Field Testing of GNSS User Protection Techniques

S. Cancela, J. Navarro, D. Calle, *GMV*
T. Reithmaier, *Ifen*
A. Dalla Chiara, G. Da Broi, *Qascom*
I. Fernández-Hernández, *European Commission*
G. Seco-Granados, *Universidad Autónoma de Barcelona*
J. Simón, *European GNSS Agency*

BIOGRAPHIES

Simón Cancela holds an MSc in Advanced Mathematics by the Universidad Complutense de Madrid. He joined GMV in 2015 and he has been working in projects related with the design and development of GNSS authentication and anti-spoofing algorithms, including the Galileo Commercial Service Demonstrator and Commercial Service enhanced PVT resilient platform.

Javier Navarro holds a BSc in Systems Telecommunications Engineering by the Polytechnic University of Madrid. He joined GMV in 2018 and he is currently working on the development of a Commercial Service enhanced PVT resilient platform.

David Calle holds a MSc. in Computer Engineering from the University of Salamanca. He joined GMV in 2008 and he has been working in the GNSS business unit involved in the design and development of GNSS algorithms, applications and systems. He is currently Head of GNSS Services Section coordinating the activities related to the Galileo Commercial Service, Open Service Authentication and High Accuracy provision services.

Andrea Dalla Chiara is designer and project manager at Qascom, with focus on GNSS simulators and receivers and authentication techniques both at signal and data level. He is an electronic engineer, and has a PhD in Information Technologies by University of Padova.

Giacomo Da Broi is a designer and developer of GNSS and TT&C simulators at Qascom, specialized in authentication techniques both at signal and data level. He holds an MSc degree in Telecommunications Engineering from University of Padova.

Tobias Reithmaier received his Certified Engineer from the Technikerschule (Engineering School) of Munich. He worked as a technical responsible for EC NACSET receiver project development, testing and documentation; NTR FPGA receiver product and enhancement, integration and test of latest NOVA STX RF signal generator among other.

Ignacio Fernández Hernández is in charge of Galileo high accuracy and authentication at the European Commission, DG GROW. He is an ICAI engineer, holds an MBA by LBS and a PhD in electronic systems by Aalborg University.

Gonzalo Seco Granados is associate professor with the Dept of Telecom. Eng. of Univ. Autónoma de Barcelona (UAB) and head of the Signal Processing for Navigation and Communications (SPCOMNAV) group. Previoulsy, he was staff member at the Radionavigation Section in ESTEC/ESA, and involved in the Galileo project and in the development of GNSS receivers and applications.

Javier Simon is Service Design Engineer within the European GNSS Agency, currently contributing to the definition and design of the Galileo OSNMA and CS services. He holds a MSc. degree in Telecommunications Engineering from the Polytechnic University of Madrid, Spain. Before joining GSA he participated in several projects for the study and design of future GNSS algorithms and systems

ABSTRACT

GNSS is a key element for a wide range of applications in our daily lives. Mass-market applications such as sports tracking or user guidance, liability-critical applications such as banking and telecommunication time synchronization, and safety critical services such as aviation and automotive-related solutions, all rely on GNSS. The huge growth experimented during the last decade puts GNSS in the target of attackers.

The Galileo program is complementing the Galileo Open Service with Navigation Message Authentication (OSNMA) and providing signal authentication through the Commercial Service signals. These new services will be able to provide added protection to the current GNSS applications. Nevertheless, these features will require the users to implement new algorithms to exploit them. In this context, the European Commission launched the Navigation Authentication through Commercial Service-Enhanced Terminal (NACSET) project aiming at researching and implementing different techniques to detect and mitigate thus improving the resilience at user-level.

In the frame of the NACSET project, a user terminal has been developed based on a high-end multi-GNSS receiver that is able to track E1/L1 and E6-B/C signals for data and signal protection. The terminal is equipped with a set of resilience techniques. Among these techniques, this paper focuses on an anti-replay technique protecting against zero-delay Secure Code Estimate-Replay (SCER) attacks based on the analysis of the unpredictable symbols from OSNMA cryptographic data.

This paper firstly describes the NACSET project and its aim. Secondly, the theory of Anti-replay protection is explained from the point of view of a receiver, and anti-replay techniques based on OSNMA are introduced. Then, we describe the SCER simulator developed to assess the performances of the technique. To conclude, an attack is defined and performed with the SCER simulator over a real receiver. The results with and without OSNMA replay protection are presented and explained, and some conclusions of the experiment are derived.

INTRODUCTION

The Navigation Authentication through Commercial Service-Enhanced Terminals (NACSET) project aims at evaluating different techniques to exploit Galileo authentication services (OSNMA and Commercial Authentication Service) in the next generation of resilient receivers, and integrate them with receiver-based resiliency checks. NACSET has developed a resilient User Terminal that can integrate Galileo authentication services with specific equipment such as IMUs, barometers and high-performance clocks to support standalone spoofing detection techniques. In order to provide the PVT resilience thanks to assisted authentication, the project includes a Synchronization and Authentication Server (SAS). This module is the responsible for providing assisted navigation and signal authentication capabilities to the User Terminal as well as accurate time synchronization. For the distribution of the cryptographic information, such as the Commercial Service keys and OSNMA information, a dedicated Key Management Simulator is also developed to emulate all the required interfaces [1].

The focus of this paper is to present the implementation and results of a specific technique aimed to protect against signal replay attacks. This technique is based on previous theoretical work ([2], [3]) on the anti-replay protection capabilities of a GNSS signal containing Navigation Message Authentication (NMA) data. Throughout this paper the theoretical details of the technique will be presented alongside the actual implementation in NACSET user terminal, ending with a performance assessment done with a simulator specifically developed for this purpose.

ANTI-REPLAY PROTECTION

If a GNSS signal stream contains data that is authenticated, a spoofer can only alter the pseudorange measurements to spoof the receiver position. This attack falls under the category of signal replay attacks. In order to protect pseudoranges from replay attacks, the pseudoranges can include authentication features. Ideally, these authentication features can be implemented at spreading-code level. However, if the data modulated includes unpredictable symbols, this unpredictability can be also exploited against replay attacks. The level of protection offered by data/symbol unpredictability, as opposed to spreading code measures, has been a subject

of debate in the GNSS literature. While some references consider both data and code level as almost equally effective under certain conditions [4], others consider data-level measures as very limited.

One of the novel features of the User Terminal is the inclusion of anti-replay protection measures based on the unpredictability bits of NMA data present on a GNSS signal, as introduced in [5].

Some cryptographic information included in the GNSS data stream is unpredictable for a real-time attacker. For this reason, an attacker has to estimate this unpredictable data. The case we will assess in this paper in the case of a zero-delay SCER attack [5] in which the spoofer transmits a signal replica that is perfectly aligned with an already tracked true signal. In this case, the NMA unpredictable bits, encoded into unpredictable symbols, can be considered as 'security codes' as per the nomenclature of [5], and the attacker needs to estimate them on the fly, as described in [6]. Therefore, a zero-delay SCER attack consists of estimating and reproducing the original signal with zero, or negligible, delay, to take control of the tracking loop. Once the attacker has the control, it starts to delay the signal in order to forge the receiver PVT solution.

The NMA solution taken as reference for this implementation is the Galileo Open Service NMA (OSNMA) protocol proposed in [7]. This service is to be included in the navigation information transmitted on the Galileo E1-B signal (I/NAV). I/NAV data is structured in frames every 750 seconds, composed by 25 subframes of 30 seconds duration each. Every subframe is divided into fifteen 2-second pages, each of which contains one word and some other fields [8]. All the data is convolutionally encoded and interleaved, to prevent data errors. The convolutional code has a ratio of 1/2, which means that it returns twice the input data. In each of the two subpages of a 2-second page, there are 120 data bits, and at the end of this conversion function we obtain 240 encoded symbols. After adding at the beginning the 10-symbol pattern needed for synchronization, the 250 symbols are transmitted through the SIS. Every satellite has his own spreading code of 4092 bits (or chips). Each symbol transmitted is modulated by one spreading code array, which means that the 4092 modulation chips are transmitted per symbol.

While standard receivers perform continuous signal correlation, the idea behind the technique proposed in this work is to perform partial correlations using different subsets of the signal chips on every symbol to be able to detect the imperfect estimation in case of an attack [3]. The high-level concept behind this technique is to detect correlation losses in the parts of the symbols the spoofer is estimating incorrectly due to the lack of information. The purpose of these methods is to perform correlations in different parts of the unpredictable symbols of the signal and compare them with the predictable data correlations. The chips are stored until the symbols are authenticated by the navigation message authentication mechanism and the analysis of the partial correlation is performed to detect a possible attack:

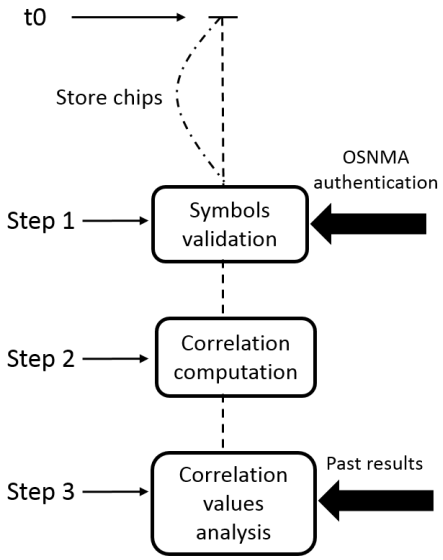

Figure 1 – Anti-replay technique high-level processing

OSNMA UNPREDICTABLE SYMBOLS

This section presents the structure of unpredictable symbols in the OSNMA message. Galileo OSNMA is transmitted in 40 of the 240 data bits of each 2-second page of the Galileo E1-B I/NAV signal. Precisely, there are included only in the odd page, as shown in Figure 2.

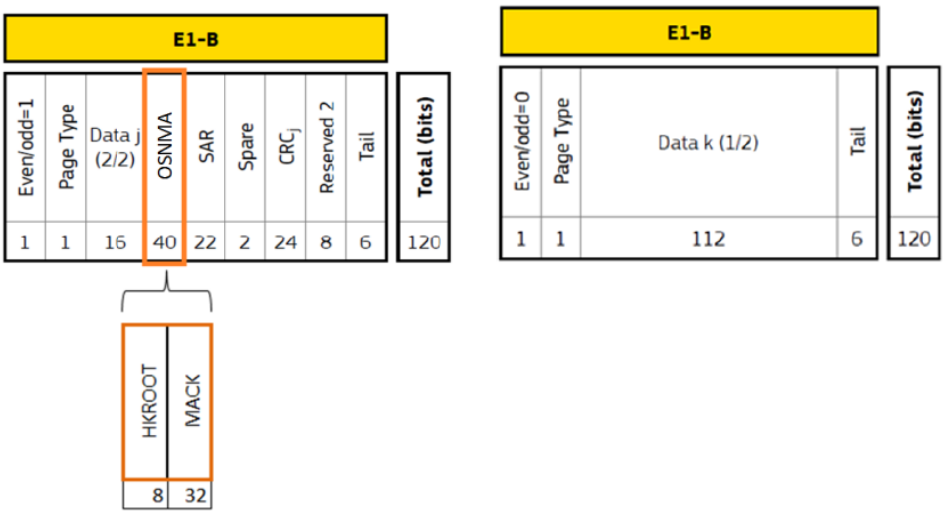

Figure 2 – I/NAV 2-second page

The OSNMA data is sent in two different sections: the HKroot (Header and Root Key) section, containing the Digital Signature Messages (DSMs); and the MACK (MAC and Key) section which contains Message Authentication Codes (MACs) and related symmetric cryptographic keys, later disclosed, as per the OSNMA GNSS adaptation of the TESLA (Timed Efficient Stream Loss-Tolerant Authentication) protocol [5]. The DSMs cannot be considered unpredictable because they are repeated over time. This DSM field is transmitted in 8 bits, included in the 40 bits of the OSNMA. The MACs and keys are transmitted in the remaining 32 bits, which we will consider within this paper as unpredictable. This is a simplification of the actual symbol unpredictability rate but does not alter the conclusions. Other considerations may be taken into account, such as considering the last bits of the keys and some MAC information as predictable, as explained in [2].

Another field sent in the I/NAV pages that should be also taken into account to determine the position of the unpredictable symbols is the Cyclic Redundant Check (CRC), used to detect possible information losses in the Navigation Data caused by noise in transmission channels. A CRC is included in each page of the I/NAV Galileo message, broadcast in the E1B signal. This CRC is calculated from that page block of data, providing redundancy information. The 24 bits of the CRC used in the E1B signal are also included in the odd subpage, as OSNMA bits. As earlier stated, 32 bits are considered unpredictable due to their cryptographic content. Assuming that the other data of the page could be fully known in advance, the CRC calculation is done using unpredictable and predictable information. To calculate the CRC code, an attacker needs all the navigation data bits. If these 32 bits are considered unknown by an attacker, then, the CRC code included could be considered *a priori* as unknown too. When those 32 unknown bits are estimated by the attacker, the CRC calculation could be done successfully.

1	9	17	25	33	41	49	57	65	73	81	89	97	105	113
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120

Figure 3 – I/NAV odd subpage

Figure 3 shows the 120 bits of an odd subpage of the I/NAV navigation data. The yellow bits are considered a priori unknown (OSNMA and CRC bits). Each CRC bit depends on 220 bits of the I/NAV E1B Galileo page (except the 'Reserved 2' and the 'Tail' fields, see [8]). Because only 32 of these 220 are unpredictable, an attacker will have only 32 unknown values in the CRC equations. Each of the 24 equations of the CRC is as follows:

$$\boldsymbol{CRC_n = g(bn_1, bn_2, \dots, bn_{188}) + f(bu_1, bu_2, \dots, bu_{32})} \quad (1)$$

where $\boldsymbol{n}$ is the index of the CRC bit, $\boldsymbol{bn}$ is each known bit and $\boldsymbol{bu}$ is each unknown bit. Each CRC equation could be written as the addition of two functions, one depending on the unknown bits ($\boldsymbol{f}()$), and the other depending on the known bits ($\boldsymbol{g}()$). In order to determine the CRC bits, the attacker needs the value of all navigation data bits. That makes the 24 bits of the CRC dependent of OSNMA bits.

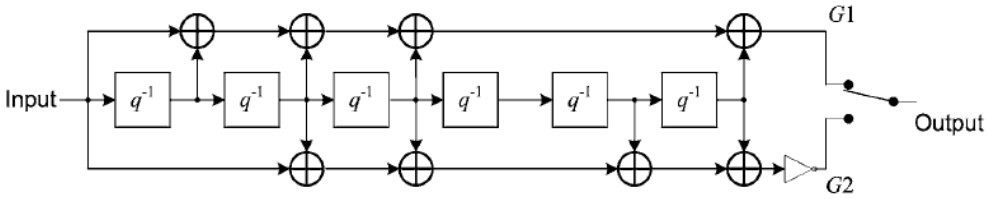

Figure 4 – I/NAV convolutional

The convolutional diagram of the E1B Galileo signal shown in Figure 3 can be given by the following equations:

$$\boldsymbol{s_{2n-1} = b_n + b_{n-1} + b_{n-2} + b_{n-3} + b_{n-6}} \quad (2)$$

$$\boldsymbol{s_{2n} = b_n + b_{n-2} + b_{n-3} + b_{n-5} + b_{n-6} + 1}$$

where $\boldsymbol{s}$ is each output symbol ($\boldsymbol{s_{2n-1}}$ is G1 and $\boldsymbol{s_{2n}}$ is G2), $\boldsymbol{b}$ is each data bit ($\boldsymbol{b_n}$ is the input bit) and $\boldsymbol{n}$ is the index of the data bit (from 1 to 120). After the convolutional process, the output of the system for the 120 input data bits is 240 symbols. The entropy of the signal will not be increasing by the convolutional code, and only one symbol for each unpredictable bit is considered as unpredictable, the other symbols having redundant information. That means that there are only 56 unknown symbols. Nevertheless, 136 symbols contain information about the unpredictable bits, 76 output symbols of the unpredictable bits, and 60 output symbols of the CRC.

1	9	17	25	33	41	49	57	65	73	81	89	97	105	113	121	129	137	145	153	161	169	177	185	193	201	209	217	225	233
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114	122	130	138	146	154	162	170	178	186	194	202	210	218	226	234
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	171	179	187	195	203	211	219	227	235
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124	132	140	148	156	164	172	180	188	196	204	212	220	228	236
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117	125	133	141	149	157	165	173	181	189	197	205	213	221	229	237
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118	126	134	142	150	158	166	174	182	190	198	206	214	222	230	238
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127	135	143	151	159	167	175	183	191	199	207	215	223	231	239
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240

Figure 5 – I/NAV encoded symbols

All the output symbols calculated from an unknown are considered unknown. For that reason, from the 32 unknown bits of the OSNMA field, 76 is the number of the unknown symbols, 64 due to an unknown bit as convolutional input and 12 due to an unknown bit in a convolutional register. Because the convolutional diagram has 6 registers and 2 output for each input, an input bit will affect the following 12 outputs symbols. The same with the 24 unknown bits of the CRC, 60 unknown symbols are generated.

After convolutional encoding, the symbols are interleaved inverting the transmission of the matrix 8 rows x 30 columns shown in Figure 4. After the reception of the first 28 symbols (from 1 to 217), the attacker has received 16 unknown symbols:

$$[s_{57}\ s_{65}\ s_{73}\ s_{81}\ s_{89}\ s_{97}\ s_{105}\ s_{113}\ s_{121}\ s_{169}\ s_{177}\ s_{185}\ s_{193}\ s_{201}\ s_{209}\ s_{217}] \quad (3)$$

With these 16 symbols, the attacker has 16 equations with the 56 unknown values. With the previous 24 CRC equations, now the attacker has 40 equations. After the reception of the following 30 symbols (from 225 to 218) the adversary has received 16 unknown symbols more:

$$[s_{58}\ s_{66}\ s_{74}\ s_{82}\ s_{90}\ s_{98}\ s_{106}\ s_{114}\ s_{122}\ s_{170}\ s_{178}\ s_{186}\ s_{194}\ s_{202}\ s_{210}\ s_{218}] \quad (4)$$

With these 16 unknown symbols more, the attacker has 16 equations more, which means, 56 equations and 56 unknown values. Now the attacker has the capacity of decode all bits of the subpage, and from them, code all the symbols of the subpage. For that reason, only the first 32 unknown symbols can be considered as unpredictable. After broadcasting 58 symbols the whole page can be known. This means that after 232 ms added to the computation time needed to resolve the equations, all the symbols broadcast could be considered as predictable. In order to develop Anti-replay techniques, it necessary to focus on the first 32 unknown symbols transmitted, including CRC symbols.

To sum up, the Anti-replay technique should focus on the first 32 unpredictable symbols to be certain of its unknown behavior for an attacker.

IMPLEMENTATION

In this section the actual implementation of the technique will be presented. The NACSET User Terminal (UT) consists on two main modules: the hardware receiver and an external software module installed in a laptop PC. The technique is implemented in the external software module as part of the Authentication Engine of the UT. This Authentication Engine the outputs of the technique to a PVT engine which is in charge of computing the resilient position.

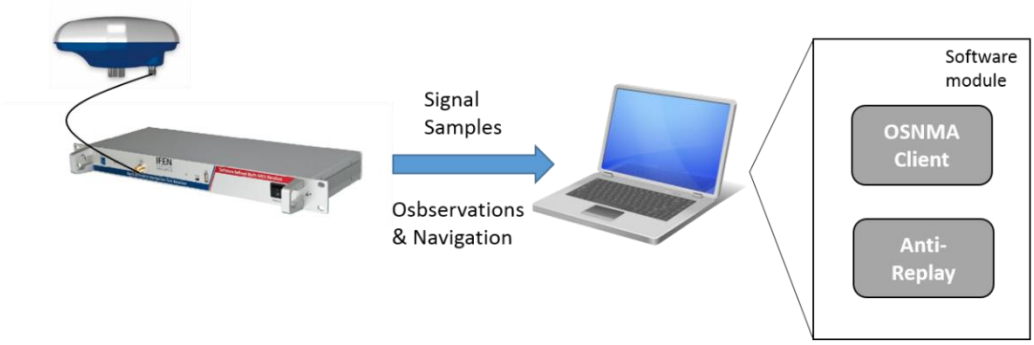

Figure 6 – User Terminal (UT) Architecture

The receiver is in charge of sending the signal samples and the navigation to the PC, which includes the Anti-replay module and the OSNMA client to perform the data authentication. Then the Anti-replay module is in charge of conducting the partial correlations of the signal to detect the attack. Two different detection metrics have been implemented in the receiver. These detection metrics are obtained from [3]. The two selected for this implementation are the following:

- Partial correlation analysis
- C/N_0 estimation

These techniques compare the beginning and the end samples of each unpredictable symbol. In normal behavior, a SCER may estimate wrongly the first chips of each unpredictable symbol, while the last part of the symbol will be practically predictable. The following techniques draw conclusions from comparing both values. $B_{beg}(k)$ and $B_{end}(k)$ are the partial correlations of the beginning and end of each unpredictable symbol k, after removing the sign of the symbol.

The partial correlation analysis consists on computing the average of the difference between the beginning and end partial correlation of the symbols:

$$\left|\frac{1}{N_b}\sum_{k=1}^{N_b} B_{beg}(k) - B_{end}(k)\right| \quad (5)$$

where N_b is the number of unpredictable symbols used in the technique and k is the number of chip samples stored to make the correlation. If the value is high, it means the receiver may be under an attack. One could argue that an attacker could also degrade the samples at B_{end} intentionally, in order to avoid any difference in the correlation with B_{beg}. However, the receiver can take random intervals at the end of the symbol, or samples from predictable symbols, forcing the attacker to degrade the signal continuously, with an appreciable effect in the receiver. The second of these methods is based on using a traditional Narrowband-Wideband Power Ratio (NWPR) estimator to estimate the C/N_0 values at the beginning and at the end of each symbol. The NWPR is calculated based on the ratio of wideband power to its narrowband power.

$$NP = \frac{NBP}{WBP} \quad (6)$$

where NP is the normalized power between the narrow-band power (NBP) and wide-band power (WBP). They are calculated from the partial samples of the begging and the end of each unpredictable symbol:

$$WBP = \left(\sum_{k=1}^{N_b}|B_x(k)|^2\right) \quad (7)$$

$$NBP = \left(\left|\sum_{k=1}^{N_b} B_x(k)\right|^2\right) \quad (8)$$

where $B_x(k)$ is even the beginning ($B_{beg}(k)$) or the end of the symbol ($B_{end}(k)$). The C/N_0 partial estimations results are calculated the code integration time in seconds (T_{coh}):

$$\widehat{C/N_0} = 10\ log_{10}\left(\frac{1}{T_{coh}}\frac{NP-1}{N_b-NP}\right) \quad (9)$$

At last, we compare the beginning with the end, a high value will mean an attack is being conducted:

$$\left|\widehat{C/N_{0_{beg}}} - \widehat{C/N_{0_{end}}}\right| \quad (10)$$

Both methods work independently and are included in the Anti-replay module. However, the result of the anti-spoofing check should be sent to the receiver logic. This result informs the receiver whether it is being spoofed or not. The Anti-replay technique in the Anti-replay module needs therefore to define a threshold in order to conclude a result for the receiver.

The software accumulates the correlation values of N_b symbols and stores them in a sliding window to be analyzed. The size of the window is a configurable number defined by the user. With these stored values, in the implementation described in this paper, the software calculates the mean $\bar{x}$ and the standard deviation σ:

$$\bar{x} = \frac{\sum_{i=0}^{N} x_i}{N} \quad (11)$$

$$\sigma = \sqrt{\frac{\sum_{i=0}^{N}(x_i - \bar{x})^2}{N}} \quad (12)$$

where N is the number of values stored in the sliding window and x_i is each value of the partial correlations/C/N_0 estimations difference. With these two values, the technique defines a dynamic threshold for each satellite.

$$Threshold = |\bar{x}| + \sigma \quad (13)$$

To calculate this threshold, the software needs as many samples as possible. If their mean overtakes the defined threshold several times, then the technique concludes that a spoofed signal is present. The size of the sliding window defines the number of times that the mean has to overtake the threshold, to avoid false alarms. This number is also configurable by the user. Other techniques based on a fixed threshold defined by statistical analysis, as proposed in [3], can be also used.

SCER ATTACK SIMULATION

In order to assess the performances of the technique and due to the fact that OSNMA is not yet transmitted by in the SIS, a SCER attack simulator has been developed. The SCER simulator has been designed aiming to test the Anti-replay tool behavior against real-time spoofing attacks. The simulator is able to generate the signal Galileo E1B, including the OSNMA service. It aims to simulate the scenario where a zero-delay SCER attack is being conducted. To do this, it generates two perfectly aligned signals simultaneously: one reproduces the behavior of the true signals coming from satellites, and the other reproduces the signal from the spoofer that tries to accomplish the attack. Like in a real zero-delay attack, signals have to be aligned and, usually, the power of the spoofed signal will be higher than the real one. At the beginning of the scenario, only the real signal is been transmitted. Later, after a configurable time, the spoofing signal starts. The scenario will follow three steps to accomplish the attack:

- Initial phase: only the simulated signal of the satellites is transmitted. In this initial phase there is no spoofer trying to carry out an attack, which means that the Anti-replay method output should report “no spoofing”, and any different output is considered a false alarm.
- Attack phase: the spoofed signal invades the scenario with a higher power than the real signal. This signal is perfectly aligned with the real one, simulating a theoretical zero-delay attack. Since this moment the Anti-replay method should

inform that the receiver is being spoofed. The detection time is calculated from the beginning of this phase to the first "spoofing" message sent by the Anti-replay technique.

- Forged phase: after achieving to force the receiver to loss the lock on the real signal, the spoofer starts modifying the position of the receiver. If the Anti-replay does not inform the receiver that it is being spoofed, then it will calculate the spoofed PVT, which means that the attack is being reached.

Figure 7 shows the development of these three steps. The gray line represents the real signal coming from the satellites, while the red line represents the spoofed signal.

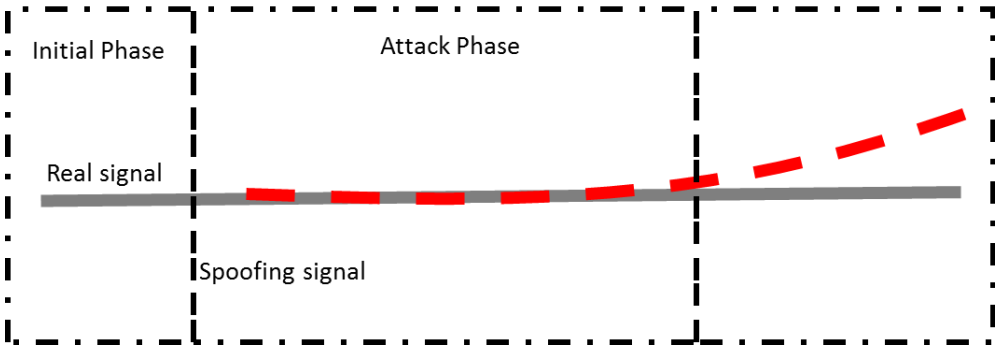

Figure 7 – Attack simulation

The SCER simulator consists of two elements: the software module and a software-defined radio (SDR) platform. The software module generates Galileo E1 signal, and has the capacity of generating two signals at the same time. The parameters of both signals can be configured in an XML file. The module writes the signal in a binary output file with a configurable sampling frequency.

The E1-B data bits containing the OSNMA are generated by the CSDemo platform [9]. Furthermore, the simulator needs a RINEX file version 3 [10] of the simulation day. It is important that the RINEX contains information about the satellites of the scenario, to estimate their orbits.

Figure 8 – SCER simulator

For each signal, the position can be configured as static or dynamic. For a dynamic behavior of the position, a CVS file is needed, and for a static behavior only WGS84 coordinates are needed. Usually, the signal coming from the transmitter has higher power than the real signal, with the purpose of being the signal tracked by the receiver. That different of power between both signals is also configurable as a parameter.

As studied in some previous references [5] [2] [11] [3], in a zero-delay attack the spoofed signal needs some time to estimate the value of each unpredictable symbol. This time varies depending on several factors, but the main one is the state of the input signal coming from satellites. In particular, C/N_0 has a direct influence on the probability of the successful estimation of the symbol. Every m set of samples has a probability of error in the estimation ($p_{err}(m)$). This probability depends on the sampling period T_s, the number of samples integrated by the spoofer m, and the carrier to noise plus interference density ratio as seen by the spoofer ($C/(N_0 + I_0)$):

$$p_{err}(m) = \frac{1}{2}\, erfc\left(\sqrt{m\, T_s\, C/(N_0 + I_0)}\right) \tag{14}$$

The number of sets of samples multiplied by the sampling period is the duration time interval used by the spoofer to estimate the symbol. The SCER simulator includes this error probability in the chips of the spoofer signal and depending on the C/N_0 of the real signal by performing a logarithm approximation of the (14) formula. Principally, this Carrier-to-noise density ratio depends on the distance between the spoofer and the satellite, because of the free space losses and the atmospheric absorption. However, an advanced spoofer may compensate these losses with additional hardware, as for example directional antennas, and higher values need also to be considered.

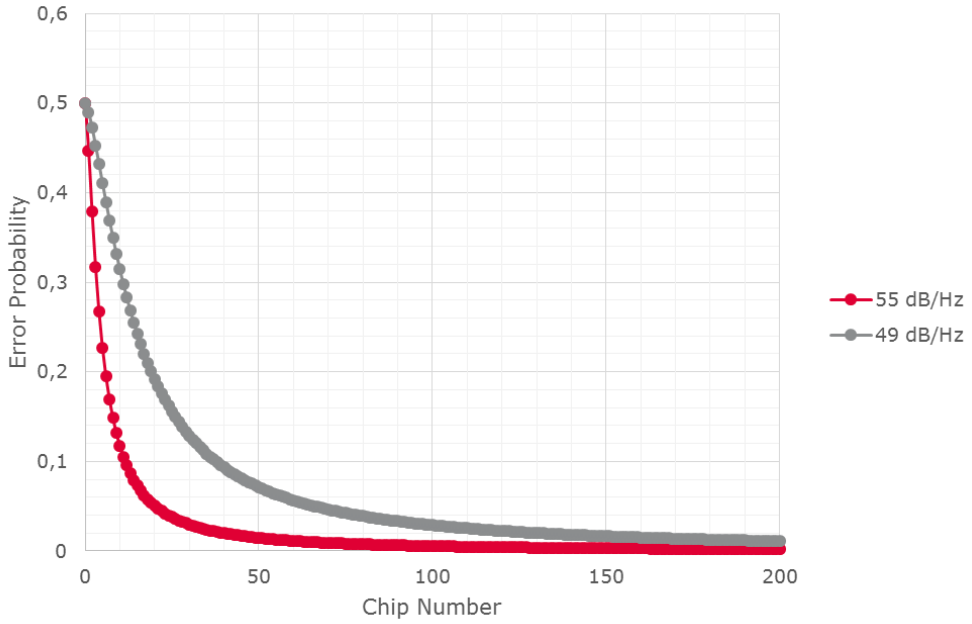

Figure 9 – Error probability function

Figure 6 shows an example of the behavior of the error probability of the chips of the spoofed signal for two different C/N_0 conditions observed by the spoofer. Only the first 200 chips are shown as after, the probability is very low, just to show the impact of different signal conditions over the imperfect estimation efficiency. Both C/N_0 values of the figure are illustrative, they are the maximum and the minimum of the satellites included the test scenario that is going to be presented in the following sections.

TEST SCENARIO AND CONFIGURATION

A dedicated testing campaign to validate both the SCER attack implementation and the effectiveness of the protection technique has been conducted. The first step was to identify an independent COTS receiver to compare the results with respect to the NACSET User Terminal. The selected receiver is a u-Blox M8T which supports Galileo E1 and GPS L1 amongst other constellations.

An Open Sky scenario has been defined to test the anti-replay technique with both detection metrics previously explained. The signal conditions and satellite availability observed by both the user and the spoofer can be seen in Figure 10.

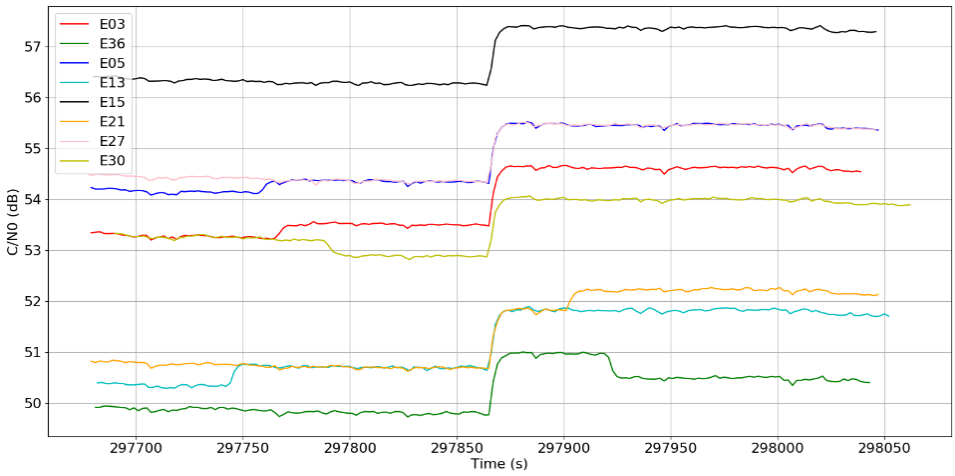

Figure 10 - C/N_0 values during the scenario

Figure 10 shows the C/N_0 development during the scenario, including the difference of power at the beginning of the attack. The scenario begins at the Time of Week (TOW) 297626 seconds. After 240 seconds the attack phase starts and the spoofing transmits its signal. The spoofed signal has 1 dB/Hz more of power than the real signal. The spoofed signal has error chips at the beginning of each unpredictable symbol. These errors are random, but depend on the error probability function explained previously. Table 1 shows the configuration of the detection technique, 32 symbols are accumulated for the partial correlations of the first (and last) 50 chips and the results are analyzed over a sliding window of 20 seconds.

Table 1 – Technique configuration

Parameter	Value
Unpredictable symbols	32
Chips to be correlated	1600 (50 per symbol)
Sliding window	20s

For the test setup, the UT is used to test the performance of the implemented anti-replay technique and the COTS receiver is used to test how a mass-market receiver without any anti-replay protection behaves receiving the signals generated by the SCER simulator. The selected COTS receiver is a U-Blox M8T which supports Galileo E1 and GPS L1 amongst other constellations. A HackRF One has been used as a SDR to radiate the signal both to the UT and to the COTS receiver.

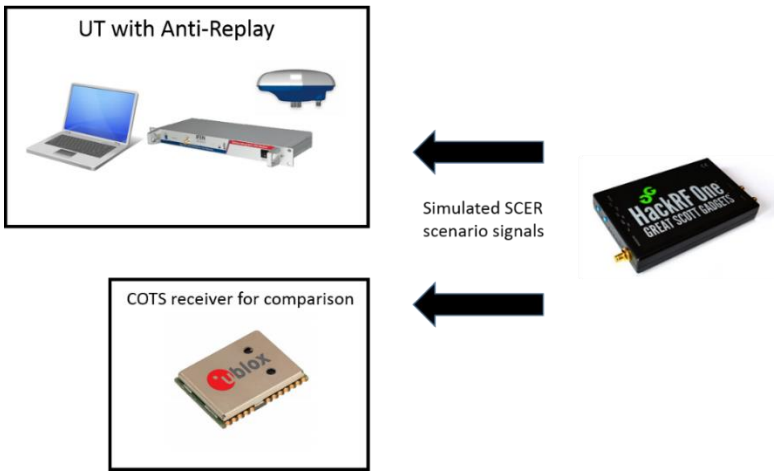

Figure 11 Test set-up

RESULTS

A dedicated testing campaign to validate both the SCER attack implementation and the effectiveness of the protection technique has been conducted. First of all we can see the impact of the SCER attack simulated on the COTS receiver. When the spoofed signal starts the graphical interface shows an increase in the power of the satellites which are been tracked, concretely 1 dBHz more, as configured in the SCER simulator. Furthermore, the PVT previously computed by the receiver starts deviating in a linear way which indicates that the attack has been successful.

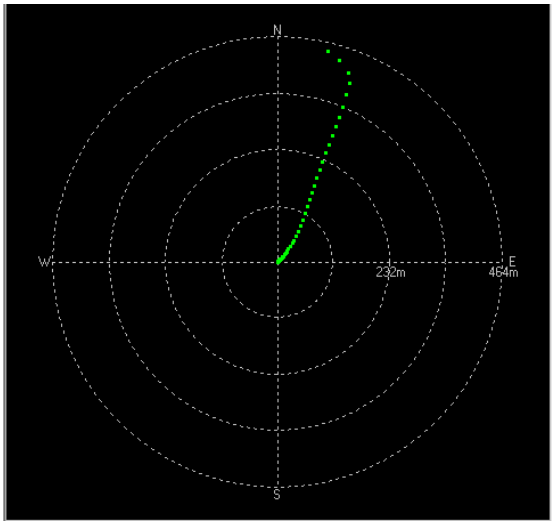

Figure 12 – U-Blox GUI computed position

Now, we will present the same scenario broadcasted to the UT to assess the performance of the two anti-replay metrics. Figure 13 depicts an example of the errors in the estimation of the symbol sign observed at the beginning of the symbol, for some random symbols, in order to illustrate the behavior of the simulated attacker. Figure 13 displays the first 1000 chips of an unpredictable symbol for several satellites. Green means that the chip has been estimated correctly by the spoofer, and red that the estimation has been unsuccessful. As the error probability function depends on the power of each satellite, the lower the C/N_0, the higher the error probability. Hence, the satellites received with less power, are sent by the spoofer with more errors in the unpredictable chips.

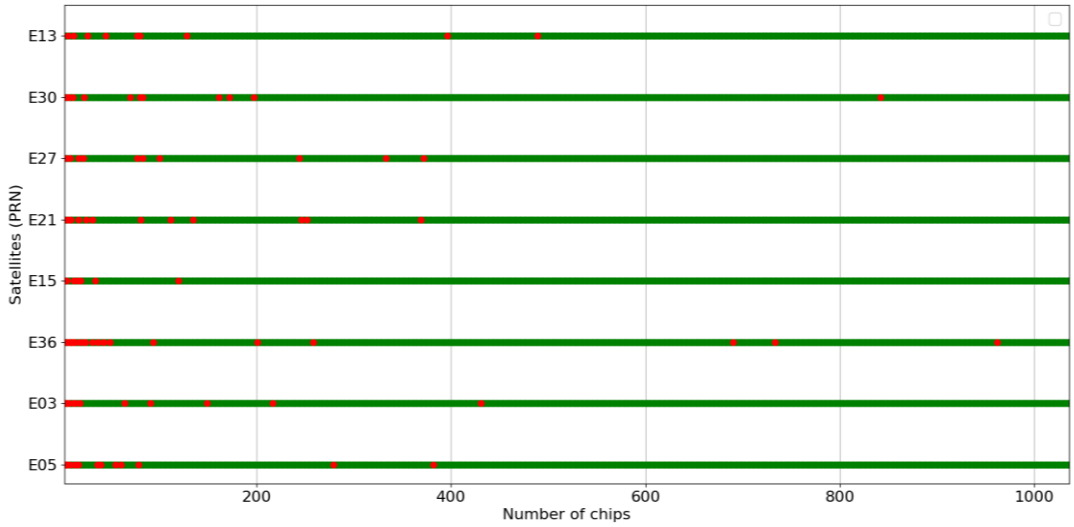

Figure 13 – First chips of an unpredictable symbol

Table 2 shows the percentage of errors of all the chips of each unpredictable symbol, which, as expected, is very low, if the totality of chips are taken into account. We can see numerically the expected behavior of the spoofer:

Table 2 – Error chip percentage in unpredictable symbols

PRN	% chip errors	PRN	% chip errors
E03	0.30	**E21**	0.38
E05	0.27	**E27**	0.27
E13	0.39	**E30**	0.32
E15	0.18	**E36**	0.42

The scenario starts at the second 297626. After 240 seconds the spoofer starts to broadcast the estimated signal. From this second, a jump should be detectable and the Anti-replay should notice that the attacker is trying to spoof the receiver.

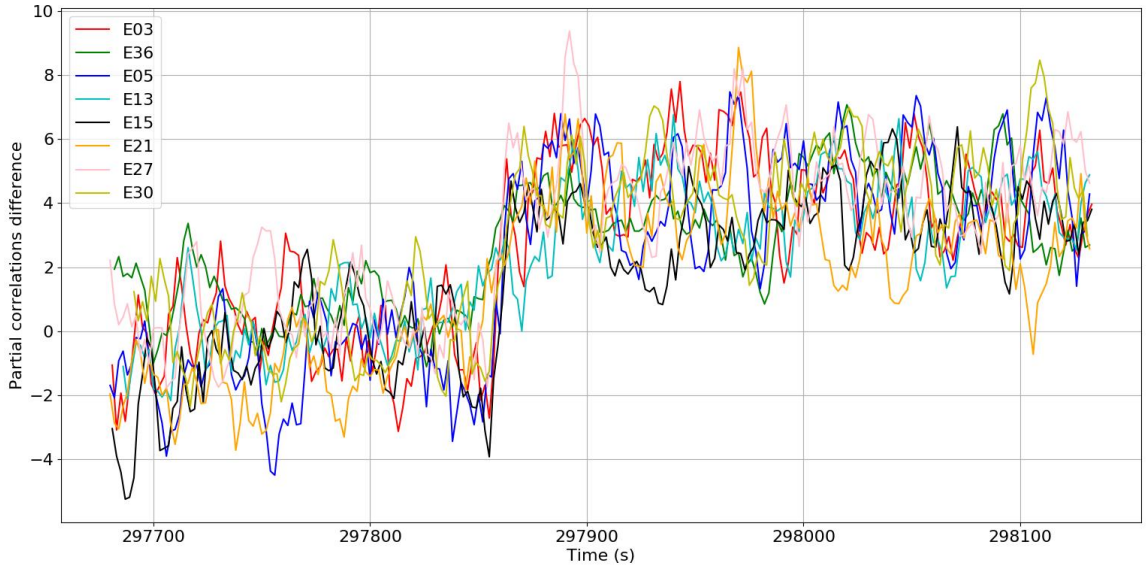

Figure 14 – Partial correlations differences

Figure 9 shows us the difference between the partial correlation results for each satellite. The values are the results of the accumulation of 32 symbols, 2 seconds. Since the beginning of the scenario and until 240 seconds later the mean of the correlation values is around 0. After this moment there is a notorious jump in all the satellites at the same time. It makes sense to conclude that the Anti-replay technique can detect the spoofing attack because of the jump. But as said above, the Anti-replay method does not compare these values with the threshold, it compares the mean of the sliding window of the last values:

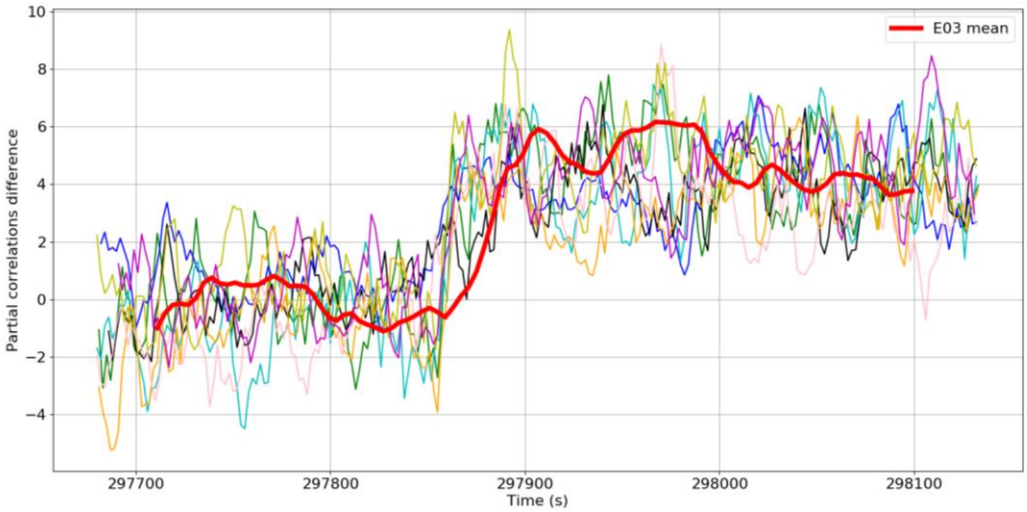

Figure 15 – Partial correlation differences with E03 mean

Figure 15 shows the mean of the sliding window of the satellite E03 in a red line as an example. The jump is as high as it was with non-averaged values, but less noisy. Equivalent to this, the same process is done for all the satellites and the attack is always detected with the floating, standard deviation-dependent threshold defined in (11) and (12).

Table 3 – Detection time of the partial correlations difference method

PRN	TTA (s)	PRN	TTA(s)
E03	25	**E21**	24
E05	26	**E27**	20
E13	20	**E30**	21
E15	29	**E36**	19

Table 4 shows the detection time of each satellite by the Anti-replay solution using the partial correlation analysis metric. The detection time is calculated since the beginning of the attack phase. With this method, the Anti-replay technique can detect the attack after, as a maximum, 29 seconds.

The same experimentation test has been done with the other Anti-replay method, the C/N_0 method. With the help of the NWPR estimator, the technique compares the C/N_0 of the beginning and the end of each unpredictable symbols.

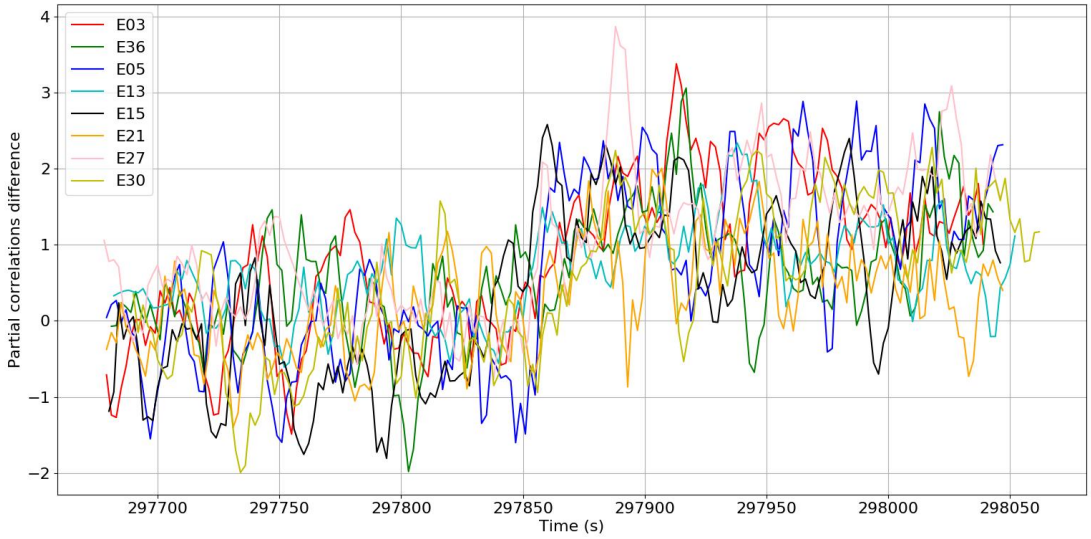

Figure 16 - C/N_0 *estimations*

Analogous to the previous case Figure 16 shows the results of the C/N_0 method of the Anti-replay technique. In this case, the noise of the C/N_0 estimations is bigger than the partial correlation but still the technique is able to detect the attack. For all the satellites an approximate increase of 1.5 dB/Hz is observed.

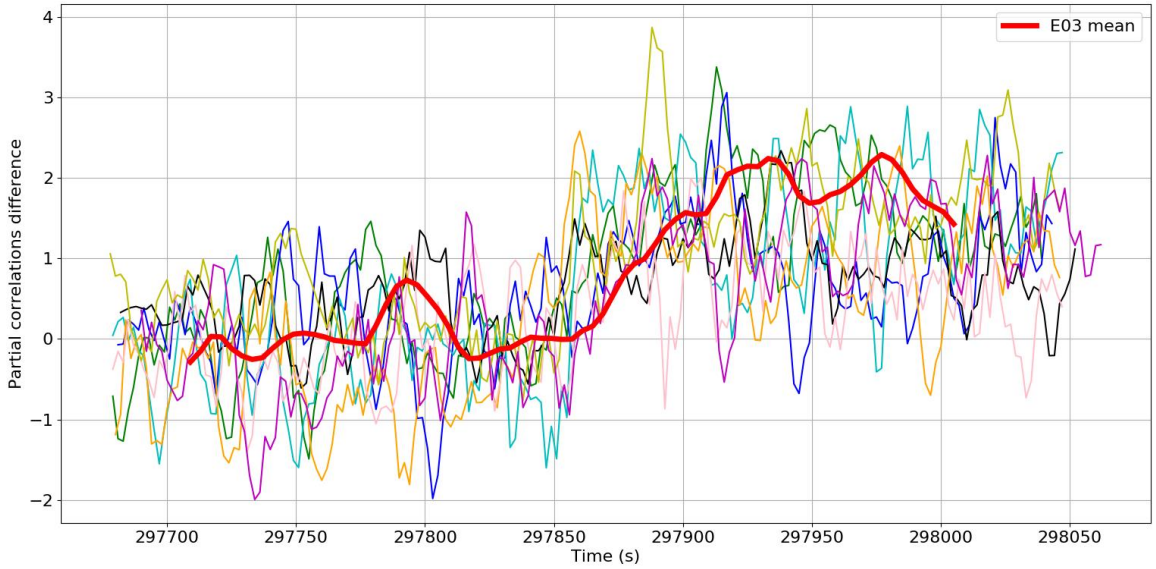

Figure 17 - C/N_0 *estimations with E03 mean*

As in the previous method, the red line of Figure 17 shows us the mean correlation of the sliding window. The method is able to detect the spoofed signal in all the satellites, but for some satellites the time to detect is greater, especially for satellites with noisier C/N_0 estimations.

Table 4 - Detection time of the C/N_0 method

PRN	TTA (s)	PRN	TTA(s)
E03	21	**E21**	29
E05	23	**E27**	18
E13	20	**E30**	31
E15	21	**E36**	38

Table 4 shows that all the satellites have been detected within a few tens of seconds, during which an attacker has several restrictions in altering the receive position significantly while remaining unnoticed. Satellites with a high C/N_0 are easier to detect even if the spoofer produce worse symbol estimations. On the other hand, those with less power, such as E30 or E36, make the Anti-replay to spend more time in their detection.

CONCLUSIONS

This paper has presented a first implementation of anti-replay protection techniques based on symbol unpredictability. It has been carried out with samples from a hardware receiver in real time. Additionally, a zero-delay SCER attack simulator have been designed and developed to validate the technique implementation.

A demonstration on anti-replay capabilities of OSNMA in an Open Sky simulation has been conducted showing that the impact of an imperfect estimation of the beginning of the unpredictable symbols by a spoofer can be observed and used for spoofing detection. Even with few unpredictable symbols (32 symbols per page, for a sliding window of 20 pages), the attack can be detected for all satellites. The partial correlations difference detector performs slightly better than C/N0 estimations but they are both promising. Tests accumulating different amounts of symbols and using other detectors shall be assessed as well.

Further work includes testing the techniques simulating harsher environments (urban, suburban, …) both for the user and the spoofer and with different spoofing capabilities. Also, other techniques taking into account different amounts of symbols and more refined detectors beyond the simple statistic here proposed, must be evaluated in the future.

This work represents one of the first attempts to test GNSS spoofing zero-delay attacks and receiver defenses with real receivers and detectors. Even if further analyses and tuning are necessary, the results show that symbol unpredictability at data level can be an effective feature to detect replayed signals for cases when the receiver is already tracking authentic signals.

REFERENCES

[1] The European Union, *Commercial Service Demonstrator website http://www.galileo-csdemo.eu/.*

[2] I. Fernández-Hernández y G. Seco-Granados, «Galileo NMA Signal Unpredictability and Anti-Replay Protection,» *ICL-GNSS,* 2016.

[3] G. Seco-Granados, D. Gomez-Casco y I. Fernández-Hernández, «Detection of Replay Attacks to GNSS based on Partial Correlations and Authentication Data Unpredictability,» *in preparation for GPS Solutions.*

[4] G. Caparra, S. Ceccato, N. Laurenti y J. Cramer, «Feasibility and Limitations of Self-Spoofing Attacks on GNSS Signals with Message Authentication,» *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017),* pp. 3968-3984, 2017.

[5] T. E. Humphreys, «Detection strategy for cryptographic gnss antispoofing,» *Aerospace and Electronic Systems, IEEE Transactions,* vol. 49, nº 2, pp. 1073-1090, 2013.

[6] G. Caparra, N. Laurenti, T. I. Rigas y C. Massimo, «Improving secure code estimate-replay attacks and their detection on gnss signals,» *Proceeding of NAVITEC,* 2014.

[7] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simón, I. Rodríguez y J. D. Calle, «A navigation message authentication proporsal for the galileo open service,» *Navigation,* vol. 63, nº 1, pp. 85-102, 2016.

[8] The European Union, «European GNSS (Galileo) Open Service Signal In Space Interface Control Document,» 2015. [En línea].

[9] C. Sarto, O. Pozzobon, S. Fantinato, S. Montagner, I. Fernández-Hernández, J. Simón, S. Cancela, J. D. Calle, P. Walker, D. Burkey, G. Seco-Granados y E. Göhler, «Implementation and testing of OSNMA for Galileo,» *Proccedings of the ION GNSS+ Meeting*, 2017.

[10] Astronomical Institute of the University of Bern; UNVACO, The Receiver Independent Exchange Format Version 3.03, International GNSS Service (IGS), RINEX Working Group and Radio Technical Commission for Maritime Services Special Committee 104 (RTCM-SC104), 2015.

[11] S. Cancela, J. Navarro, J. D. Calle, E. Göhler, A. Dalla Chiara, G. Da Broi, I. Fernández-Hernández, J. Simón y G. Seco-Granados, «Designing and evaluating next generation of resilience receivers,» *International Technical Symposium on Navigation and Timing (ITSNT) 2018*, 2018.

[12] M. L. Psiaki y T. E. Humphreys, «GNSS Spoofing and Detection,» *Proceedings of the IEEE*, vol. 104, nº 6, 2016.

[13] A. Perrig, R. Canetti, J. D. Tygar y D. Song, «The TESLA Broadcast Authentication Protocol,» *CryptoBytes*, 2002.

[14] J. T. Curran y C. O'Driscoll, «Message Authentication, Channel Coding & Anti-Spoong,» *Institute of Navigation*, 2016.

[15] S. Cancela, J. D. Calle, G. Arroyo, A. Dalla Chiara, G. Da Boir, O. Pozzobon, C. Sarto, J. Winkle, I. Krol, P. Webster, I. Fernández-Hernández, J. Simón y G. Seco-Granados, «Designing and evaluation next generation of resilience receivers,» *Institute of Navigation*, 2017.

[16] K. Wesson, M. Rothlisberger y T. E. Humphreys, «Practical crytographic civil GPS signal authentication,» *Navigation, Journal of the Institute of Navigation*, 2011.

[17] T. Ebinuma, *gps-sdr-sim https://github.com/osqzss/gps-sdr-sim*.