bits Unknown)

The results for Case 3 are shown on the next page, and are very similar to the previous results. Again, only 16 of the a-priori unknown symbols need to be received before the attacker can potentially have full knowledge of all the unknown bits.

Finally, Figure 9 presents the aggregation of the unpredictable symbols in a Galileo OSNMA configuration with a 96-bit key size, 32-bit MAC size, 2 MAC-key blocks per subframe (left), and 1 MAC-key block per subframe (right), considering the keys to be predictable. This figure encompasses the results of Figure 6, Figure 7, and Figure 8. One can see that most unpredictable parts are placed at the beginning of the subframe, as shown in the previous figures. It also shows a fully predictable period when the key is transmitted, as it is considered predictable in this work for reasons mentioned previously.

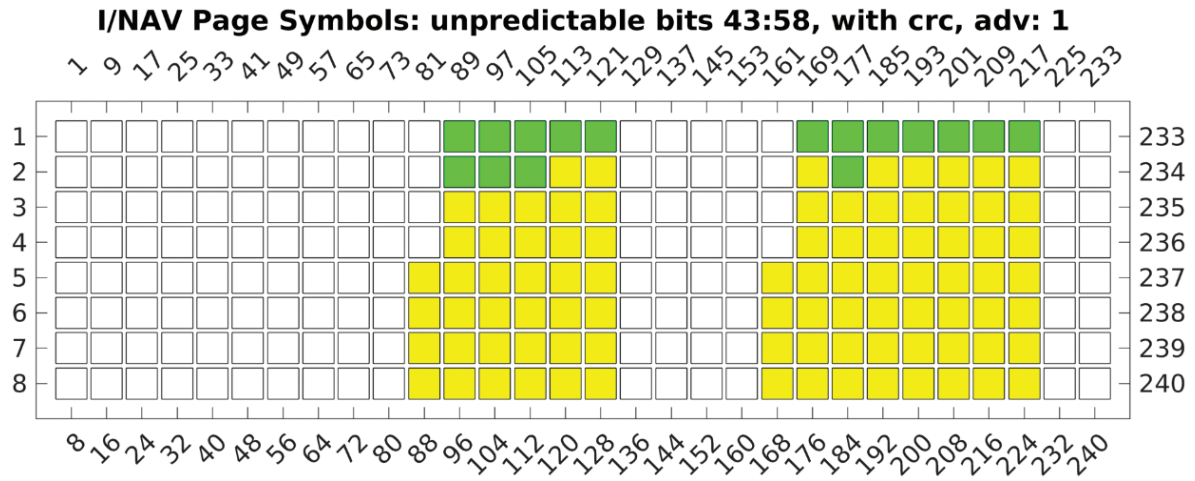


FIGURE 8 Location of unpredictable symbols for I/NAV Case 3 when accounting for the CRC; known symbols are white, unpredictable symbols are green, and predictable symbols are yellow.

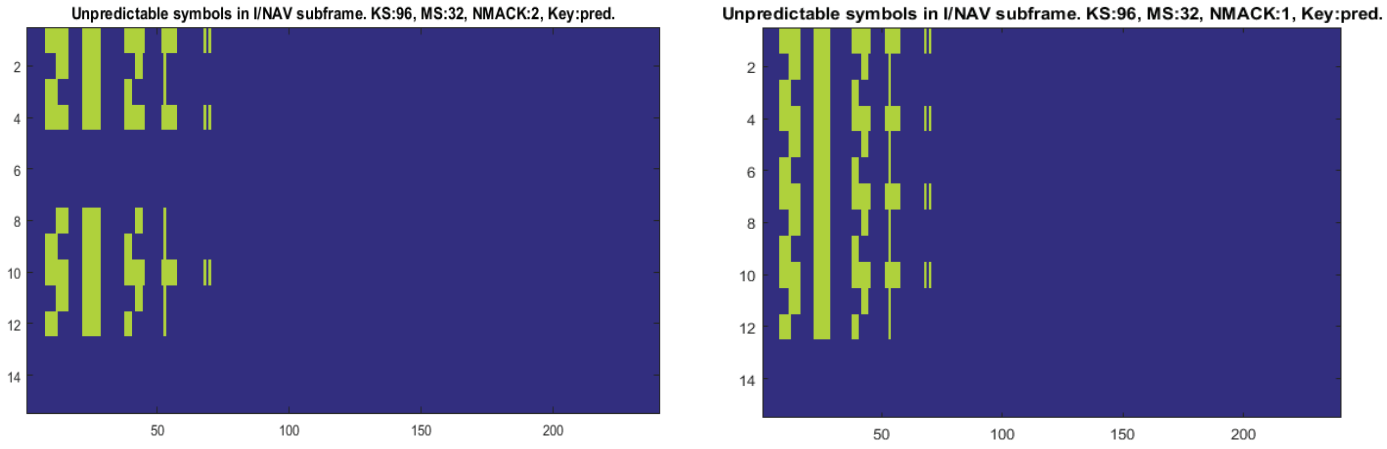


FIGURE 9 Aggregated unpredictable (green) and predictable (blue) symbols in a 30-second I/NAV subframe; each row represents a single I/NAV odd page part of 240 symbols, with one symbol per column.

5 | ATTACK SIMULATION RESULTS

To verify the above analysis, a set of simulations were conducted using a Candidate E1 I/NAV OSNMA subframe structure as a starting point. In each simulation, the attacker generated a spoofed signal a number of symbols in advance of the currently received symbol (the advance is varied from one to eight symbols). The resulting symbols were decoded using a standard hard-decision Viterbi decoder. The decoded bits are re-encoded to determine which symbols were received in error. In each case, 100 subframes were spoofed and the results were computed by averaging across these 100 separate subframes.

5.1 | Verification of Symbol Unpredictability

The first stage in the verification is to analyze the symbol error rates for the symbols generated by the spoofer. Figure 10 shows the modeled unpredictable

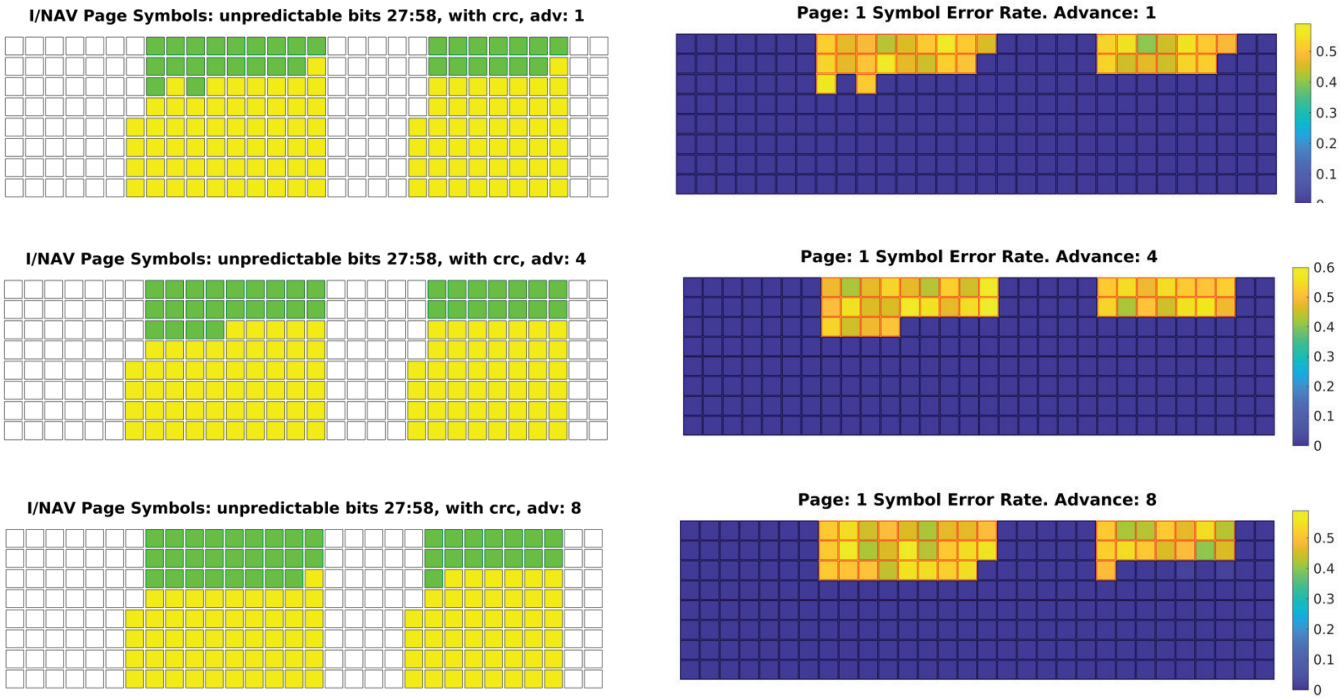


FIGURE 10 Validation of symbol predictability models; Left: Modeled predictability; Right: symbol error rates from the spoofed I/NAV odd page parts; Top to bottom: symbol advances of one, four, and eight symbols

symbols and the symbol error rates measured across all 100 of the Page 1 messages generated by the spoofer. The figure shows results for spoofer advances of one, four, and eight symbols. Note that the symbol error rate for all known and predictable symbols is zero across all spoofer advances, and the symbol error rate is approximately 0.5 for all unpredictable symbols. Note also that the number of unpredictable symbols increases as the symbol advance increases. Thus, the baseline of a single symbol advance represents the most conservative estimate from the defender's perspective, and only these symbols should be considered unpredictable in the design of a symbol-level defense against a forward estimation attack.

Similar results were obtained for all the other I/NAV pages. This shows that the models derived above correctly account for both the linear dependence between the symbols and the effect of the CRC. The list of unpredictable symbols can be used by a receiver to detect a FEA by analysis of the error rates for these symbols.

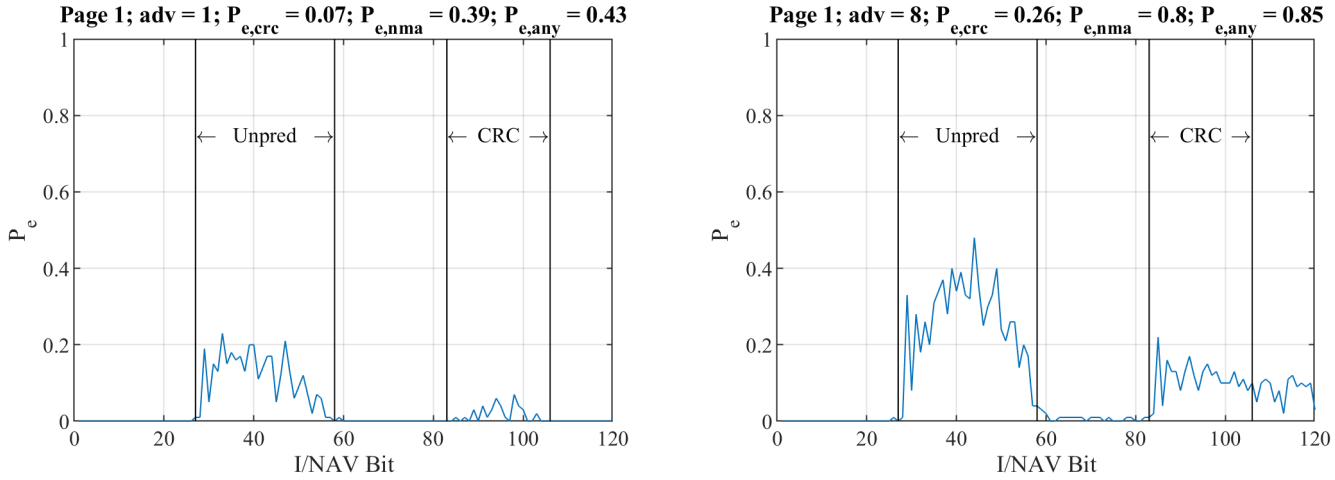
5.2 | Decoder Results

The preceding results demonstrate that the models are correct, but there is a complex interaction between the symbol errors and their manifestation as bit errors. A successful FEA attack requires: a) that the receiver decodes the correct NMA data bits; b) that the receiver decodes the correct CRC (we assume the receiver requires the CRC to pass before using the NMA bits); and c) that the receiver does not cross-check the actual received symbols and the expected symbols given a correct decoding of the navigation message.

In this section, we analyze the results of a hard-decision Viterbi decoder applied to the spoofed symbols, under the assumption that all the spoofed symbols are

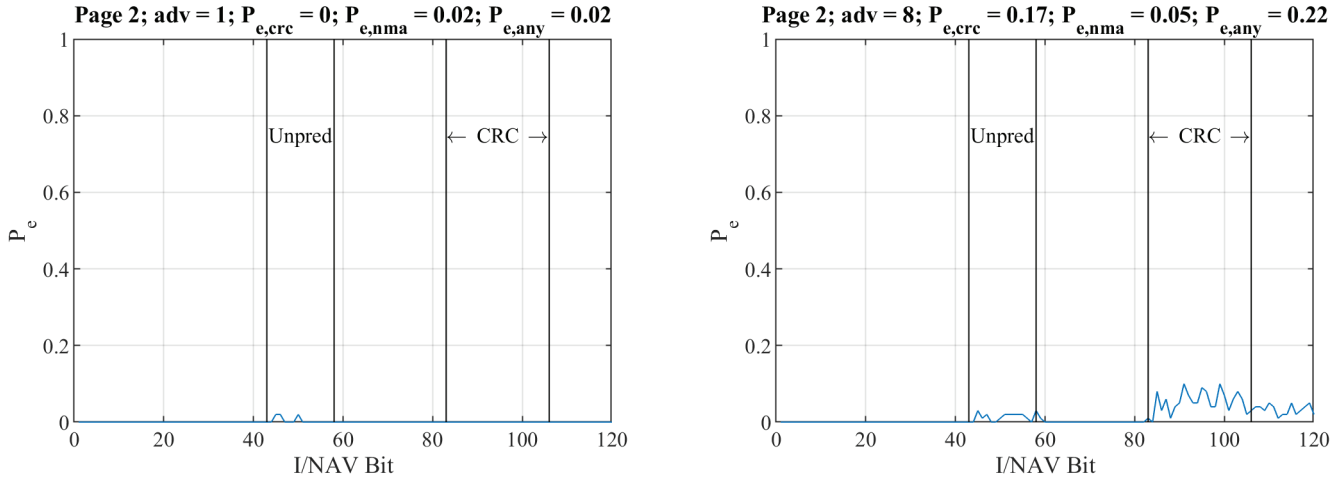
received without error (in the sense that the receiver extracts precisely those symbols transmitted by the spoofer, even if these symbols are not the same as those generated by the satellite). This represents a lower bound on the probability of correctly decoding a FEA spoofed navigation message, since Curran and O'Driscoll (2016) show that the use of soft-decision decoding (and appropriate weighting of the symbols by the attacker) can result in much higher probabilities of successful decoding by the receiver.

Figure 11 shows the distribution of bit errors across the decoded odd page parts for Pages 1 and 2 (corresponding to 32 and 16 unknown bits, respectively) for both one- and eight-symbol advance attacks. We assume that the attacker takes advantage of the CRC (i.e., the received CRC symbols are used to aid in predicting other a-priori unknown symbols, as shown in Figure 10 as shown in Section 4). Note that the overall probability of a bit error varies from 0.02 for the one-symbol advance attack on Page 2, to 0.85 for the eight-symbol advance attack on Page 1. For a single symbol advance, the attacker has a greater than 50% chance of a successful attack



a) Page 1: One Symbol Advance Attack

b) Page 1: Eight Symbol Advance Attack



c) Page 2: One Symbol Advance Attack

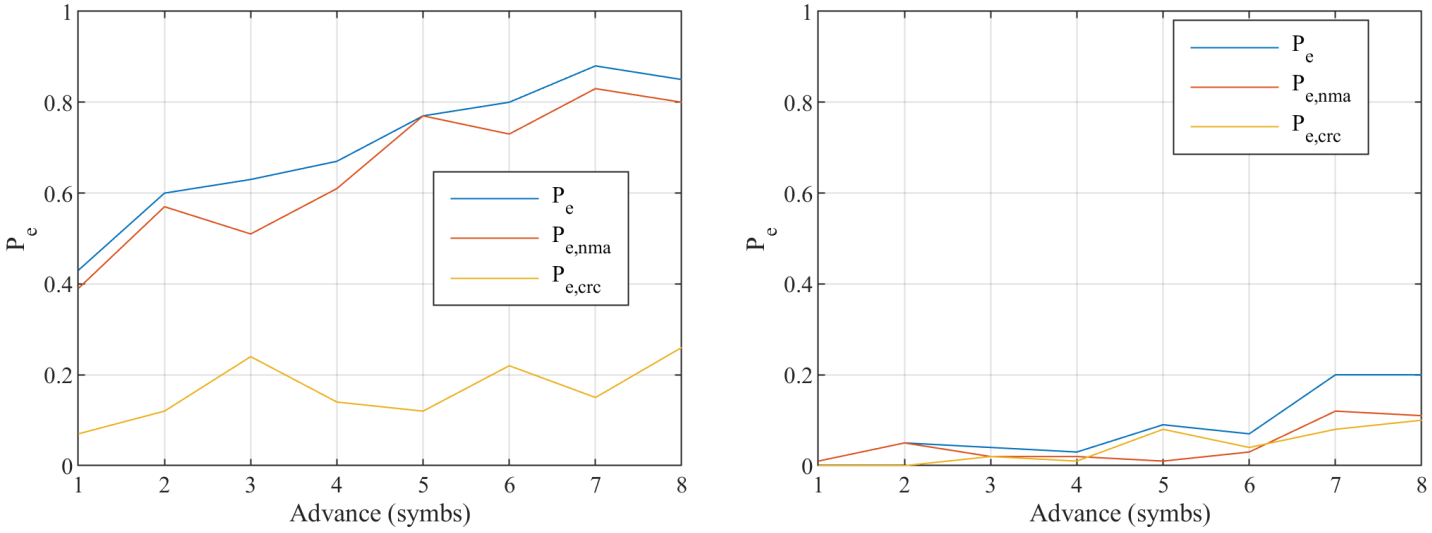
d) Page 2: Eight Symbol Advance Attack

FIGURE 11 Bit errors for hard-decision Viterbi decoder applied to the FEA attack using CRC on I/NAV. Top: Page 1 (all 32 NMA bits unknown); Bottom: Page 2 (last 16 NMA bits unknown); Left: One symbol advance FEA; Right: Eight symbol advance FEA

(in the sense that the probability that the true bits will be successfully decoded by a hard-decision Viterbi decoder is greater than 50%).

The plots show that the bit errors are evenly distributed throughout the unknown NMA bits and the CRC bits, but also that the bits after the CRC are potentially decoded in error.

The impact of the symbol advance is shown in more detail in Figure 12 below, which presents the overall probability of a bit error and the probabilities of bit error for the NMA and CRC bits for both Page 1 and Page 3, with symbol advances from one to eight symbols. In this case, P_e denotes the probability of error in any bit within the page, while $P_{e,nma}$ denotes the probability of error in any bit in the 40-bit NMA field, and $P_{e,crc}$ denotes the probability of error in any of the CRC bits. As expected, the probability of a successful attack (essentially one minus the probability of a bit error) decreases as the symbol advance increases, but overall, the probability of successful attack is high. This confirms the necessity of cross-checking the symbol error rates in the unpredictable symbols as a mechanism for detecting a FEA attack.



a) Page 1: 32 Unknown Bits

b) Page 3: First 16 Bits of NMA Unknown

FIGURE 12 Error rates for the hard-decision Viterbi decoding of FEA symbols.; each plot shows three probabilities of error: P_e, the probability that any one bit is in error; $P_{e,nma}$, the probability that one of the NMA bits is in error; and $P_{e,crc}$, the probability that one of the CRC bits is in error.

6 | CONCLUSION

This paper has investigated the distribution of unpredictable symbols in a GNSS message introducing unpredictable bits which are later verified by a message authentication scheme, in which the message can include error correction and detection mechanisms. In particular, the analysis is specialized for Galileo OSNMA, which is transmitted in 40 bits every other second in the I/NAV message. The I/NAV message also includes a 24-bit CRC and is convolutionally encoded and interleaved.

We consider *known* symbols as those fully derived from known bits, and *a-priori unknown* symbols as those derived from unknown bits. The latter are divided into

predictable symbols and unpredictable symbols. *Predictable symbols* are those that, at the time of transmission, can be determined based on the reception of the previous symbols, while *unpredictable symbols* are those that cannot be determined before reception.

A simple method is proposed to determine which symbols are unpredictable. The method defines a system of equations in which each a-priori unknown symbol received adds one equation to a linear system. It refines previous work by evaluating the rank of the system matrix with every new symbol, as every new equation may be linearly dependent on the previous ones, which occurs in a few cases. The application of the method to the CRC is based on two observations: that the CRC can be expressed as a linear mapping from the input bits to CRC bits; and that this linear mapping can be computed symbolically with a simple algorithm. Based on this process, a receiver can define by a priori a mask of unpredictable symbols and use it to detect potential message replay attacks, including forward estimation attacks (FEA).

The method is particularized for Galileo OSNMA with a configuration of 96-bit keys, 32-bit MACs, two MAC and key blocks per subframe, and three MACs per block, considering the keys to be predictable. An FEA attack was generated and the results were evaluated by investigating the distribution of the spoofed symbol errors and the impact of the spoofed navigation message on a hard-decision Viterbi decoder. It was shown that the proposed models are correct, which highlights the necessity of utilizing an anti-FEA defense in which the decoded data are re-encoded and the error rates of the unpredictable symbols are evaluated. The results showed that the error rate for these symbols should be 50%, irrespective of the symbol advance employed by the attacker. The results also show that the more symbols in advance the attacker operates, the more symbols are unpredictable to the attacker. This might be expected, as the attacker has potentially received less information about the symbol before the time of transmission when the advance is large.

The same approach can be applied directly to other convolutionally encoded schemes, such as satellite-based augmentation system (SBAS) signals, and can be trivially extended to work for any other encoding scheme that can be expressed as an affine transformation. For example, the GPS C/NAV message is encoded using a low-density parity check code, which can be expressed as a matrix product between the encoding matrix and the vector of data bits. Thus, this same approach can be used in the context of the proposed GPS chips message robust authentication (Chimera) scheme (Air Force Research Laboratory, 2019).

While the specifics of a defense against the FEA are not considered in this work, a rough outline is as follows. The defense takes on the form of a binary hypothesis test, wherein under the null hypothesis, the distribution of symbol errors within a page part should follow a Gaussian model, depending only on the received signal-to-noise ratio (SNR) values per symbol. Under the alternate hypothesis, the distribution of the symbol errors is a function of whether or not the symbol is predictable. For predictable symbols, the error distribution should be the same as that under the null hypothesis, while for unpredictable symbols, the error rate would be 50%, irrespective of the SNR per symbol. Knowing both the SNR per symbol and the distribution of the unpredictable symbols is essential for the correct implementation of such a defense.

Further work may characterize the time an adversary needs to estimate every a-priori unknown predictable symbol, or characterizing a replay attack probability of detection and false alert in a real environment, where all symbol types may be

corrupted due to signal impairments. Also, this symbol-level analysis may be complemented with lower-level signal analysis to attempt detecting tracking lift-off replay attacks based on the unpredictable parts of the signal.

REFERENCES

Air Force Research Laboratory. (2019). *IS-AGT-100: Chips message robust authentication (Chimera) enchancement for the L1C signal: Space segment/user segment interface.* Air Force Research Laboratory, Space Vehicles Directorate, Advanced GPS Technology.

Cancela, S., Navarro, J., Calle, D., Reithmaier, T., Chiara, A. D., Broi, G. D., Fernández-Hernández, I., Seco-Granados, G., & Simón, J. (2019). Field testing of GNSS user protection techniques. *Proc. of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, FL, 1824–1840.* https://doi.org/10.33012/2019.17087

Curran, J. T., & O'Driscoll, C. (2016). Message authentication, channel coding, & anti-spoofing. *Proc. of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, OR, 2948–2959. https://doi.org/10.33012/2016.14670

European Commission. (2018). *Test platform on Galileo HAS/CAS/OSNMA.* https://etendering.ted.europa.eu/cft/cft-documents.html?cftId=6271

European Union. (2021). *OSSISICD: Open service signal-in-space interface control document (Issue 2.0).* https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf

Fernández-Hernández, I., & Seco-Granados, G. (2016). Galileo NMA signal unpredictability and anti-replay protection. *2016 International Conference on Localization and GNSS (ICL-GNSS),* Barcelona, Spain. https://doi.org/10.1109/ICL-GNSS.2016.7533686

Fernandez-Hernandez, I., Rijmen, V., Seco-Granados, G., Simon, J., Rodriguez, I., & Calle, J. D. (2016). A navigation message authentication proposal for the Galileo open service. *NAVIGATION, 63*(1), 85–102. https://doi.org/10.1002/navi.125

Perrig, A., Canetti, R., Tygar, J. D., & Song, D. (2002). The TESLA Broadcast Authentication Protocol. *CryptoBytes, 5*(2). https://people.eecs.berkeley.edu/~tygar/papers/TESLA_broadcast_authentication_protocol.pdf

Psiaki, M. L., & Humphreys, T. E. (2016). GNSS spoofing and detection. *Proc. of the IEEE, 104*(6), 1258–1270. https://doi.org/10.1109/JPROC.2016.2526658.

How to cite this article: O'Driscoll, C., & Fernández-Hernández, I. (2022). Mapping bit to symbol unpredictability with application to Galileo open service navigation message authentication. *NAVIGATION, 69*(2). https://doi.org/10.33012/navi.519

APPENDIX: DERIVATION OF LINEAR DEPENDENCE BETWEEN CRC AND INPUT BITS

Here we show how the CRC bits can be written as a linear combination of the unknown bits. Assume that we have transformed our message such that there are N_u unknown bits followed by m zeros, so that the message over which the CRC is to be computed is given by:

$$m(x) = u_1 x^{m+N_u-1} + u_2 x^{m+N_u-2} + \cdots + u_{N_u-1} x^{m+1} + u_{N_u} x^m \tag{17}$$

The CRC computation can be written as a long division as follows:

$$
\begin{array}{rl}
 & \quad f_1 x^{m+N_u-N_g} + f_2 x^{m+N_u-1-N_g} + \cdots \\
G(x) & \overline{\left| u_1 x^{m+N_u-1} + u_2 x^{m+N_u-2} + \cdots + u_{N_{u-1}} x^{m+1} + u_{N_u} x^m \right.} \\
 & \quad f_1 g_1 x^{m+N_u-1} + f_1 g_2 x^{m+N_u-2} + \cdots + f_1 g_{N_g} x^{m+N_u-N_g} \\
 & \overline{(u_2 + f_1 g_2) x^{m+N_u-2} + (u_3 + f_1 g_3) x^{m+N_u-3} + \cdots} \\
 & \quad f_2 g_1 x^{m+N_u-2} + f_2 g_2 x^{m+N_u-3} + \cdots \\
 & \overline{\qquad\qquad\qquad\qquad \vdots}
\end{array}
\tag{18}
$$

$$
= \begin{array}{rl}
 & F(x) \\
G(x) & \overline{\left| \; m^1(x) \right.} \\
 & P^1(x) \\
 & \overline{m^2(x)} \\
 & P^2(x) \\
 & \overline{\vdots}
\end{array}
$$

such that $m(x) = F(x)G(x) + r(x)$. We have introduced the notation $m^i(x)$ to denote the remaining denominator at the i-th iteration of the long division, so that $m(x) = m^1(x)$, and $P^i(x)$ is the i-th product polynomial $P^i(x)$. Here:

$$
P^1(x) = p_1 x^{m+N_u-1} + p_2 x^{m+N_u-2} + \cdots + p_{m+N_u} = f_1 G(x) x^{m+N_u-N_g} \tag{19}
$$

Now f_1 is chosen to ensure that the highest order term of $P^1(x)$ equals the highest order term of $m^1(x)$:

$$
f_1 g_1 = u_1 \Rightarrow f_1 = u_1 \tag{20}
$$

In fact, in general f_i is the highest order term in $m^i(x)$, but as the order of $m^i(x)$ decreases by one as i increases by one, we have: $f_i = m_i^i$

So, the i-th product polynomial $P^i(x)$ is given by: $P^i(x) = f_i G(x) x^{m+N_u-N_g+1-i} = m_i^i G(x) x^{m+N_u-N_g+1-i}$

Finally, the next numerator polynomial $m^{i+1}(x)$ is computed as: $m^{i+1}(x) = m^i(x) + P^i(x)$

The process continues until the degree of $m^i(x)$ is less than the degree of $G(x)$, at which point $m^i(x)$ is the remainder sought:

$$
r(x) = m^{m+N_u-N_g+1} \tag{21}
$$

We can express each of the polynomials as a vector where the i-th element represents the coefficient of the i-th highest power:

$$
\begin{aligned}
\boldsymbol{g} &= \left[g_1, g_2, \ldots, g_{N_g} \right]^T \\
\boldsymbol{m^1} &= \left[u_1, u_2, \ldots, u_{N_u}, 0, 0, \ldots, 0 \right]^T \\
\boldsymbol{p^1} &= \left[p_1^1, p_2^1, \ldots, p_{m+N_u}^1 \right]^T
\end{aligned}
\tag{22}
$$

Now we can express $\boldsymbol{m^i}$ in terms of the initial message $\boldsymbol{m^1}$ as follows: $\boldsymbol{m^i} = \boldsymbol{N^i m^1}$

where $\boldsymbol{N^i}$ is a $(m+N_u)\times(m+N_u)$ matrix. Given that the last m elements of $\boldsymbol{m^1}$ are zero, then we can write:

$$\boldsymbol{N^1} = \begin{bmatrix} \boldsymbol{I}_{N_u,N_u} & \boldsymbol{0}_{N_u,m} \\ \boldsymbol{0}_{m,N_u} & \boldsymbol{0}_{m,m} \end{bmatrix} \tag{23}$$

The polynomial $P^i(x)$ can be expressed as:

$$\boldsymbol{p^i} = \boldsymbol{g^i}\boldsymbol{N^i_{i,-}}\boldsymbol{m^1} \tag{24}$$

Where $\boldsymbol{g^i}$ is the length $m+N_u$ representation of the polynomial $G(x)x^{m+N_u-N_g+1-i}$:

$$\boldsymbol{g^i} = \left[\boldsymbol{0}_{1,i-1}, \boldsymbol{g^T}, \boldsymbol{0}_{1,m+N_u-N_g+1-i}\right]^T \tag{25}$$

So putting this together, we obtain the matrix relationship:

$$\boldsymbol{m}^{i+1} = \left(\boldsymbol{N}^i + \boldsymbol{g}^i\boldsymbol{N}^i_{i,-}\right)\boldsymbol{m}^1 \tag{26}$$

This gives us a recursive equation for computing $\boldsymbol{N}^i$:

$$\boldsymbol{N}^{i+1} = \boldsymbol{N}^i + \boldsymbol{g}^i\boldsymbol{N}^i_{i,-} \tag{27}$$

By iterating this procedure until $i = m+N_u-N_g+1$ we obtain a relationship between the input message $\boldsymbol{m}^1$ and the CRC bits $\boldsymbol{r}$:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{0}_{m+N_u-N_g+1,1} \\ \boldsymbol{r} \end{bmatrix} &= \boldsymbol{N}^{m+N_u-N_g+1}\boldsymbol{m}^1 \\ &= \boldsymbol{N}^{m+N_u-N_g+1}\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{0}_{m,1} \end{bmatrix} \end{aligned} \tag{28}$$

So, we can write the linear relationship:

$$\boldsymbol{r} = \boldsymbol{H}_\rho\boldsymbol{u} \tag{29}$$

Where $\boldsymbol{H}_\rho$ is the matrix obtained from the last N_g-1 rows and first N_u columns of $\boldsymbol{N}^{m+N_u-N_g+1}$.