Universitat Oberta de Catalunya

uoc.edu

Técnicas de autenticación criptográfica en señales abiertas GNSS. El fenómeno del spoofing y métodos de defensa.

José Luis Iglesias Fernández
Máster Universitario en Ingeniería de Telecomunicación
Tecnologías de la radiocomunicación

Gonzalo Seco Granados
Germán Cobo Rodríguez

17 de junio de 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Técnicas de autenticación criptográfica en señales abiertas GNSS. El fenómeno del spoofing y métodos de defensa.*
Nombre del autor:	*José Luis Iglesias Fernández*
Nombre del consultor/a:	*Gonzalo Seco Granados*
Nombre del PRA:	*Germán Cobo Rodríguez*
Fecha de entrega (mm/aaaa):	*06/2019*
Titulación::	*Máster Universitario en Ingeniería de Telecomunicación*
Área del Trabajo Final:	*Sistemas de radionavegación*
Idioma del trabajo:	*Castellano*
Palabras clave	*Spoofing, Autenticación, OSNMA.*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Los sistemas GNSS son un referente en cuanto a tecnologías de posicionamiento y temporización. A medida que crecen los ámbitos de aplicación de este tipo de sistemas, también evolucionan las amenazas a su seguridad. La más importante a corto y medio plazo es la que afecta a la integridad de sus transmisiones mediante ataques de spoofing. Consisten en emitir una señal GNSS falseada para proporcionar unos datos de posicionamiento falsos al usuario.

La creciente relevancia que está adquiriendo la seguridad de la información implica también a los sistemas GNSS. En este Trabajo se tratan diversos métodos de ataque de spoofing GNSS. También se detallan métodos de defensa basados tanto en la detección de ataques como en la mejora de las capacidades de resistencia a los mismos, algunos de ellos basados en la encriptación de los sistemas de comunicación.

En este Trabajo se detallan diversos métodos de autenticación implementados sobre señales GNSS de servicio abierto al público o uso civil, cuya transmisión no está basada en la encriptación. Galileo utiliza OSNMA para encriptar los mensajes de navegación de sus servicios Open-Service. GPS ha propuesto el esquema Chimera, que basa la autenticación de sus señales transmitidas en abierto en enlazar el mensaje de navegación y el código de ensanchamiento. QZSS ofrece un sistema para autenticar los mensajes de navegación GPS. Se ha realizado la implementación en Matlab de una prueba de concepto simplificada del funcionamiento de OSNMA. Finalmente se realiza una

comparativa entre las prestaciones y aplicaciones prácticas de OSNMA y Chimera.

Abstract (in English, 250 words or less):

GNSS are a very important source of information in terms of location and timing techonologies. Everyday new systems and devices get connected via these systems. Of course, as the number of devices connected via GNSS grows, so does the security concerns related to the emerging threats. The most important threat in short and mid term are spoofing attacks. These aim to deceive the receiver with a fake signal that pretends to be authentic, so the user gets located with an unreal position.

Security information is one of the basis of IT systems and the number and complexity of attacks to GNSS keep growing. Different kinds of spoofing attacks are discussed on this paper. Related to this, defense methods to face this sort of attacks are also displayed. Advanced methods of defence are cryptography based and involve the entire system.

On this paper different authentication methods used in GNSS are discussed, focusing on those designed for open civil signals and plaintext communications. Galileo uses ONSMA to encrypt and authenticate navegation messages used in Open-Service applications. GPS proposal scheme Chimera involves time-binding and spreading code encryption. QZSS aims to authenticate GPS signals by using L1SAIF signal. A proof-of-concept has been designed and developed to analyze OSNMA authentication method, based on TESLA algorithm and used in Galileo system. A benchmark is displayed to discuss Galileo OSNMA and GPS Chimera main features, disadvantages and requirements. In the last chapters of this paper, conclusions and future work can be found.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

Los sistemas de radionavegación son sistemas de posicionamiento que estiman la posición de un blanco o usuario mediante la utilización de señales radio. Aquellos que se basan en la emisión continua de señales a través de una constelación de satélites se denominan sistemas de radionavegación por satélite, obteniendo la ventaja de aumentar la cobertura con respecto a los sistemas basados únicamente en estaciones terrestres. Si el objetivo es conseguir una cobertura global basada en la utilización de satélites, se trata de un sistema global de navegación por satélite, o Global Navegation Satellite System (GNSS).

El número de dispositivos conectados ha aumentado exponencialmente en los últimos años. Desde el siglo XX, las necesidades de conexión y de posicionamiento de estos dispositivos han aumentado, del mismo modo que las capacidades tecnológicas de comunicación han acompañado este incremento. El abaratamiento de costes y la miniaturización de la tecnología han permitido ofrecer servicios de posicionamiento por satélite a una variedad de dispositivos cada vez mayor, pasando de cubrir únicamente aplicaciones militares y medios de transporte colectivos a la actual oferta de servicios de localización, como los de búsqueda y rescate, servicios comerciales o servicios de uso abierto.

En la actualidad, la información ocupa el puesto más alto en las auditorías de análisis y gestión de riesgos y activos. La gestión de la información se fundamenta en la confidencialidad, la integridad y la disponibilidad de la misma. El tratamiento de la información de forma inadecuada puede conllevar desde la bancarrota de una empresa multinacional puntera hasta un incidente geopolítico de grandes magnitudes. No es de extrañar pues que la seguridad de la información sea uno de los pilares del desarrollo tecnológico.

Para ofrecer diversos servicios de localización, los sistemas GNSS comparten información con sus usuarios, quienes la utilizan para obtener su posición, velocidad o como una fuente fiable para un sistema de tiempos. Ante la posibilidad de que estas transmisiones se vean comprometidas y también los servicios que dependan de ellas, los sistemas GNSS están implementando sistemas de autenticación de mensajes que permitan a los usuarios asegurar que la información que obtienen de dichas señales procede de una fuente fiable y que no han sido comprometidas.

En este Trabajo de Fin de Máster se analizan las implementaciones de sistemas de autenticación que se han desarrollado previamente y se están desarrollando en la actualidad para los diversos sistemas GNSS disponibles. También se realizará una implementación simplificada del procedimiento de generación de claves y códigos utilizado en el sistema de autenticación Open-Service Navigation Message Authentication (OSNMA), propuesto para Galileo y de próxima implementación, de forma que se pueda comprobar su funcionamiento.

1.2 Objetivos del Trabajo

Resumen del problema de spoofing en GNSS y soluciones de autenticación.

- Realizar una revisión del problema de spoofing en GNSS mediante la definición del fenómeno, formas de llevarlo a cabo mediante el ataque a sistemas y métodos de defensa para contrarrestarlo.
- Describir cómo la autenticación del mensaje de navegación puede combatir algunos tipos de ataques de spoofing, y comprender las limitaciones o imposibilidad que la autenticación también tiene para combatir otros ataques.
- Recopilar y presentar de forma ordenada los diferentes esquemas de autenticación propuestos para los sistemas GNSS.
- Estudiar las soluciones de autenticación de Galileo, GPS y QZSS. El énfasis del trabajo será en el sistema Galileo, mientras que el resto se abordarán con menor detalle.

Implementación de un caso simplificado de Galileo OSNMA

- Delimitar de forma muy precisa lo que se pretende implementar. No es necesario que el desarrollo sea parametrizable, puede un valor concreto de los muchos parámetros que hay en OSNMA. La idea general es generar una cadena de claves utilizando uno de algoritmos de hashing de la especificación OSNMA y combinarla con mensajes en texto plano para generar HMACs. Se comprobará la correcta detección en el receptor en el caso de que existan errores en la transmisión que modifiquen alguno de estos elementos.
- Diseñar los principales bloques del software a implementar.
- Desarrollar el software.
- Comprobar su funcionamiento.

1.3 Enfoque y método seguido

Se opta por estudiar las diferentes variedades de ataque a sistemas GNSS, indagando en las diferencias entre jamming y spoofing. Será este último apartado el que se desarrolla más a fondo, centrando la investigación entre las diferentes formas de ataque disponibles y los métodos de defensa ante ellos.

Se estudian los sistemas de autenticación propuestos para sistemas GNSS. Como primera aproximación, se opta por el algoritmo TESLA de autenticación en el que se basa la solución para Galileo. Después se detallan las modificaciones que se han realizado sobre este algoritmo para obtener la definición teórica de OSNMA, así como las especificaciones y características técnicas de la implementación del sistema. Se estudian las implementaciones de sistemas de autenticación desarrollados para los sistemas GPS y QZSS.
Se realiza una comparativa de los sistemas de autenticación para Galileo y GPS, teniendo en cuenta sus requerimientos y funcionalidades.

Se desarrolla una prueba de concepto del sistema OSNMA para Galileo en el que se busca comprobar el correcto funcionamiento del sistema de autenticación.

1.4 Planificación del Trabajo

Se detalla a continuación la planificación de tareas a realizar a lo largo del Trabajo de Fin de Máster. Se dividen los recursos en función del tiempo que se espera destinar a los distintos apartados. Se analiza su extensión y dificultad esperadas.

Se detallan la duración esperada para los entregables o PEC según la planificación realizada.

	Nombre	Duracion	Inicio	Terminado
	Introducción	14 days	18/02/19 8:00	7/03/19 17:00
	Spoofing y Defensa	15 days	12/03/19 16:00	2/04/19 16:00
	NMA. OSNMA	30 days	28/03/19 8:00	8/05/19 17:00
	NMA. QZSS	10 days	9/05/19 8:00	22/05/19 17:00
	Spreading. Chimera	15 days	20/05/19 8:00	7/06/19 17:00
	Comparativa OSNM	5 days	6/06/19 9:00	13/06/19 9:00
	Conclusiones	4 days	10/06/19 8:00	13/06/19 17:00
	Diseño e implement	30 days	29/04/19 8:00	7/06/19 17:00
	PEC1	13 days	21/02/19 8:00	11/03/19 17:00
	PEC2	20 days	12/03/19 8:00	8/04/19 17:00
	PEC3	25 days	9/04/19 7:00	13/05/19 17:00
	PEC4	25 days	14/05/19 7:00	17/06/19 17:00

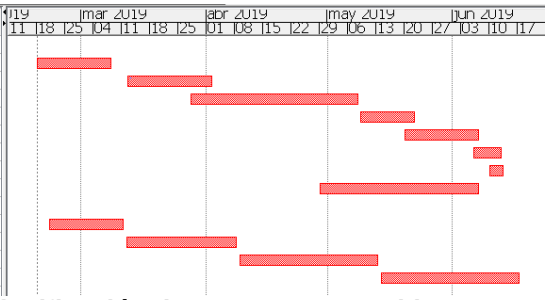

Ilustración 1. Diagrama de Gantt de planificación de tareas y entregables.

1.5 Breve sumario de productos obtenidos

- Enumeración de métodos de ataque mediante spoofing enfocados a sistemas GNSS, en relación con los métodos de defensa ante ellos, incluyendo ventajas e inconvenientes de los mismos.

- Características principales de los sistemas de autenticación para señales abiertas en GNSS.

- Análisis en profundidad del sistema de autenticación de mensaje de navegación OSNMA implementado sobre el sistema Galileo. Detalle de funcionamiento del algoritmo TESLA y particularización del mismo para implementar el sistema de autenticación OSNMA.

- Análisis del sistema implementado en QZSS para autenticación de mensajes de navegación GPS.

- Análisis en profundidad del sistema autenticación por encriptación del código de ensanchamiento denominado Chimera, planteado para ser implementado sobre el sistema GPS.

- Diseño e implementación de prueba de concepto software para comprobar el funcionamiento de sistema de generación de la cadena de claves utilizado en OSNMA y basado en el algoritmo TESLA.

- Comparativa de OSNMA de Galileo y Chimera de GPS, poniendo énfasis en las principales funcionalidades ofrecidas y sus limitaciones, relacionados con los requerimientos necesarios para su implementación.

- Conclusiones extraídas a lo largo del desarrollo del Trabajo de Fin de Máster, así como las posibles líneas de trabajo futuras a desarrollar.

1.6 Breve descripción de los otros capítulos de la memoria

Spoofing.

Introducción a los métodos de ataque a los sistemas GNSS basados en la suplantación de señales reales. Métodos de defensa ante este tipo de ataques a implementar en receptores y sistemas GNSS.

NMA. Técnicas de autenticación por encriptación de mensaje de navegación.

Características generales de los métodos de autenticación mediante clave simétrica y asimétrica. Implementación de OSNMA sobre Galileo y características principales. Estructura de mensajes de navegación y autenticación en los que se basará el funcionamiento del sistema.
Análisis del sistema de autenticación japonés QZSS para autenticación de señales GPS, entre otras.

Técnicas de autenticación por encriptación de código de ensanchamiento.

Características generales de este tipo de sistemas de autenticación. Funcionalidades principales y requerimientos. Implementación de Chimera sobre GPS, principales características y funcionalidades, así como requerimientos hardware y software para su implementación.

Implementación de una prueba de concepto de Galileo OSNMA.

Diseño e implementación de una prueba de concepto simplificada para comprobar el funcionamiento del algoritmo TESLA modificado que implementa el sistema de autenticación OSNMA, de Galileo.

Comparativa entre OSNMA de Galileo y Chimera de GPS.

Principales características, ventajas y funcionalidades que ofrecen ambos sistemas, así como sus requerimientos y limitaciones.

Conclusiones.

Principales objetivos conseguidos en el Trabajo a través de sus diferentes apartados. Líneas de trabajo futuro que podrían seguirse.

2. Spoofing

El *spoofing* de señales GNSS se basa en la transmisión de señales falsas con la intención de que el receptor de la víctima las confunda con señales auténticas, obteniendo un posicionamiento erróneo. Un sistema que busca suplantar las señales auténticas GNSS enviadas desde un satélite es un *spoofer*.

Los sistemas utilizados en este tipo de prácticas no siempre son diseñados únicamente con el fin de realizar *spoofing*. Debido principalmente a la miniaturización de dispositivos, el aumento de la capacidad de procesamiento y a la disminución de costes de dispositivos, es posible desarrollar simuladores de señal programables a precios muy bajos, utilizando como software almacenado en repositorios online. Los objetivos de estos ataques son tanto militares como civiles, siendo estos últimos más vulnerables debido a las características de las señales que utilizan. En este capítulo se van a analizar los diferentes tipos de ataques y métodos defensas ante ellos. Se hará especial énfasis en las diversas técnicas de autenticación que permiten aumentar la robustez ante estos ataques y que serán el principal objeto de análisis de este Trabajo de Fin de Máster.

Métodos de ataque mediante *spoofing*

Descripción de un ataque de *spoofing*.

Un *spoofer* envía un conjunto de señales falsas que son similares a las auténticas. Para conseguir engañar al receptor de la víctima, debe replicar las señales generadas en un satélite GNSS destinadas a facilitar el posicionamiento del receptor. Esto incluye la portadora RF, el código PseudoRandom Noise (PRN) y los bits de datos de cada señal *Open-Service* GNSS auténtica.

Spoofer Auto-consistentes.

Sus ataques buscan evitar la estrategia de defensa de monitorización mediante Receiver Autonomous Integrity Monitoring (RAIM), que estudia los residuos de los pseudorrangos. Esto se consigue mediante la generación de falsos códigos de fase para inducir un falso posicionamiento o temporización. Los cálculos necesarios para ello son sencillos. Para que el receptor víctima no detecte la suplantación de la señal, la fase de portadora y la fase de código de la señal falsa han de variar en consonancia, evitando así que pueda detectarse el ataque por una divergencia inusual entre portadora y código.

Un *spoofer* busca inducir al receptor de la víctima para que se enganche a las señales falsas no procedentes de un satélite GNSS. Existen dos formas de conseguir este efecto.

- Una es mediante el efecto de interferencia intencionada *jamming*, interrumpiendo el seguimiento de la señal auténtica para que se produzca una re-adquisición de la señal. En ese momento, el receptor busca

engancharse de nuevo a una señal. Si las señales falsas tienen una amplitud significativamente mayor que las señales auténticas, entonces obtendremos una alta probabilidad de que se enganche a las señales falsas durante la fase de readquisición.

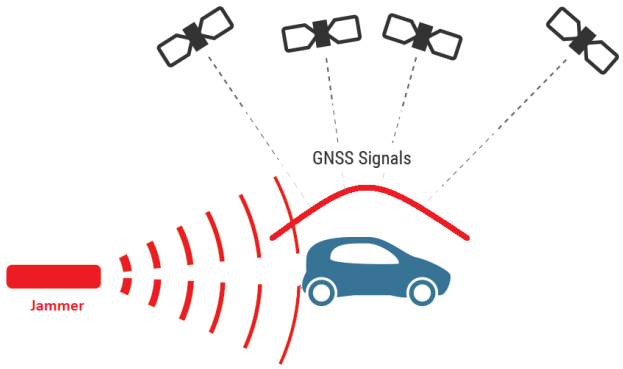

Ilustración 2. Técnica de ataque mediante jamming. Imagen obtenida de [1].

- Otro método es la transmisión de señales falsas que puedan ser confundidas con las reales. Para ello, las señales auténticas y las falsas deben tener los mismos parámetros en cuanto a fase de código y portadora en la localización de la antena del receptor, evitando así las consecuencias negativas del Efecto Doppler. El *spoofer* emite una señal de baja potencia, aumentándola gradualmente hasta que consigue arrastrar el bucle de seguimiento. Finalmente, el *spoofer* consigue arrastrar la fase de código y de portadora. Puesto que es necesario conocer los valores de amplitud y los valores de fase de código de las señales auténticas, el *spoofer* ha de ser también receptor. Además, deberá conocer su relación geométrica con la víctima. Al no necesitar de *jamming* y etapa de re-adquisición, este método de ataque tiene mayor probabilidad de no ser detectado. Las técnicas de RAIM no son capaces de detectarlo.

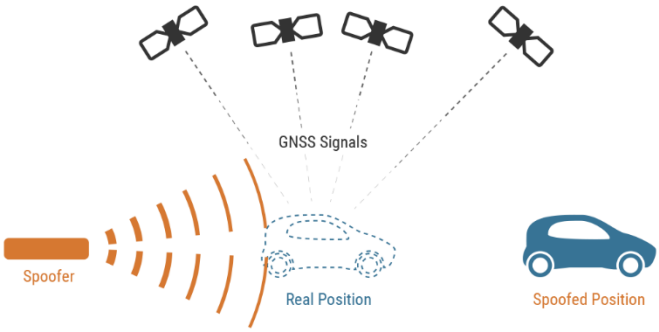

Ilustración 3. Técnica de ataque mediante spoofing. Imagen obtenida de [1].

Desde el punto de vista del atacante, para evitar que el ataque de spoofing pueda ser detectado por la víctima es importante que durante las primeras fases del ataque la amplitud de la señal falsa se mantenga a un nivel menor que la amplitud de la señal auténtica, y que la fase de portadora no varíe bruscamente [2]. Una vez que la fase de código falsa está suficientemente lejos de la auténtica, se puede comenzar a variar la fase de portadora, siempre que se haga de forma

que se mantenga la relación entre las fases de código y portadora. De esta forma, el ataque de *spoofing* habrá conseguido arrastrar el bucle de seguimiento.

Los *spoofers* descritos en el Trabajo hasta este punto deben recrear el código de ensanchamiento $C_i(t)$ y el flujo de bits de datos transmitido $D_i(t)$. Esto es una tarea sencilla si ambos son perfectamente predecibles, como en el caso de las aplicaciones civiles GNSS. Sin embargo, si uno de ellos no lo es, como es el caso de las señales militares con el antiguo código P(Y) o el nuevo código M, el *spoofer* deberá generar réplicas aproximadas al vuelo. Otra forma de obtener señales con seguridad añadida es hacer impredecibles algunos segmentos cortos de $C_i(t)$.

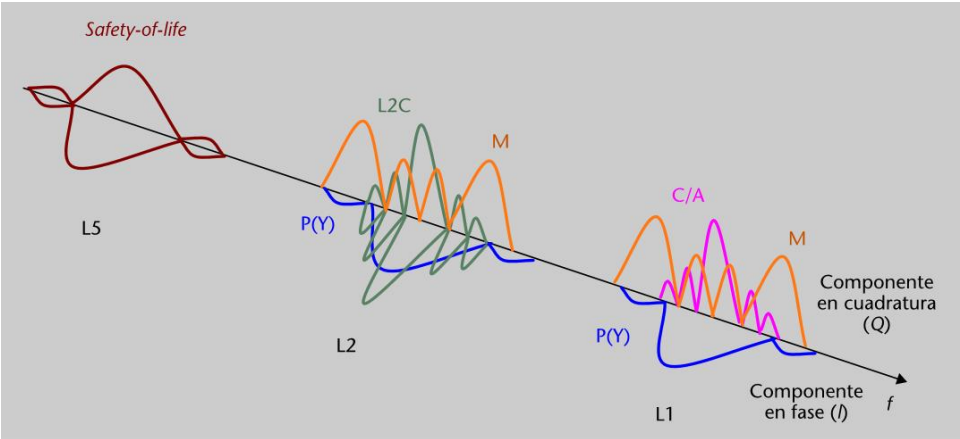

Ilustración 4. Conjunto de señales GPS a lo largo del espectro. Imagen obtenida de [3].

Meaconing y SCER

Un tipo de *spoofer* más complejo son aquellos que consiguen realizar **Meaconing**. Esta práctica conlleva la grabación de las señales GNSS auténticas para después reproducirlas a mayor potencia que la original. De esta forma se puede hacer *spoofing* a cualquier señal, incluso a una militar encriptada. Según su complejidad y sofisticación, los *meaconers* pueden divirse en:

- Los más sencillos utilizan una única antena de recepción. Las fases del código que emite son las mismas que en la señal original, con un desfase añadido, generado por el tiempo de procesamiento y de la propagación de la señal. Además, el receptor de la víctima recibirá el posicionamiento falso modificado de la antena del receptor del *meaconer*, y el tiempo falso modificado será ligeramente menor que el real.

- Un *meaconer* más sofisticado cuenta con múltiples antenas de recepción y un procesamiento de señal de array de fases. Esto hace posible grabar y reproducir múltiples canales de forma individual, lo que permite generar variaciones del desfase independientes. Gracias a ello consigue simular localizaciones falsas para que la víctima tome un posicionamiento erróneo.

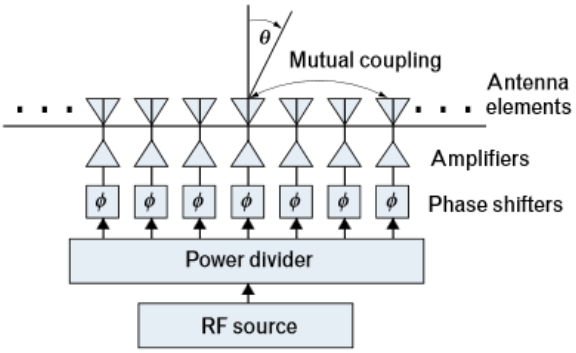

Ilustración 5. Emisor con antena de array de fases. Imagen obtenida de [4].

En el caso de que la parte impredecible de la señal contenga únicamente los bits de bajo bitrate de $D_i(t)$, existe la posibilidad de realizar un *spoofing* sin utilizar la técnica de meaconing. Se trata en este caso ataque por Security Code Estimation and Replay (**SCER**), donde el *spoofer* estima los bits impredecibles de $D_i(t)$ y los transmite en cuanto tiene una estimación fiable. Se puede configurar su funcionamiento, modificando el tiempo de procesamiento dedicado generar una estimación, de forma que se obtenga una estimación más fiable a cambio de un desfase en la transmisión mayor.

A diferencia del *meaconer*, no necesita un receptor de múltiples antenas con ganancias independientes para conseguir inducir un posicionamiento falso en la víctima. Sin embargo, es difícil que este tipo de técnicas surta efecto contra códigos de ensanchamiento completamente encriptados o con segmentos cortos encriptados [2]. Como ventaja, este sistema puede compensar una baja probabilidad de acierto en chips generados al aumentar la potencia de salida de transmisión.

Formas avanzadas de spoofing

En la técnica conocida como *Nulling*, el *spoofer* transmite dos señales por cada señal que se quiere suplantar. Una de ellas, como en el resto de casos, busca inducir un falso posicionamiento y temporización. La otra es la señal auténtica desfasada 180º, de forma que se cancele la recepción de la señal original por parte de la víctima.
Conseguir tener éxito mediante un ataque de *Nulling* es difícul, puesto que conseguir el alineamiento exacto de fase de portadora y la replicación de la amplitud de la señal no es una tarea sencilla. Para conseguir un ataque de *Nulling* exacto se requiere la calibración de diferentes parámetros, como ganancia de antena y patrones de fase [2]. Se ha constatado que la presencia de una segunda antena en el *spoofer* puede ayudar a la calibración en esta técnica.
Un tipo específico de ataque de *Nulling* es el que usa únicamente señales para anular las señales auténticas. En lugar de usar pseudorrangos falsos para inducir el posicionamiento falso en la víctima, utiliza efemérides de satélites y datos falsos de calibración de reloj.

Técnicas de defensa ante *spoofing*

La defensa ante *spoofing* busca la detección de un ataque y la recuperación por parte del sistema del posicionamiento y temporización auténticos. La mayoría de los esfuerzos en la investigación para la defensa ante ataques de spoofing se centran más en la parte de la detección de ataques que en la recuperación de los mismos. Por esto, queda mucho camino que recorrer [2]. Existen dos tipos de ataques a sistemas de navegación GNSS de los que es muy difícil recuperarse: los ataques que buscan la anulación de las señales de navegación auténticas y las que utilizan un *spoofer* emitiendo a muy alta potencia, especialmente si se consigue saturar el receptor de radiofrecuencia de la víctima.

Técnicas avanzadas de procesado de señales para receptores de una única antena

Existen diversos métodos de detección de ataques que pueden ser implementados mediante algoritmos de procesado de señal. Los más sencillos buscan cambios bruscos en la señal, ya sea en su amplitud, fase de portadora o fase de código.

La técnica de Received Power Monitoring (RPM) comprueba la potencia recibida en valor absoluto. Un aumento repentino de potencia podría significar un ataque, especialmente si es mayor de 1 ó 2 dB.

Otra técnica consiste en comprobar la función de autocorrelación de la señal recibida, la cual mostrará distorsión si se trata de una señal falsificada, debido al desalineamiento en la fase de la portadora en los momentos iniciales del ataque. Esta anomalía no será sencilla de detectar en señales afectadas por el efecto multicamino, o en aquellos ataques que buscan saturar el receptor con potencias muy altas, aunque pueda limitarse este efecto usando RPM para detectar este tipo de señales.

Este tipo de defensas cuenta con una desventaja. Sólo son capaces de detectar los ataques en su estado inicial del mismo, durante el arrastre del bucle de seguimiento. Una vez terminada esta fase, el *spoofer* ha conseguido arrastrar el bucle de seguimiento del receptor y desaparecen las variaciones abruptas en los parámetros [2]. En esta fase del ataque, el *spoofer* puede reducir la potencia de emisión a niveles habituales y evitar así ser detectado.

Existe un tipo de defensa ante aquellos ataques que ya han tenido éxito en el receptor, basada en una búsqueda de señales replicadas. Si se detecta una señal duplicada, la víctima pasa al estado inicial de adquisición de señales y trata de descubrir cuál de las señales replicadas es la señal GNSS auténtica, y recuperar el posicionamiento y temporización correctos. Esta técnica puede ser burlada por el atacante mediante la emisión de señales de potencia muy alta para interferir en la detección de las señales auténticas. Como contrapartida, el método RPM sería capaz de detectar señales de potencias tan elevadas.

Defensas basadas en la encriptación.

A día de hoy, las señales GNSS utilizadas en aplicaciones civiles se transmiten en texto plano, de forma que los receptores civiles que no cuentan con sistemas

de defensa toman como auténtica cualquier señal que tenga la estructura GNSS. Como hemos visto, existen diversas alternativas para implementar sistemas de defensa ante ataques de *spoofing*, y la criptografía es una de ellas.

Técnicas de encriptación de clave simétrica

Del mismo modo que se hace en las aplicaciones militares, emisor y receptor comparten una misma clave secreta. El mensaje de navegación $D_i(t)$ es encriptado por parte del emisor usando esta clave, y sólo los receptores que dispongan de ella serán capaces de descifrar el mensaje.
Este tipo de técnicas tienen la ventaja de ser muy resistentes ante ataques de *spoofing* [5]. Por otro lado, la distribución de la clave secreta a los receptores a través de un canal de comunicaciones seguro sin que sea interceptada por terceros limita su aplicación para sistemas GNSS de aplicación civil.

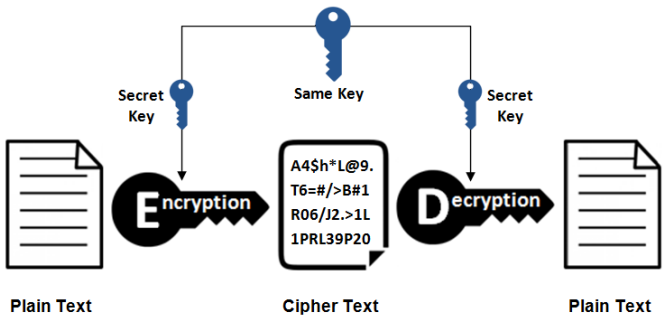

Ilustración 6. Técnica de encriptación por clave simétrica. Imagen obtenida de [6].

Otra aproximación consiste en utilizar la encriptación mediante claves simétricas sobre el código de ensanchamiento completo $C_i(t)$ [2]. Se trata de una buena defensa ante los ataques de meaconing, y existen diversas formas de usar la encriptación para generar segmentos impredecibles en las señales. Para esta implementación, se mantiene el requisito de que tanto el satélite GNSS transmisor como el receptor han de tener copias de la clave secreta, por lo que continúa siendo necesario contar con un medio seguro para distribuir las claves a los receptores.

Pueden implementarse técnicas de clave simétrica sin que tanto emisor como receptor compartan una clave secreta, lo que evita la necesidad de que exista un medio seguro de distribución de claves. En su lugar, se saca provecho de la relación que existe entre el código de ensanchamiento de las señales open-service de uso civil y las señales encriptadas de uso militar. Este método de defensa puede utilizarse en receptores GNSS civiles.

En sistemas GPS, tanto el código de ensanchamiento Open-Service civil como el encriptado para uso militar se envían en cuadratura en la misma portadora. Esto conlleva que la función de correlación cruzada entre ellas sea alta en el caso de que ambas sean auténticas. Gracias a esta relación entre ambas, se puede contrastar la autenticidad de la señal recibida por parte de una posible víctima. Como requisito en este caso, será necesario comparar la señal Open-Service recibida por el receptor civil con la señal militar encriptada recibida por un receptor encriptado, el cual debemos asegurarnos de que no ha sido comprometido para que la verificación se realice de forma segura.

Un método que combina le encriptación del código de ensanchamiento con la utilización y distribución de claves simétricas es el de encriptación de clave simétrica desfasada [2]. Consiste en intercalar segmentos cortos de un código de seguridad de espectro ensanchado (SSSC) con segmentos largos de código de ensanchamiento en la señal $C_i(t)$. El receptor almacena las partes desconocidas de la señal $C_i(t)$, y utiliza las partes conocidas para el seguimiento habitual GNSS. Después, el receptor recibe una clave en $D_i(t)$, la cual es firmada y enviada al segmento de control para su validación. Una vez hecho esto, se utiliza la clave para sintetizar el código de ensanchamiento desconocido que se había almacenado. Finalmente, el receptor comprueba la correlación entre ambas partes de la señal para comprobar la autenticidad de la misma. Debido al tiempo que conlleva el firmado digital, este método implica una latencia alta que puede llegar a varios minutos.

Técnicas de encriptación de clave asimétrica

En este tipo de sistemas, la clave se divide en dos partes.

a. La clave privada, conocida sólo por el emisor y usada para generar el mensaje encriptado.
b. La clave pública, distribuida a los receptores para que puedan desencriptar el mensaje de navegación y verificar que proviene de una fuente fiable.

En este tipo de técnicas es necesario definir un método de distribución de la clave pública o PKI para asegurar que proviene de una fuente fiable. Como desventaja, para obtener un nivel de seguridad similar al de las claves simétricas, se requiere que las claves asimétricas sean de una longitud mucho mayor [7]. Esto afecta a los recursos computacionales necesarios para su implementación, aumentando tanto las especificaciones hardware requeridas como el tiempo de computación.

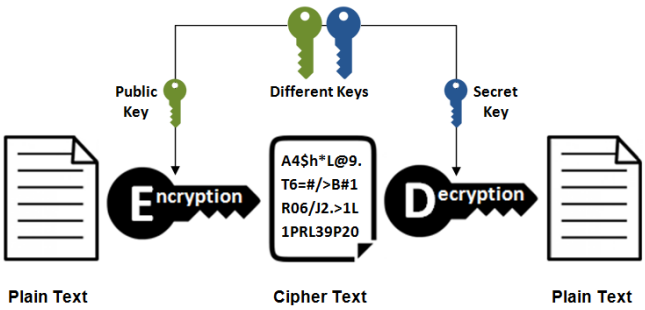

Ilustración 7. Técnica de encriptación por clave asimétrica. Imagen obtenida de [6].

Técnicas de autenticación de mensajes

La autenticación de mensajes se usa en múltiples campos en el ámbito de las comunicaciones. Como objetivos principales, busca proporcionar al receptor la seguridad de que el mensaje recibido que se ha recibido:

a. No ha sido modificado por un intermediario y es el mismo que ha sido transmitido en su origen.
b. Proviene de una fuente fiable.

La aplicación del concepto de autenticación de mensajes en el ámbito de los sistemas GNSS conduce al Navigation Message Authentication (NMA).

3. NMA. Técnicas de autenticación por encriptación de mensaje de navegación

La encriptación del mensaje de navegación permite aumentar lo impredecible que son las señales para hacerlas más robustas ante ataques. Esto dificulta la generación de señales falsificadas por parte de un *spoofer*.

En la autenticación de mensajes, el emisor utiliza una clave secreta para generar una firma a partir del mensaje original. Después, envía tanto el mensaje como la firma para que sean obtenidos por el receptor. Para asegurarse de que el mensaje es auténtico, éste utiliza una clave de receptor para verificar que el mensaje y la firma se corresponden.

Para generar las firmas de autenticación se utilizan principalmente dos técnicas, las cuales se han tratado en el capítulo anterior. Si la clave del emisor para generar la firma y la clave del receptor para verificar el mensaje son la misma, entonces se trata de un sistema con clave simétrica. Si no son la misma, el sistema es de clave asimétrica.

Este Trabajo se centra en estudiar las implementaciones de esta metodología de autenticación de mensajes a las señales utilizadas en el ámbito civil de diversos sistemas GNSS. A continuación, se procede a desglosar las diferentes implementaciones que están desarrollando los sistemas GNSS.

Galileo

Galileo es el programa europeo de GNSS, cuyo desarrollo se lleva a cabo entre la Comisión Europea y la Agencia Espacial Europea. Está diseñado principalmente para ofrecer servicios de uso civil. Estos son Open Service (OS), Commercial Service (CS), Safety of Life (SOL) y Search and Rescue (SAR).
Mientras que los datos emitidos para el servicio Commercial Service son protegidos mediante cifrado, los datos de servicios como Open Service se envían en texto plano. En este último servicio se centrará el sistema de autenticación de mensajes que se está implementado en Galileo en la actualidad.

Ilustración 8. Sistema GNSS europeo Galileo.

La implementación del NMA en Galileo se denomina Open Service Navigation Message Authentication (OSNMA), y se trata de una aproximación del algoritmo TESLA aplicado a las especificaciones de este sistema.

Timed Efficient Stream Loss-tolerant Authentication (TESLA)

El algoritmo TESLA es un mecanismo de autenticación mediante *broadcast*, cuyo funcionamiento es un híbrido entre las técnicas de claves simétrica y asimétrica. Sus principales características son:

- Los requisitos de recursos de computación y de comunicación para su funcionamiento son bajos.
- Alta resistencia ante la pérdida de datos, como es el caso de receptores GNSS en entornos con visibilidad reducida.
- Óptimo para transmisiones de uno a varios (*multicast*), como es el caso de GNSS.

El algoritmo se basa en la transmisión de un Message Authentication Code (MAC) para autenticar el mensaje que es transmitido en texto plano. Para generar el MAC se utiliza una clave, la cual se transmite con un retardo temporal con respecto a la transmisión de su MAC.

Para que TESLA pueda utilizarse en aplicaciones de seguridad deben cumplirse los siguientes requisitos:

a) Las funciones utilizadas deben ser criptográficamente seguras.
b) La conexión entre emisor y receptor debe cumplir el requisito de *loose time synchronization*. Según esta, el momento de recepción de un paquete no debe exceder el límite esperado, delimitado por el error

máximo en el tiempo de sincronización [8]. Si lo excede, el paquete es descartado.

c) Se requiere la utilización un sistema de autenticación diferente en la inicialización del sistema.

En el emisor, se genera el MAC mediante la combinación del mensaje y la clave privada. Tanto el mensaje como el MAC son transmitidos, y un tiempo después, se envía también la clave privada. De esta forma, el sistema se asegura de que no se reciba la clave privada utilizada para generar la MAC hasta que el mensaje y el MAC hayan llegado a su destino [7]. Mediante este método se cumple con el primer requisito de la autenticación de mensajes[a], según el cual el mensaje no ha sido modificado en su camino. Sin embargo, este procedimiento no cumple con el segundo requisito [b] de asegurar que procede de una fuente fiable. Para solventar este problema se utiliza el método de la cadena de claves, basado en clave simétrica.

En el método de la cadena de claves se genera de forma aleatoria una clave inicial llamada clave semilla (*seed key*), K_n, la cual y tiene una longitud fija (L). A esta clave inicial se le aplica una función de una sola dirección que suele ser una función hash criptográfica. A partir de la clave semilla y aplicando la función hash criptográfica F se obtiene un hash, que al ser la primera será la clave inicial $K_{n-1} = F(K_n)$. El hash o clave obtenida se introduce de nuevo en la función F, y se repite de forma iterativa n veces, generando una cadena de 'n' claves. Finalmente se obtiene la clave raíz (*root key*), $K_o = F^n(K_n)$, que será el resultado de aplicar 'n' veces la función F a la clave semilla.

Una vez se han generado todas las claves, está creada la cadena de claves. Después se generan los MAC, combinando cada uno de los mensajes de navegación con su clave asociada mediante $MAC_i = S(M_i, K_i)$, donde S es el algoritmo utilizado para la autenticación, M_i es el mensaje de navegación y k_i es la clave necesaria para obtener M_i a partir de MAC_i.

En las técnicas de clave simétrica existe la necesidad de distribuir las claves privadas de forma segura. Para cumplir con el requisito de TESLA en las aplicaciones de seguridad, se utiliza un sistema diferente de autenticación basado en clave asimétrica. Se genera una firma digital asociada a la clave raíz, para que el receptor pueda comprobar su autenticidad en la inicialización del sistema. Finalmente, el emisor comienza a enviar las claves de forma inversa al orden en el que fueron generadas, comenzando por la clave raíz.

Cada uno de los paquetes emitidos P_i está compuesto por $P_i = [M_i, MAC_i, K_{i-d}]$, con $d > 0$, donde M_i es el mensaje i, MAC_i es el MAC asociado a mensaje M_i, y K_{i-d}, que será la clave que relaciona MAC_{i-d} y M_{i-d}, contenidos en el paquete que se transmitió con anterioridad [5].

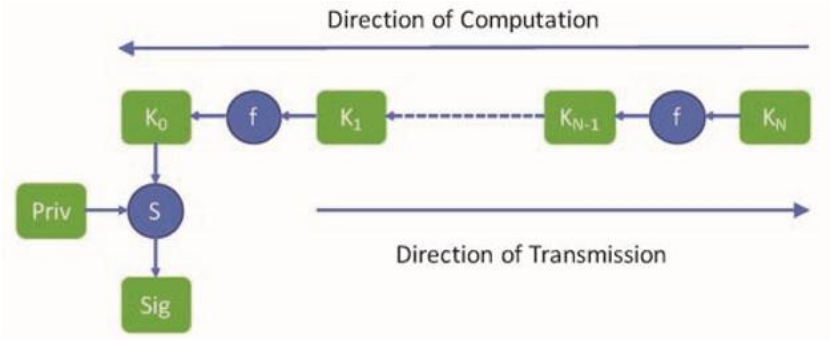

Ilustración 9. Generación de una cadena de claves [7].

Una vez el receptor obtiene la clave raíz y su firma digital, si ambas concuerdan se autentifica la recepción. Después, se trata ir deshaciendo la cadena de claves para obtener los diferentes MAC y poder autentificar los mensajes.

El receptor no puede autenticar el paquete recibido de forma instantánea porque no conoce la clave del MAC que contiene. Para ello, tendrá que esperar 'd' instantes hasta recibirla. Una vez la ha obtenido, puede generar su propia versión del MAC_i con la clave K_i y el mensaje M_i, que será la $\widehat{MAC_i}$. Si este MAC generado localmente $\widehat{MAC_i}$ concuerda con la recibida previamente del transmisor MAC_i, entonces el mensaje ha sido autenticado de forma satisfactoria.

Puesto que el receptor conoce la función F, y se trata de una función de una sola dirección, el receptor puede verificar si las clave recibidas pertenecen a la misma cadena de claves. Sin embargo, no puede predecir las claves futuras que aún no se han recibido. En el receptor, este proceso se repite en el orden opuesto al que se ha seguido en el emisor. No es necesario repetir este proceso completo hasta desencriptar toda la cadena de claves cada vez que se reciba una de las claves. Sólo será necesario que el receptor haga la comprobación de la cadena hasta alcanzar la última clave autenticada mediante firma digital, sea esta la clave raíz u otra más reciente.

A continuación, se enumeran las vulnerabilidades criptográficas que afectan al algoritmo TESLA, así como su relación con los requisitos de seguridad que se han descrito previamente en este capítulo:

- Tiempo de sincronización entre emisor y receptor elevado. Existe la posibilidad de que una vez se ha publicado una clave, pueda ser usada por un atacante para crear un mensaje falso con un MAC falso. Esta vulnerabilidad está relacionada con el tiempo de espera entre la publicación de claves, o ventana de ataque [9]. Afecta al requisito de seguridad b), directamente relacionado con la aplicación del requisito de *loose time synchronization*.
- Vulnerabilidades en la autenticación de la clave raíz. Puesto que es necesario un método de autenticación por clave simétrica, si el algoritmo utilizado para esta tarea es frágil, un atacante podría romperlo y crear una cadena de claves falsa y firmarla como auténtica, engañando al receptor. Esta vulnerabilidad está relacionada con los requisitos de seguridad a) y c).
- Descubrimiento de la cadena de claves. Mediante ataques de fuerza bruta basados en la computación se puede llegar a obtener la cadena de claves

y conseguir la clave más reciente utilizada. Esta vulnerabilidad está relacionada con el requisito de seguridad a).

Open Service Navigation Message Authentication (OSNMA)

La autenticación de mensajes de navegación de sistemas GNSS mediante el algoritmo TESLA se realiza de la siguiente forma:

- El receptor obtiene los datos de navegación y el MAC.
- El receptor obtiene más tarde una clave a partir de la cual se puede generar el MAC a partir de los datos de navegación anteriores y la clave recién recibida.
- El receptor autentica la clave actual con una clave que ha sido recibida previamente, la cual se trata de la clave raíz o una anterior que fue firmada digitalmente, y por tanto se considera auténtica. Para ello se realiza la función F las veces que sea necesario hasta llegar a la clave que fue firmada.
- El receptor genera un MAC de referencia con la clave del segundo paso y los datos de navegación del primer paso. Se compara esta MAC generada con la recibida en el primer paso. Si concuerda, los datos de navegación han sido autenticados.

Medidas de rendimiento

Uno de las prioridades a la hora de implantar una solución NMA es mantener el nivel de disponibilidad de servicio y la precisión de navegación ofrecida al usuario no. A la hora de evaluar la idoneidad de una solución y su aplicación dentro de un sistema, se utilizan diversos parámetros que afectan al rendimiento y que aportan información sobre su efectividad.

Uno de los parámetros utilizados en un sistema de navegación es el *Time To First Fix* (TTFF), o tiempo que se tarda en obtener un posicionamiento. En el caso de utilizar sistemas con sistemas de autenticación, se refiere al *Time To First Authenticated Fix* (TTFAF). El objetivo es reducir lo máximo posible la diferencia entre estos dos parámetros, de forma que el proceso de autenticación transparente para el usuario. [10]

La tasa de error de autenticación o *Authentication Error Rate* (AER) representa la probabilidad de error a la hora de autenticar un satélite en un entorno en el que no hay ataques, sino perturbaciones en el canal de transmisión. Equivale a la tasa de error de paquetes, incluyendo en éstos los datos necesarios para la navegación y la autenticación. Se calcula mediante $AER = 1 - (1 - BER)^{NNA}$, donde BER es la tasa de error de bit, mientras que NNA se refiere al número de bits usados en la autenticación, que incluyen los bits de navegación en texto plano NN y los bits de autenticación NA.

El tiempo que pasa entre dos autenticaciones consecutivas de un satélite por parte de un usuario, o *Time Between Authentications* (TBA). Este parámetro afecta al TTFAF, y por tanto al tiempo durante el cual un usuario puede ser víctima de *spoofing* sin que el sistema sea consciente de ello. Un TBA de 10 segundos implica durante este espacio de tiempo, cualquier ataque que altere la

información impredecible enviada por un satélite será indetectable por el sistema de autenticación.

Para caracterizar el nivel de protección de la señal contra ataques de repetición de señal o *replay attacks*, como son los ataques SCER, además de AER y TBA se utiliza el tiempo máximo en que la señal puede ser predecible, o *Maximum Predictable Time* (MPT). También es útil para la caracterización de la señal frente a *replay attacks* la tasa de símbolos imprecedibles, o Unpredictable Symbol *Ratio* (USR). Representa el porcentaje de símbolos impredecibles con respecto al total de símbolos a lo largo de un espacio de tiempo, y sirve de indicador para averiguar cuánto tiempo ha de esperar el receptor hasta obtener una estadística fiable que ayude a la protección contra *replay attacks*.

Como resumen de los temas tratados hasta el momento y para relacionarlos con los parámetros tratados en este apartado, la propuesta de un esquema de NMA para Galileo tiene como objetivo principal obtener un sistema de autenticación que obtenga un rendimiento lo más parecido posible al funcionamiento del sistema sin modelo de autenticación. Para ello, se busca minimizar los parámetros AER y TBA, a la vez que se aumenta la probabilidad de detección de ataques de repetición [10]. Esto se llevará a cabo implementando comprobaciones de señal en los receptores que utilizan el sistema de autenticación.

Generación de las claves y formación de la cadena

A la hora de implementar un sistema de autenticación se tienen en cuenta tanto la robustez del sistema ante ataques como la viabilidad de implantarlo sin que se vea afectada la disponibilidad del servicio. Por esto es importante encontrar un equilibrio entre estos aspectos para ofrecer una aproximación adecuada.

La robustez criptográfica que presenta el sistema varía en función de diversos parámetros y sus características. En el caso del proceso de generación de la cadena, ésta aumenta a medida que se rompe la simetría en cada una de las iteraciones para generar las claves [10]. Para ello se puede optar por añadir información conocida por el receptor al proceso de creación de hashes, como puede ser un contador o una etiqueta de tiempo. Para reducir los requisitos de comunicación se puede optar por reducir la longitud de las claves mediante el recorte o truncamiento de las claves, o hashes de salida de la función F. En [10] la función se propone con la siguiente estructura:

$$F\left(K_m, GST_j\right) = trunc(len, hash(K_m||GST_j))$$

Donde K_m es la clave usada como entrada a la función F y GST_j es la variable temporal de Galileo o *Galileo System Time*, asociado al authentication frame j sobre el cual se aplicará la clave que se va a generar. La Función $trunc$ recorta la salida de F a una longitud definida por len, $hash$ es la función hash usada y || es el operador para concatenar K_m y GST_j.

Incluir GST_j en el proceso de creación de claves hace el sistema más resistente ante ataques de pre-computación, en los cuales el atacante genera largas

cadenas de hashes y las almacena para que, en el caso de que concuerde con una que el emisor ha transmitido, pueda averiguar cuál es la siguiente clave dentro de la cadena. La etiqueta temporal GST_j hace inviable este tipo de ataques, puesto que el generar previamente los hashes tendría que especificar el instante de tiempo para el cual va a aparecer. Otras opciones para protegerse ante este tipo de ataques son aumentar la longitud de las claves, con el aumento de los requerimientos de comunicación que eso conlleva, o la introducción de patrones impredecibles en cada cadena y transmitiéndolo justo antes de que la cadena entre en funcionamiento.

Una única cadena y las mismas claves para diversos emisores

En una aproximación estándar de Tesla, cada emisor usa una cadena de claves propia. De esta forma, si un receptor autentica a tres emisores, en este caso satélites, debería recibir tres MACs y tres claves, una por cada satélite. Cada clave pertenece a una cadena distinta y necesita de una clave raíz distinta, por lo que el receptor necesitará almacenar tres cadenas de claves, aumentando los requisitos de almacenamiento y computación.

En el esquema propuesto para Galileo se utiliza una única cadena de claves para todos los satélites. Esto reduce el AER, permitiendo que todos los satélites se autentiquen mediante la misma cadena. También reduce el TTFAF y el número de bits necesarios para que un usuario pueda calcular el tiempo, su posición y su velocidad, o *Position Velocity Time* (PVT). Este sistema es especialmente útil para corregir el aumento de BER que aparece cuando algunos satélites se encuentran en ángulos de elevación bajos o se encuentran bajos los efectos del multicamino, como en entornos urbanos. Si de forma temporal no se consigue recibir ninguna clave, la recepción de claves posteriores permite autenticar los mensajes de las claves que no se han podido recibir en su momento.

Puede darse el caso de que varios satélites envíen la misma clave, la cual llegará al receptor en distintos momentos. Esto provoca que la clave emitida por un satélite es impredecible, mientras que el resto de claves pueden replicadas. Para solventar este problema se ha propuesto el siguiente funcionamiento, basado en transmitir una única cadena formada por diferentes claves desde cada emisor [10].

Una única cadena y diferentes claves para diversos emisores

En este esquema todos los satélites envían claves que pertenecen a la misma cadena, pero estas son distintas entre sí. De esta forma se consigue evitar la replicación de claves por medio de un ataque de *spoofing*. Además, si no se recibe un cierto número de claves por falta de cobertura o degradación temporal del canal de comunicación, al recuperar la conexión y obtener cualquier clave posterior de la cadena, ésta puede utilizarse para recuperar las claves anteriores que no se habían recibido. Por tanto, el receptor necesita una única clave procedente de cualquier satélite para autenticar todos los MACs enviados hasta ese momento, a la vez que consigue que todas las claves emitidas sean impredecibles.

Siguiendo este esquema, en un *authentication frame* j, la clave usada para crear el MAC_j será $K_{j,MAC}$, y podrá obtenerse a partir de la clave recibida por otro satélite i mediante el siguiente procedimiento:

$$K_{j,MAC} = F^{i-1}(K_{j,i}, GST_j)$$

Donde F^{i-1} es la función F aplicada $i-1$ veces sobre $K_{j,i}$, la clave recibida del satélite i en el authentication frame j. En este caso, la clave $K_{j,1}$ generada por el satélite 1 para generar MAC_j, de forma que $K_{j,1} = K_{j,MAC}$.
Así, para validar MAC_j será necesario la clave del authentication frame previo, generada mediante:

$$K_{j-1,MAC} = F^{S}(K_{j,MAC}, GST_{j-1})$$

Donde S es una constante que representa el número máximo de satélites, y por tanto el número máximo de veces que se aplicará la función F en un authentication frame.

Gracias a este sistema, un *spoofer* no será capaz de obtener la clave de un satélite a partir de la clave de otro satélite. Además, la complejidad de operaciones en el receptor disminuye al tener que almacenar una única cadena de claves para todos los satélites. La principal desventaja de esta aproximación es la reducción del número de bits impredecibles que se pueden utilizan en técnicas de anti-replay.

Autenticación cruzada (*Cross-Authentication*)

Para que el método de autenticación NMA pueda aplicarse en la actualidad, OSNMA debe ser generado en el segmento de control, situado en tierra, y transmitido en tiempo real al segmento espacial. Esto reduce el impacto que implica en la infraestructura del sistema.

Los satélites conectados con el segmento de control generan sus MACs a partir de sus mensajes de navegación. Los satélites que no tienen conexión con tierra, tendrán que transmitir sus mensajes de navegación en texto plano.
Puesto que no todos los satélites tienen conexión con tierra de forma continua, sus mensajes de navegación han de ser autenticados por parte de los satélites que en ese momento sí la tienen, y pueden transmitir datos autenticados.
Esto se consigue mediante la asignación de varios MACs a cada clave, lo que permite la autenticación cruzada entre diferentes satélites. Debido a esto, un satélite conectado a tierra puede transmitir sus propios MACs y también los MACs asignados a otros satélites no conectados.

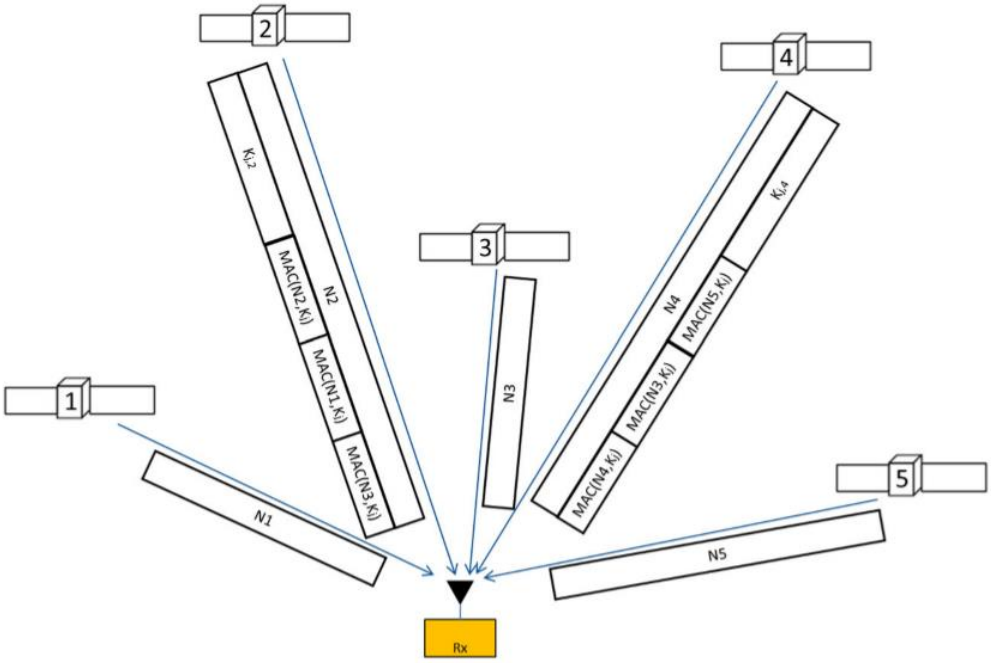

Ilustración 10. Cross-authentication. [2]

Este esquema permite que se autentique toda la información de navegación que reciben los usuarios del sistema. La generación de MACs de realiza del siguiente modo:

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

$MAC_{j,i,l}$ será el MAC truncado, también llamado *tag*, transmitido en el *frame* j por parte del satélite i para autenticar los datos de navegación del satélite l. En el caso de que el satélite esté conectado a tierra y pueda autenticar su propio mensaje, $i = l$. CTR será un contador con la posición del *tag* en la transmisión. $P_{j,l}$ se refiere a los datos de navegación que se van a autenticar, procedentes del satélite l en el frame j.

En un escenario en el que existen satélites no conectados a tierra y satélites que sí lo están, los datos navegación de un satélite no conectado son precedibles y se envían en texto plano, por lo que pueden ser víctima de un ataque de repetición o *replay attack*.

Decisiones de diseño.

Técnicas de firmado digital

Para que el sistema de autenticación se puede implementar, ha de contar con un sistema independiente gracias al cual el receptor pueda obtener información certificada, como una clave pública. Elegir un método asimétrico evita la necesidad de que los usuarios compartan una clave secreta, los cuales se dividen en dos grandes grupos:

- Firmas digitales, como RSA, DSA o ECDSA. Los satélites transmiten una firma digital dentro de sus datos de navegación. Como ventaja cuenta con métodos y funciones estandarizados que los hacen fiables

criptográficamente. Por contra, requieren de recursos computacionales para cada autenticación y de ancho de banda para transmitir la información de autenticación.

- Envío desfasado de clave simétrica, como TESLA. Tienen tolerancia ante la pérdida de datos y requieren de un ancho de banda menor, pero al no estar estandarizados en el ámbito criptográfico, son potencialmente más vulnerables ante ataques.

En la implementación propuesta para OSNMA, la frecuencia de uso de la clave pública y el ancho de banda necesario son muy bajos.

Requisitos de computación

Teniendo en cuenta un procesador de 1 GHz, se necesitarían 0,4 microsegundos para una iteración mediante una función hash SHA-256 de 32 bytes, que se extenderían a 40 microsegundos en el caso de autenticar más de 30 satélites. Para verificar una clave raíz con una antigüedad de una semana tardaría alrededor de 2,5 segundos. Ambos son tiempos admisibles para una implementación satisfactoria del sistema.

Consideraciones de seguridad.

Se tratan a continuación algunos aspectos relacionados con la seguridad del sistema:

- Una única cadena de un solo sentido: al utilizar una única cadena de claves para todos los satélites, si ésta es comprometida todo el sistema también lo estará, como en el caso de que un atacante obtenga la clave semilla. Puesto que la seguridad de la cadena depende del hash primitivo y de la longitud del hash obtenido, el nivel de seguridad puede modificarse mediante la elección de la función hash y la longitud de las claves. En el caso de necesitar un nivel seguridad mayor, se puede hacer más restrictivo el requisito de *loose time synchronization* y reducir el periodo de validez de cada cadena o aumentar la longitud de las claves. La implementación propuesta en [10] permite la modificación de la longitud de las claves y MACs, así como la elección de las funciones criptográficas en función de las necesidades.

- Recorte de MAC y salidas de funciones hash: en sistemas con un caudal de bits muy reducido como son los GNSS, estos parámetros son muy sensibles puesto que afectan a AER y TBA. Por tanto, su longitud debería reducirse lo máximo posible siempre que se mantenga un nivel de seguridad aceptable. Puesto que el periodo de transmisión de MACs y claves se puede configurar, pueden usarse MACs muy reducidas. Una MAC de 10 bits obtendrá una probabilidad de ser comprometida menor al 0,1%. Si a esto se le añade una configuración de un TBA bajo, la probabilidad de que un ataque tenga éxito se reducirá aún más.

- Autenticación de clave raíz y gestión de claves públicas: para la emisión y gestión de claves puede utilizarse una autoridad de certificación PKI externa. Para que una pareja de claves pública y privada pueda ser revocada es necesario emitir nuevas claves públicas a los usuarios. Esto puede hacerse mediante un sistema Over-The-Air (OTA) de distribución

de claves públicas o mediante una conexión que pueda realizar el receptor con una red de distribución de claves. Cualquiera de estas soluciones de gestión de claves pública y privada permite conseguir una elevada autonomía a los receptores.

Implementación

La implementación propuesta en [10] se basa en introducir el concepto NMA en el interior de la estructura de mensaje I/NAV. Estos se transmiten en las señales E5b-I y E1-B, lo que permite un aumento de la tasa de recepción en aquellos receptores que tengan capacidad de recibir en ambas frecuencias, aunque también es compatible con los receptores de una única frecuencia. En este caso, la implementación se centra en la parte del mensaje I/NAV transmitido mediante la señal E1-B.

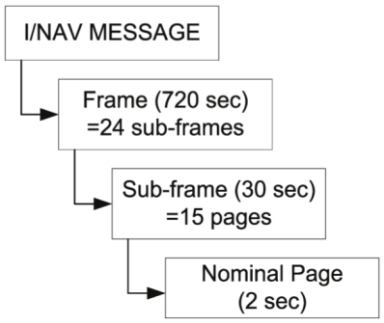

Ilustración 11. Estructura de mensaje I/NAV. Imagen obtenida de [11].

Un *frame* de I/NAV está compuesto de 24 *subframes* de 30 segundos de duración. Cada *subframe* se divide en 15 páginas (*pages*), con 2 segundos de duración en el modo nominal. Cada page está formada por dos partes de igual duración, una denominada *even* (par) y otra *odd* (impar) [12].

En la implementación de [10] se propone utilizar el campo 'Reserved 1' para transmitir la información NMA. Puesto que se encuentra en la parte *even* de cada page en modo nominal, permite la transmisión de 40 bits cada 2 segundos, obteniendo una tasa de transmisión de 20 bps, que serán 600 bits por cada *subframe*. Las razones que han aportado los autores de la especificación para decidirse por este campo en lugar de otros disponibles es que puede transmitirse a los satélites con un impacto muy bajo sobre las tareas centrales de navegación y control. Además, al utilizar este campo, los datos de autenticación NMA se diseminan dentro del propio mensaje de navegación. Esto permite hacer las tareas de autenticación más robustas ante ataques puesto que se reduce el tiempo máximo en el que el mensaje es predecible, definido en el parámetro MPT.

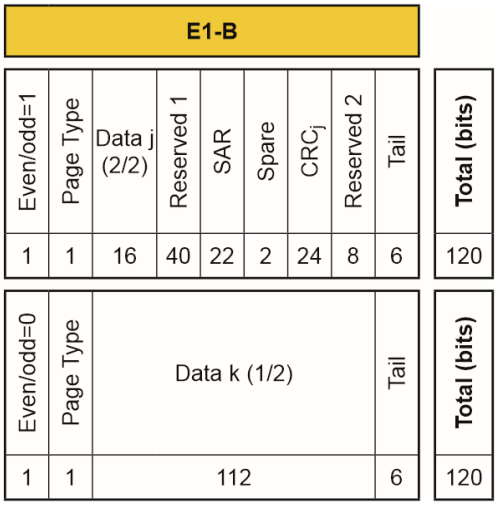

Ilustración 12. Presentación de una page I/NAV en modo nominal. [13]

El espacio utilizado dentro de cada page será de 40 bits, reservados para la transmisión de datos NMA. Esta información de autenticación se basa en dos secciones, las cuales serán transmitidas de forma paralela:
Sección H-K-root, con una cabecera global y una clave raíz firmada digitalmente mediante *Digital Signature Message* (DSM).
Secciones MAC-K, con los MACs truncados y sus claves desfasadas asociadas.

El servicio de autenticación se basa principalmente en la sección MAC-K, que ocupará 32 bits de los 40 disponibles, dejando los 8 bits restantes para la sección H-K-root.
La autenticación de MAC-K implementa un esquema de autenticación basado en el algoritmo TESLA. Para autenticar las claves usadas en esta sección, se utilizan las claves semilla (*root key*) de la sección H-K-root. Puesto que ambas secciones se transmiten en paralelo, el emisor ofrecerá una clave semilla firmada digitalmente de forma continua. El receptor leerá esta sección para obtener una clave semilla solo cuando lo necesite, lo cual será de forma menos frecuente que la lectura de la sección MAC-K. Al separar estas secciones, se mantiene un nivel constante de impredecibilidad en el mensaje y permite una mayor flexibilidad en el diseño de la solución.

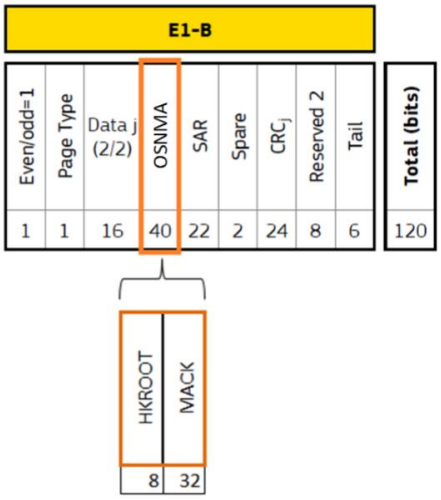

Ilustración 13. Secciones dentro de la estructura OSNMA. [13]

Sección H-K-root

Contiene tres campos:

- Cabecera NMA (*NMA Header*): define el estado del servicio NMA, que puede ser operacional, en modo testeo o modo de NMA deshabilitado, informando al receptor de que debe evitar el uso de los datos NMA para navegar. También define el identificador de la cadena en operación, así como el estado de la misma. Es en este campo donde se indica si la cadena en operación o una utilizada previamente ha sido revocada y no debe usarse para autenticación.

- Cabecera DSM (*DSM Header*): habitualmente mensaje firmado en el DSM es una clave raíz TESLA, o KROOT, aunque también puede contener una clave pública nueva. En este campo es donde se define este aspecto, además de el identificador del DSM.

- Bloque DSM (*DSM-KROOT*): se utiliza para autenticar la clave raíz (KROOT) de la cadena en operación o la cadena siguiente. Para ello utiliza una clave pública confiable conocida por el receptor. Puesto que el proveedor de OSNMA se encargará de publicar de forma segura una o varias claves públicas a los receptores, también contiene el identificador de la clave pública (PKID) a utilizar para autenticar la clave que contiene en su interior.
  Contiene la firma digital (DS) de la clave raíz y la propia clave raíz (KROOT)
  También contiene otra información importante, como la información de la función hash que se utiliza en la cadena (SHA-256, SHA3-224), el tipo de función MAC utilizada en las claves de la cadena (HMAC-SHA-256, CMAC-AES), el tamaño de las claves de la cadena (entre 80 y 256 bits), el tamaño de los MAC generados (entre 10 y 32 bits).

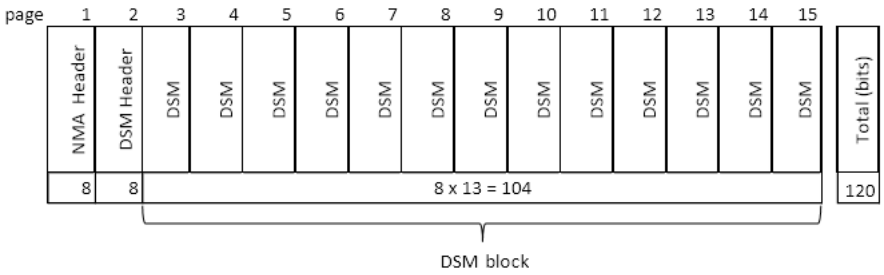

Ilustración 14. Cabecera y campos de la sección H-K-ROOT en subtrama de I/NAV. [13]

Sección MAC-K

Puesto que una misma clave puede tener varios MACs, en una la sección MAC-K se almacena la clave, sus MACs asociados y la sección MAC-Info asociado a cada uno de ellos. Se puede calcular el número de MACs que puede almacenar una sección MAC-K mediante la siguiente fórmula [13]:

$$n_m = floor\left(\frac{l_{MK} - l_K}{l_M + l_{MI}}\right)$$

Donde n_m es el número de MACs por sección MAC-K, l_{MK} es la longitud de la sección MAC-K, l_M es la longitud de un MAC y l_{MI} es el tamaño de la sección MAC-Info, todos ellos definidos en DSM-KROOT. $floor(x)$ se refiere a la función que devuelve el mayor entero posible que no sea mayor que x.

Tomando un tamaño de clave de 82 bits y un tamaño de MAC truncado de 10 bits, se pueden llegar a transmitir 3 MACS por cada sección MACK-K. Por cada *subframe*, esto hacen 9 MACs. Si estos MACs se refieren a satélites distintos, se puede llegar a autenticar hasta 9 satélites cada 30 segundos.

Las secciones que constituyen la sección MAC-K son:

- MAC: se trata de un MAC truncado, también llamado *tag*.

- MAC-Info: incluye el PRN del satélite que transmite la información que se desea autenticar. Ya que se almacena en 8 bits permite identificar hasta 255 satélites distintos gracias, lo que permite la autenticación cruzada entre satélites y permite la posible implementación en un futuro de autenticación cruzada entre distintas constelaciones GNSS. Contiene también el *Issue-Of-Data* (IOD) de la información autenticada.

  El campo *Authentication Data & Key Delay* (ADKD) proporciona información sobre los datos de navegación que se han firmado. Puesto que cada MAC puede firmar diferentes tipos de información, este campo permite concretar si el MAC en particular se va a utilizar para autenticar, por ejemplo, las efemérides, la subtrama, el almanaque, la información ionosférica, o todas ellas de una vez. Dividir los parámetros a autenticar en diferentes MACs tiene un efecto positivo, ya que al autenticar una menor cantidad de bits se reduce el AER.

  Este mismo campo permite la utilización de slow MACs. En el caso de que se busque relajar el requerimiento de seguridad de *loose synchronization*

en el receptor, los MACs pueden validarse con una clave transmitida un tiempo después. El contenido del campo especifica el número de subtramas de diferencia que habrá entre la emisión del slow MAC y su clave con respecto al desfase que habría tenido en condiciones normales [13]. Además, se consigue evitar cierto tipo de ataques de spoofing [10].

- Clave: contiene la clave perteneciente a la cadena de claves TESLA.

La ventaja de asociar un IOD con un conjunto de datos de autenticación es que asegura que la latencia de autenticación tiene muy bajo impacto en el rendimiento de NMA [9] [10]. Mientras un usuario receptor espera a que se valide una un IOD de navegación (IODnav), puede utilizar el IOD recibido previamente hasta recibir el subframe que valide la información nueva. Esto permite obtener una degradación nula o muy baja en el rendimiento de navegación durante un tiempo de 30 segundos.

El campo utilizado para implementar OSNMA en la señal E1b contiene 40 bits por *page*, lo que permite implementar una tasa de transmisión de hasta 20 bps. Puesto que este parámetro está directamente relacionado con el tiempo entre autenticaciones, permite alcanzar un TBA muy bajo, al tiempo que consigue una alta redundancia en la autenticación cruzada entre satélites.

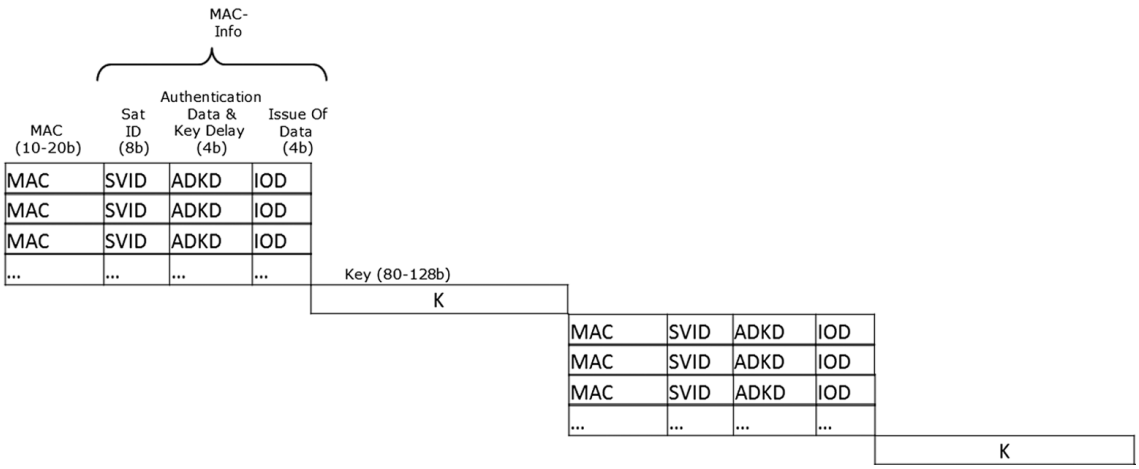

Ilustración 15. Estructura de sección MAC-K. [10]

Protección ante ataques de repetición en OSNMA

NMA está orientado a la autenticación de los datos de navegación de los satélites. A diferencia de los datos de navegación en plano, la información criptográfica generada en el proceso de autenticación se vuelve impredecible, algo beneficioso en términos de seguridad del sistema ante ataques. La protección contra ataques de repetición está sujeta las implementaciones que puedan hacerse en el receptor, aunque en este caso se puede hacer uso de los símbolos impredecibles en los que se codifica el mensaje NMA.

La mejor protección contra los ataques de repetición, o *replay attacks*, se consigue mediante autenticación a nivel de código de ensanchamiento o *Spreading Code-level Authentication* (SCA). Esta posibilidad no es viable para

los usuarios de servicios Open Service en la primera generación de Galileo. Por lo tanto, aumentar cómo de impredecibles son los datos a través del sistema OSNMA es una alternativa deseable para mejorar la defensa ante este tipo de ataques, siempre que el rendimiento en la navegación basado en autenticación no se vea reducido.

Al igual que otras señales utilizadas en Galileo, el mensaje I/NAV se codifica mediante convolución. Cada bit de datos de navegación se codifica, resultando en dos símbolos o bits codificados. Puesto que los bits pertenecientes al mensaje de navegación en texto plano son predecibles y los datos introducidos en este proceso no lo son, se obtiene un mensaje formado por un bit predecible y otro impredecible a continuación. Según la definición del sistema OSNMA en [13], los primeros 8 bits de los 40 bits que forman la estructura NMA son predecibles, y pertenecen a la sección H-K-ROOT.

Para conseguir que el mensaje generado sea más robusto ante efectos temporales de desvanecimiento en el canal, los bits codificados que lo forman se interpolan mediante un sistema de ecuaciones lineales [14]. En este sistema, cada símbolo recibido es una nueva ecuación, y los bits impredecibles son incógnitas. Una vez el número de ecuaciones (símbolos recibidos) iguala el número de incógnitas (bits impredecibles), el sistema de ecuaciones se puede resolver, situando los bits en las posiciones obtenidas en la solución.

Tanto el proceso de codificación mediante convolución como el de interpolado de los símbolos mantienen la entropía y la impredicibilidad de la señal. Esta característica puede utilizarse para que los receptores capaces de utilizar NMA puedan averiguar si una señal que está recibiendo está siendo replicada mediante un *spoofer*. Puesto que un ataque de repetición necesita de una estimación, la cual conlleva un retraso en la transmisión de la señal replicada, el atacante no es capaz de transmitir los símbolos impredecibles correctos a tiempo.

El receptor sabe dónde se encuentran los símbolos impredecibles, ya que almacena las primeras muestras recibidas de cada uno de ellos. Gracias a ello, crea una secuencia cuya ganancia de correlación se verá reducida en el caso de que la señal que se encuentra en seguimiento comience a ser replicada por un *spoofer* [14]. Por otro lado, este método cuenta con la desventaja de que la secuencia de correlación pierde las propiedades de correlación cruzada de los códigos PRN, las cuales ayudan reducir la interferencia entre satélites. Estos efectos negativos pueden reducirse si la secuencia es suficientemente larga y el número de símbolos es alto.

QZSS

El sistema GNSS japonés denominado Quasi-Zenith Satellite System (QZSS) ha desarrollado un sistema de autenticación de mensajes de navegación. QZSS junto con GPS proporciona una alta disponibilidad del servicio de posicionamiento incluso en zonas metropolitanas densamente pobladas. Además, puede aplicarse a diferentes sistemas que puede utilizarse para dar soporte de autenticación a GPS, Multi-functional Satellite Augmentation System (MSAS), European Geostationary Navigation Overlay Service (EGNOS) y GPS Aided Geo Augmented Navigation (GAGAN).

QZSS comparte frecuencias de transmisión de señales de uso civil con GPS, como son las bandas L1, L2 y L5. Además, transmite la señal experimental LEX en la misma banda que GPS transmite L6.

La señal sobre la que se basa el funcionamiento del sistema de autenticación es L1SAIF y se utiliza para distribuir datos de corrección a los usuarios GNSS para conseguir una precisión de posicionamiento de menos de un metro en las zonas próximas a Japón. Esta señal se emite en la misma frecuencia que L1C/A y es compatible con la estructura de señales *Satellite Based Augmentation System* (SBAS). Su funcionalidad es la de transmitir datos de firmas digitales para la autenticación de señales de QZSS y GPS.

Metodología de autenticación.

Los pasos de autenticación se definen en [15]. Los datos de firma digital se transmiten dentro del mensaje de navegación de la señal L1SAIF perteneciente a QZSS. Estos datos se identifican mediante un *Message ID* que indica con qué mensaje de navegación están relacionados.

Generación y retransmisión de los datos de firma.

Por otro lado, a partir del mensaje de navegación transmitido en la señal L1C/A se genera el *Reference Authentication Navigation Data* (RAND). Se puede utilizar la señal L1C/A de GPS o de QZSS.
Algunos de los campos que forman el RAND son el *flag* AS, que estará activo si el sistema de autenticación está activado, el *Time of Week* (TOW), y el PRN ID, que define el satélite del que proviene el mensaje. Estos dos últimos campos son importantes puesto que concretan el mensaje a verificar mediante las claves de autenticación pertinentes.
Puesto que el TOW cambia cada 6 segundos y se utiliza para generar el RAND, también el RAND se modifica en el mismo periodo. Una vez se ha generado, el tamaño de RAND es de 80 bits.

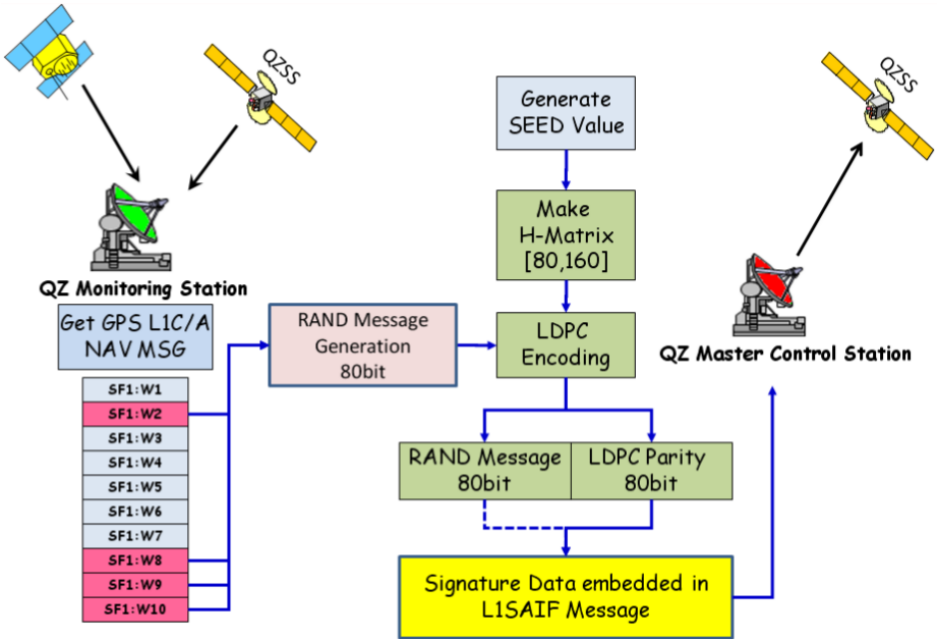

Ilustración 16. Generación de mensaje de autenticación en QZSS. [15]

Después el RAND se codifica mediante la técnica de *Low Density Parity Check* (LDPC). Para ello necesita una H-matrix, creada a partir de unos y ceros distribuidos de forma aleatoria. El RAND codificado pasa entonces a estar compuesto de 160 bits, a los cuales se le añade información relacionada con los datos de firmado y el *Public Key Infrastructure* (PKI). Finalmente se obtiene una firma digital con un tamaño de 212 bits. Cada 6 segundos, se generan datos de firmado en el *Authentication Data Center* (ADC). Estos se envían desde el centro de control de QZSS hasta los propios satélites QZSS, los cuales son capaces de retransmitirlos empaquetados dentro de la señal L1SAIF.

Proceso de autenticación de mensajes en el receptor

Para autenticar los mensajes de navegación, el receptor hace uso de las señales L1C/A de QZSS/GPS y de la señal L1SAIF de QZSS.

La señal L1C/A le permite calcular los datos de posicionamiento. Si se requiere la autenticación de los mismos, el receptor necesita obtener datos adicionales a partir del ADC. Estos datos son el valor de SEED, la matriz H-matrix o los datos de firmado con las claves para desencriptarlos. La ventaja de este algoritmo es que no necesita de información relacionada con la autenticación puesto que los datos de posicionamiento no están encriptados [15].

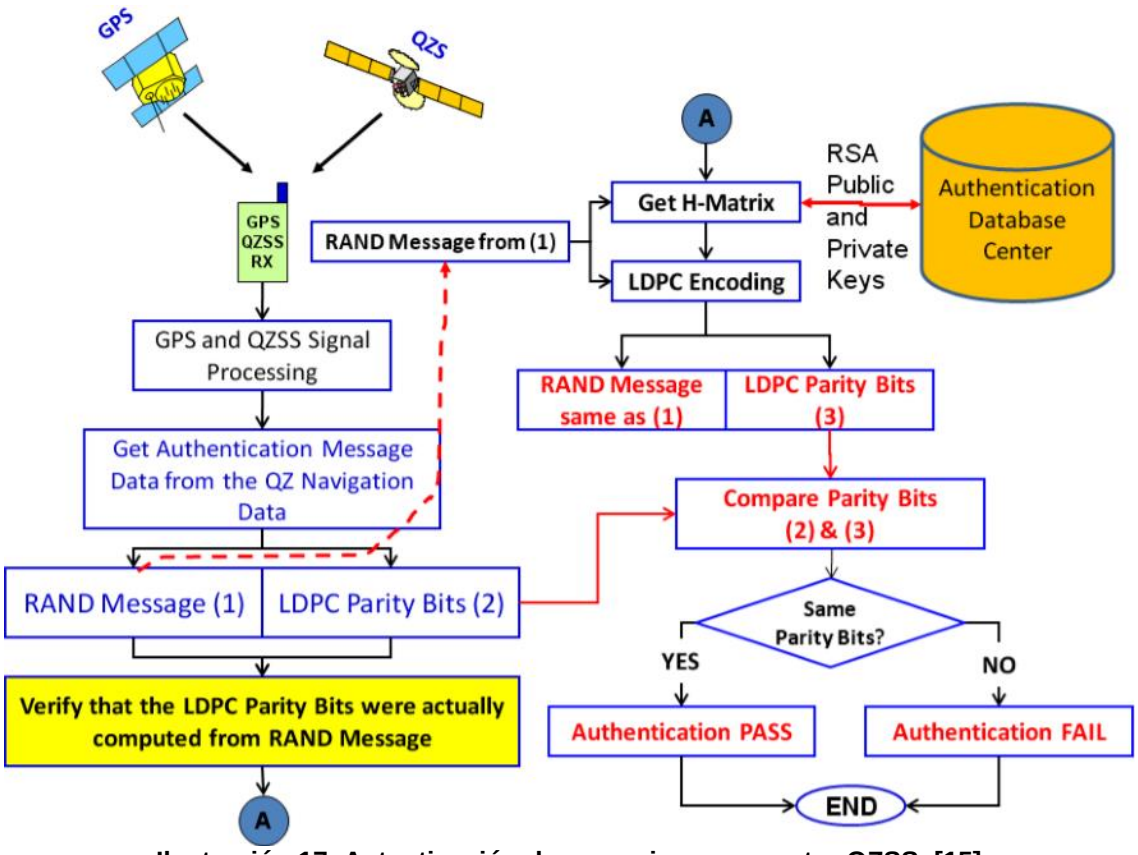

Ilustración 17. Autenticación de mensajes en receptor QZSS. [15]

El receptor recibe la señal L1SAIF y la decodifica. De estos datos se extraen el *Message ID* y los datos de firmado, los cuales se desglosan en RAND y los bits de paridad de LDPC.
A partir del *Message ID*, se realiza una petición al ADC para que envíe las claves RSA que referentes a ese mensaje. Una vez se han recibido, se utilizan para codificar el RAND que se ha recibido en L1SAIF. Si los bits de paridad LDPC obtenidos mediante esta comprobación son iguales que los recibidos, entonces la autenticación del mensaje es satisfactoria y se confirma que el mensaje de navegación es auténtico.

Beneficios obtenidos en el proceso de autenticación.

En [15] se hace mención a otros sistemas de autenticación basados en encriptación de mensajes o codificación del PRN que necesitan hacer modificaciones en la estructura de la señal para su correcta implementación, mientras que la implementación de esta metodología solo requiere modificar la generación del mensaje L1SAIF.
Al utilizar las señales L1C/A presentes tanto en QZSS como en GPS, puede usarse en ambos sistemas. Además, la señal L1SAIF puede modificarse definiendo un nuevo *Message ID*.
El receptor sólo necesitará de conexión con el ADC y acceso a sus datos en el caso de que la autenticación de mensajes esté activada.

4. Técnicas de autenticación por encriptación de código de ensanchamiento

GPS

Se han desarrollado diversos métodos de autenticación de señales GNSS para aumentar la defensa ante ataques de *spoofing*. Algunos de ellos se basan en la observación de las características físicas de las señales o en el análisis de propiedades obtenidas en el receptor. También se han desarrollado métodos criptográficos como NMA para enlazar el mensaje transmitido con la fuente de información, de forma que el receptor tenga la certeza del origen fiable de la información y de que no ha sido alterada, obteniendo así la autenticación de los mensajes de navegación transmitidos. Todos estos sistemas ofrecen protección ante diversos frentes de ataque. El sistema de autenticación que se está implementando en el sistema GPS propone enlazar el mensaje de navegación con el código de ensanchamiento.

Metodología

El código de ensanchamiento está formado una serie de códigos pseudoaleatorios deterministas que no transportan información y cuya unidad básica es el chip. Este tipo de códigos permiten obtener buenas prestaciones en las estimaciones de distancia entre transmisor y receptor [3]. También ayuda al receptor a distinguir entre señales recibidas en la misma frecuencia por distintos satélites mediante *Code Division Multiple Access* (CDMA).
El periodo de chip es menor que el periodo del bit de navegación. Al realizar el producto del mensaje de navegación y el código de ensanchamiento se obtiene un periodo de bit menor que en la señal original, y por tanto un espectro mayor y ensanchado. Esto permite mejorar la robustez del sistema ante interferencias de banda estrecha puesto que la energía de la señal de distribuye en un espectro más amplio.

En [16] se propone el sistema *Chips-Message Robust Authentication* (Chimera) para autenticar de forma conjunta los datos de navegación y el código de ensanchamiento de las señales civiles GPS mediante el concepto de *time-binding* o enlazado temporal.
Se puede aplicar funciones criptográficas tanto al código de ensanchamiento como a la portadora. Los métodos basados en TESLA como OSNMA no autentican el tiempo de transmisión puesto que los bits de datos de navegación cambian a un ritmo demasiado lento. En el caso de los chips del código de ensanchamento, tienen una velocidad de variación mayor, lo que hace posible obtener una autenticación temporal más precisa. Por eso, para enlazar temporalmente el código de ensanchamiento y el de navegación se utiliza el término *marker* (marcador), y hace referencia a las propiedades de la señal utilizadas para realizar un enlace temporal. Es habitual que para realizar un enlace temporal se utilicen métodos de clave simétrica basados en la

compartición de una clave secreta entre las distintas partes que forman el sistema de comunicación.

Chimera es una técnica que extiende el concepto de NMA, combinando la autenticación de datos con métodos de autenticación temporal comprendidos dentro del código del código de ensanchamiento. Utiliza una firma digital para generar marcadores criptográficos que son diseminados en el interior del código de ensanchamiento. Una vez que el usuario posee una clave pública autenticada del sistema GPS, puede constatar que tanto el mensaje de navegación como el código de ensanchamiento proceden del mismo origen certificado.

Diseño

El protocolo de autenticación Chimera implementa características de seguridad que se hacen presentes tanto en los datos de navegación como en el código de ensanchamiento.

- Los datos del mensaje de navegación se protegen mediante la firma digital de sus *frames*. De esta forma, los mensajes de navegación se firman de forma unívoca.
- Los marcadores de autenticación sustituyen un conjunto de chips del código de ensanchamiento y se pueden utilizar para autenticar este código.

Para enlazar los datos de navegación con el código de ensanchamiento, se utilizan los marcadores criptográficos embebidos dentro del código de ensanchamiento. Estos marcadores son el núcleo del concepto Chimera. Se generan partiendo de la firma digital que se encuentra en el mensaje de navegación, en un proceso que comprende tanto criptografía simétrica como hashing criptográfico.

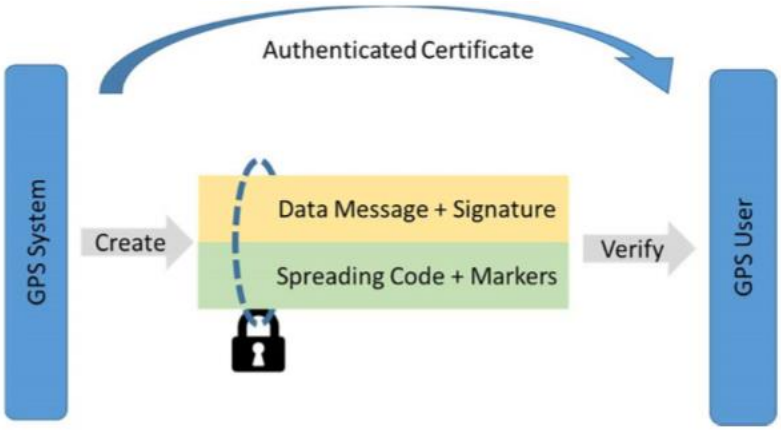

Ilustración 18. Time-Binding en Chimera. [16]

Como aproximación básica al proceso de posicionamiento de marcadores, en [16] se habla del *puncturing* o punzado. Consiste en la introducción de marcadores dentro del código de ensanchamiento. La posición de los marcadores, sus valores y el *Duty-Cycle* (DC) o ciclo de trabajo de los marcadores dentro del código se determina antes de la transmisión.

En el momento de recepción de la señal, el usuario no conoce dónde se encuentran los marcadores dentro del código, ni tampoco sus valores. Esta es la etapa de *Bit-commitment* o esquema de compromiso. El usuario debe almacenar en memoria una cantidad de muestras del *Analog Digital Converter* (ADC). Una vez que se recibe la clave, puede generar una secuencia de referencia de los marcadores. Para comprobar la integridad de los datos recibidos, se realiza una correlación entre los datos almacenados y la secuencia de referencia. Si la correlación es alta, se toman los datos recibidos como válidos.

El diseño de la señal genérica de Chimera se basa en el uso de un certificado público/privado para generar una protección criptográfica tanto en los datos de navegación como en el código de ensanchamiento. Para ello se requiere que el equipo receptor obtenga los certificados públicos a través de una conexión ocasional con el PKI a través de canales de comunicación distintos a los canales GPS. Esto se realizará al menos una vez al año. El funcionamiento de Chimera varía en función de si el receptor cuenta con una conexión paralela a la conexión GPS de forma continua o no. Ambos sistemas no son excluyentes y pueden utilizarse de forma simultánea. A continuación, se definen los protocolos y sus diferencias.

Protocolo Slow-channel

Destinado a usuarios que sólo cuentan con conexión a través del sistema GPS. En el transmisor se utiliza el certificado privado para firmar digitalmente una parte de los datos del mensaje de navegación, como se hace en NMA. Se utiliza un algoritmo de hash seguro para convertir la firma digital en un hash, que será conocido como *marker key* o clave de marcador. Esta *marker key* se aplica al texto para generar dos secuencias separadas de texto cifrado: una para codificar dónde se encontrarán los marcadores y otra de se almacena qué valores tomarán.
Una vez el usuario ha recibido ambas partes de la firma digital, puede averiguar la localización y los valores de los marcadores para generar su propia secuencia de referencia de marcadores. Sin embargo, debido a la periodicidad de la transmisión de las firmas digitales, se podría realizar una denegación de autenticación mediante un ataque de colisión únicamente a la página que contiene la segunda parte de la firma digital. Esta vulnerabilidad sólo afecta al protocolo *slow-channel*.
Finalmente, se aplicará la función de correlación sobre esta secuencia de referencia y los datos recibidos que se han almacenado para comprobar que los datos recibidos proceden de una fuente fiable y son auténticos. Los usuarios que no formen parte del sistema Chimera ignorarán las firmas digitales y obtendrán los datos de mensaje de la forma que hacen habitualmente.

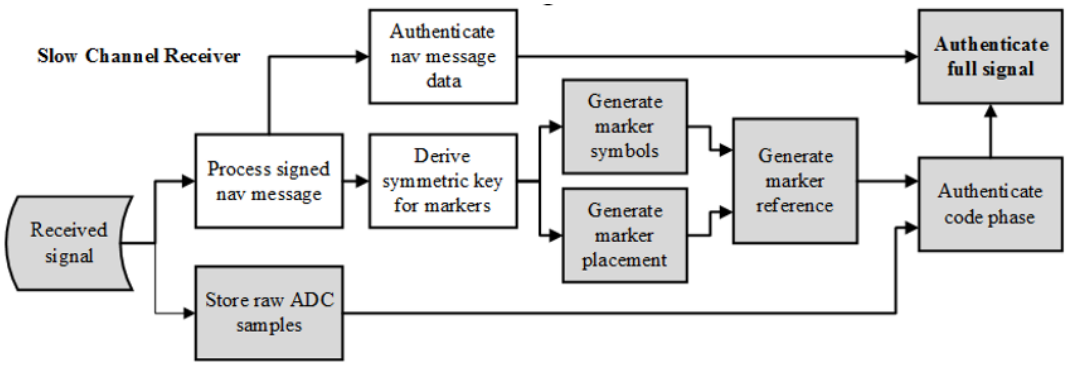

Ilustración 19. Recepción en receptor slow-channel mediante Chimera. [16]

Protocolo Fast-channel

Destinado a usuarios que cuentan con una conexión continua con otra red de comunicación *out-of-band* distinta a la del sistema GPS. En este protocolo, las *marker keys* se generan, se firman y se distribuyen a los usuarios a través de los canales *out-of-band.* Esto evita a los usuarios la necesidad de demodular los datos del mensaje de navegación para averiguar el posicionamiento y los valores de los marcadores, como se hace en *slow-channel*. También permite cambiar las *marker keys* más a menudo, haciendo este protocolo más robusto. También ofrece la posibilidad de que los datos de navegación se firmen y sean transmitidos a los usuarios mediante el canal *out-of-band.*

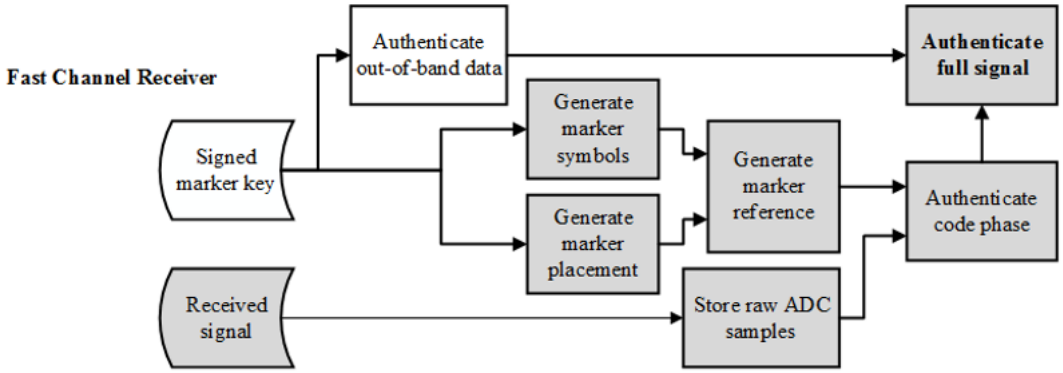

Ilustración 20. Recepción en receptor fast-channel mediante Chimera. [16]

La estructura Chimera Epoch representa el flujo de símbolos completo que es necesario para la transmisión de un conjunto de firma digitales y los marcadores generados a partir de ella. En el protocolo *slow channel*, un Chimera epoch hace referencia a una serie de *frames* necesarios para transmitir la firma digital, los datos firmados y los marcadores. En el caso del protocolo *fast channel*, un Chimera epoch se refiere al periodo de tiempo en el cual se hace uso de la misma *marker key*. Este último caso es independiente de epoch de slow channel, y su duración es igual al TBA, que está alrededor de 2 segundos.

Implementación sobre la señal GPS L1C

Chimera puede aplicarse a diferentes señales y sistemas. En el documento de *Interface Specification* (IS) [17] se definen las características de la implementación sobre la señal L1C de GPS.

Estructura y características de la señal

Chimera es capaz de autenticar tanto el mensaje de datos CNAV-2 como el código de ensanchamiento. La autenticación del mensaje de datos se realiza mediante la inserción de marcadores en el código de ensanchamiento de la señal L1C pilot. Un canal está formado por: datos autenticados, una fuente para obtener las claves autenticadas y una serie de marcadores dispersados a lo largo del código de ensanchamiento. Los marcadores para *slow-channel* se obtienen a partir del mensaje de navegación, mientras que en *fast-channel* se obtienen de una fuente *out-of-band*. Ambos tipos de marcadores se generan de forma independiente, conviven en el mismo código de ensanchamiento, y nunca ocupan la misma localización. En función de si el usuario utiliza un protocolo u otro, buscará los marcadores pertinentes en distintas posiciones.

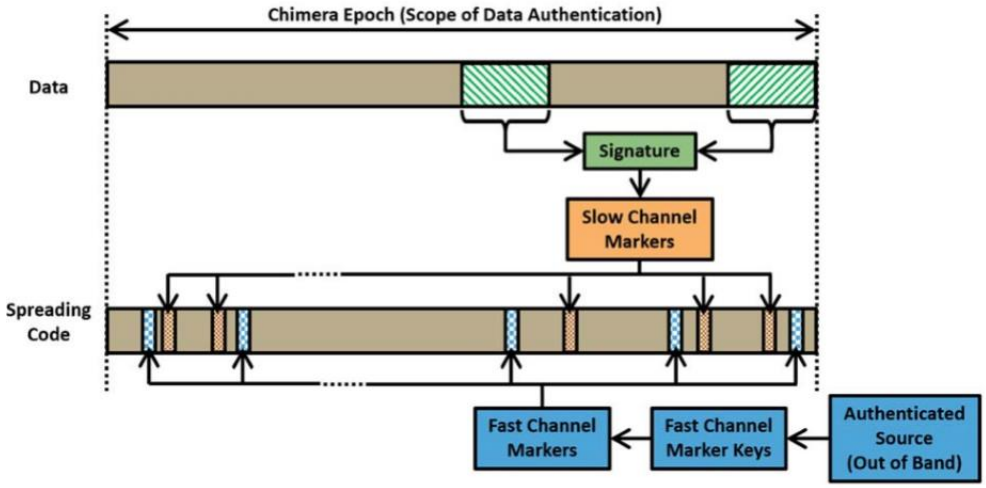

Ilustración 21. Enlazado de Datos de mensaje y Código de ensanchamiento en Chimera. [17]

Cada marcador que se inserta en el código de ensanchamiento de la señal L1C pilot es un símbolo modulado mediante *Binary Offset Carrier* (BOC), más concretamente BOC(1,1), y reemplaza al símbolo que se encontraba en esa posición.

Puesto que la posición de los marcadores es desconocida en el momento en que se recibe la señal, el sistema Chimera introduce una pérdida de correlación los receptores L1C [17]. El ciclo de trabajo de inserción de marcadores $DF_{markers}$ se define como el porcentaje del código de ensanchamiento que ha sido reemplazado por marcadores. Puesto que los protocolos *fast-channel* y *slow-channel* cuentan con su propio ciclo de trabajo independiente, $DF_{markers}$ se refiere a la suma del ciclo de trabajo debido a ambos protocolos. La pérdida de correlación L_{corr} debido a $DF_{markers}$ se cuantifica como:

$$L_{corr}(dB) = 20Log(1 - DF_{markers})$$

Para compensar la pérdida de correlación causada por la implantación de Chimera y alcanzar el nivel de potencia equivalente necesaria, el transmisor deberá aumentar la potencia media de transmisión del canal pilot de L1C.

Características de los marcadores

Los valores que toman los marcadores deberán no estar repetidos, haber sido obtenidos mediante derivación criptográfica y generados de forma independiente a los códigos de ensanchamiento de L1C. Los marcadores serán únicos para cada satélite o *Space Vehicle* (SV). Como se ha detallado previamente, el ciclo de trabajo de inserción de marcadores afecta a la pérdida de correlación. Existe la posibilidad de que el ciclo de trabajo sea nulo, obteniendo una implementación sin marcadores, de forma que se autentica los datos de navegación, pero no los códigos de ensanchamiento. Como se ha visto previamente en este Trabajo, a esta implementación se la conoce como NMA.

Definición del mensaje

El alcance de autenticación de datos para *slow-channel* y *fast-channel* de Chimera es el Chimera epoch. Su duración es de 3 minutos, comprende 10 mensajes L1C y comienza en un momento temporal que ha de ser un múltiplo de 18 segundos con respecto al inicio de un *GPS week.*

La firma digital que se utiliza en Chimera se genera a partir de la clave privada, perteneciente al par de claves público y privado. Aplicando sobre la clave privada el algoritmo de firmado por curvas elípticas *Elliptic Curve Digital Signature Algorithm* (ECDSA), se consigue una firma digital de 448 bits. Se emite una firma digital para cada Chimera epoch, dividida en dos partes, cada una de las cuales se inserta en un mensaje. Los primeros 202 bits se transmiten en la página 8 de la subtrama 3, mientras que los 236 restantes utilizan la página 9 de la misma subtrama para ser emitidos.

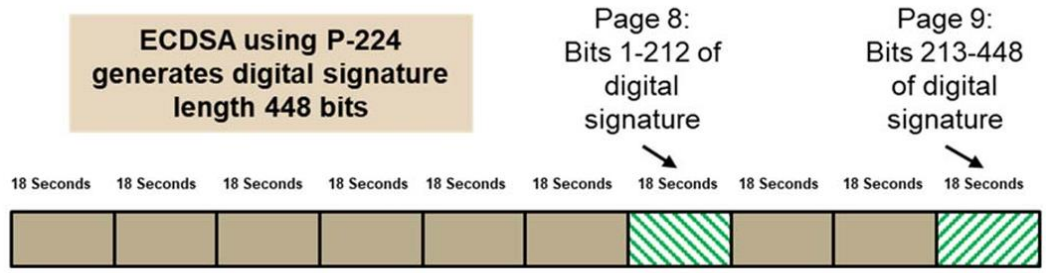

Ilustración 22. Estructura de Chimera epoch e inserción de firma digital. [17]

Generación de key markers

En los receptores que utilizan el protocolo *slow-channel* de Chimera, los *marker keys* se obtienen a partir de la firma digital emitida por el transmisor, mediante la aplicación de la función hash SHA-512 y su posterior truncado para quedarse con los 256 bits menos significativos.

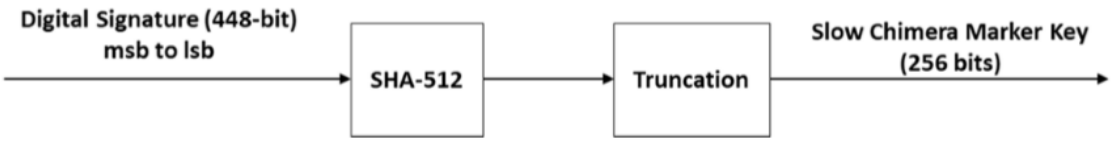

Ilustración 23. Generación de *marker key* en protocolo *slow-channel* de Chimera. [17]

En el caso del protocolo *fast-channel*, no se utiliza la firma digital. Para crear el *marker key*, se crea un HMAC SHA-512 a partir de la *Fast Channel Key Generation* Key obtenida *out-of-band* y el tiempo de inicio de periodo del *fast-channel*. Del mismo modo que en *slow-channel*, se trunca para crear un *marker key* de 256 bits.

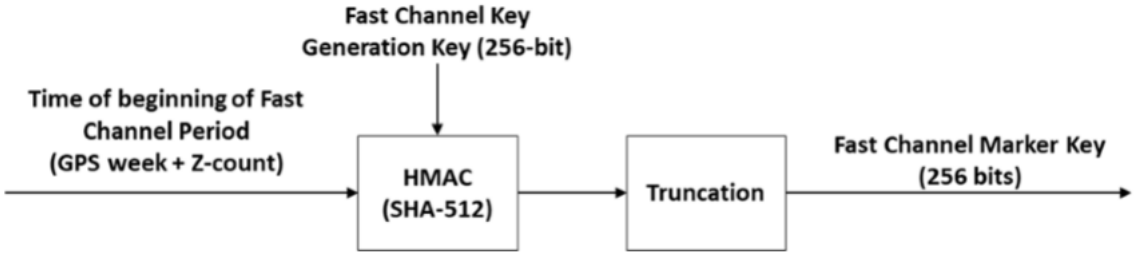

Ilustración 24. Generación de *marker key* en protocolo *fast-channel* de Chimera. [17]

Generación de marcadores

Una vez se obtiene un *marker key*, el receptor puede averiguar dónde se encuentran los marcadores dentro del código de ensanchamiento de la señal L1C pilot. Estarán situados dentro de los *marker segments*, de 33 chips de duración. Cada uno de estos *marker segments* están asignados al protocolo *slow-channel* o *fast-channel*, de forma que los marcadores pertenecientes a un protocolo no pisen a los del otro.

Dentro de un *marker segment*, los marcadores podrán situarse en 29 de los 33 chips que están modulados mediante BOC(1,1). Los cuatro chips restantes están modulados sobre BOC(6,1) y no se modifican. En la Ilustración 25 se puede ver que un sector se divide en 31 segmentos, los cuales se utilizarán para introducir marcadores de *slow-channel* o *fast-channel* en función del *sector pattern* que se obtenga del campo PRN.

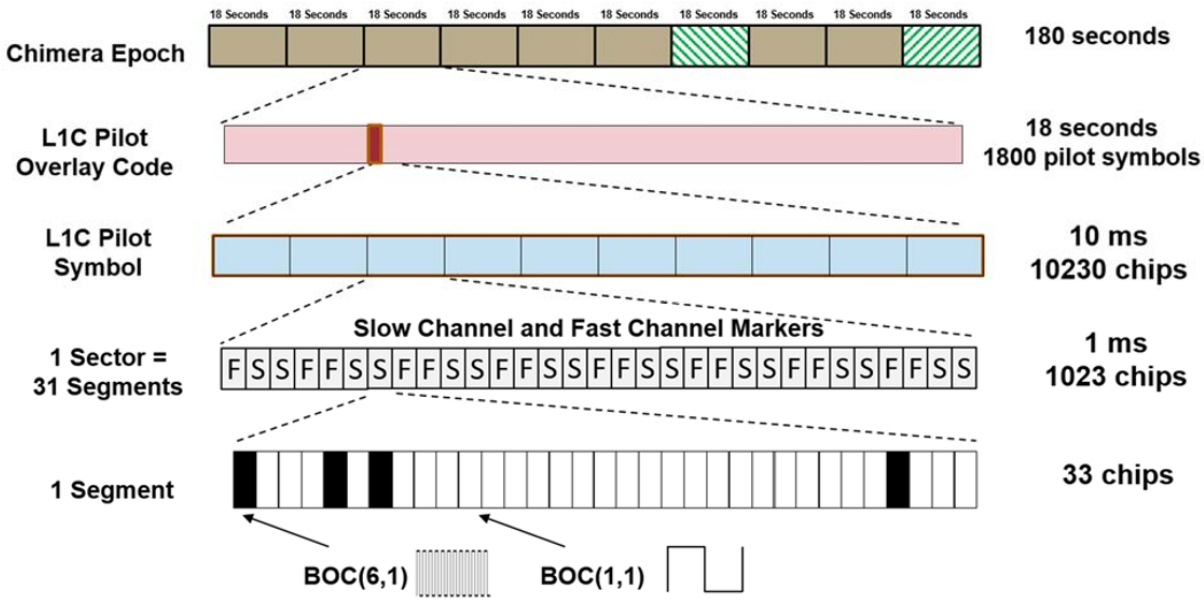

Ilustración 25. Estructura de Chimera Epoch y asignación de segmentos para marcadores. [17]

Los segmentos utilizados para marcadores en un sector de duración 1 ms dependen del ciclo de trabajo de inserción de marcadores. Como se ha visto previamente, este ciclo de trabajo es independiente para los protocolos *fast-channel* y *slow-channel*.

Para obtener la posición y el valor de los marcadores, el receptor aplica el algoritmo de hash AES-256 sobre el *marker key* y sobre cada uno de los bloques de 128 bit en texto plano que indican características propias de los marcadores. Dentro de estos bloques se encuentran campos que definen, entre otros, el identificador de la fuente, el enlace, y si se trata de marcadores para *fast-channel* o *slow-channel*. También contiene el campo PRN, que definirá cuál de los 31 *sector patterns* posibles tomarán los marcadores *fast-channel* o *slow-channel* dentro de un sector, recogidos en el Anexo 1.

En relación a la utilización en Chimera de una tabla con una cantidad limitada de *sector patterns* para decidir la posición de los marcadores, se han realizado análisis para comprobar si este diseño afecta a la seguridad del sistema. Se ha estudiado la utilización del esquema de la tabla con *sector patterns* en comparación con una distribución uniforme ideal de los marcadores, en términos de robustez ante ataques de colisión y de capacidad de averiguar la posición y valores de marcadores [18]. Se realizan diferentes tipos de ataques al sistema y se observa la probabilidad de éxito de los mismos, comparando el esquema de *sector patterns* de Chimera con el de una distribución uniforme de los marcadores.

Como es de esperar, si el atacante conoce la posición de uno de los marcadores, la probabilidad de que un ataque de colisión tenga éxito es mayor a la obtenida cuando el atacante no tiene ninguna información previa sobre los marcadores. Sin embargo, la probabilidad de un ataque de colisión en la distribución ideal uniforme es muy parecida a la obtenida en el esquema utilizado en Chimera, por

lo que la utilización de una tabla de *sector patterns* delimitada no afecta a la seguridad del sistema de forma significativa. Aunque el atacante conozca previamente la tabla incluida en el Anexo 1, no será capaz de utilizarla para obtener una ventaja en la estimación *on-the-run* de las posiciones y valores de los marcadores.

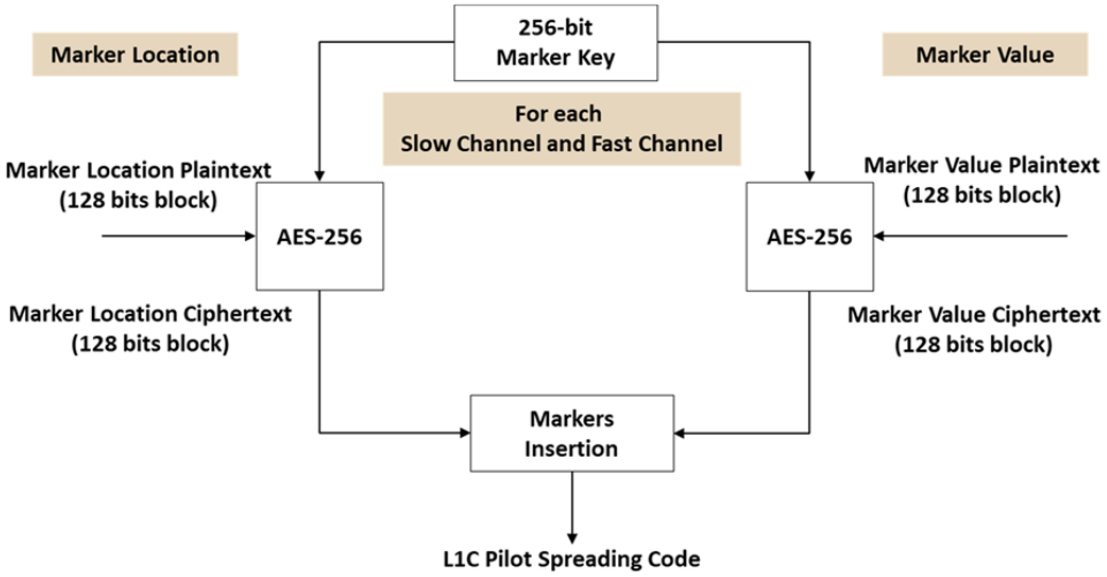

Ilustración 26. Obtención de posición y valor de los marcadores a partir de un marker key. [17]

Finalmente, a partir del texto cifrado o *Ciphertext* se obtienen los valores y las posiciones que tomarán los marcadores dentro del código de ensanchamiento de la señal L1C pilot.

5. Implementación de una prueba de concepto de Galileo OSNMA

Estudio del sistema y acotación del alcance

Con el fin de realizar un estudio del funcionamiento de sistema de autenticación para Galileo OSNMA, se ha realizado una prueba de concepto simplificada del mismo.

En este caso la implementación busca comprobar el correcto funcionamiento del sistema de generación de una cadena de claves. Para ello se ha generado una secuencia de bits aleatoria, a la cual se le aplica la función hash para la generación de la cadena de claves. A partir de este momento se obtiene la primera clave de la cadena, que será la salida de la función hash aplicada.
A esta clave obtenida se le vuelve a aplicar la función hash, generando así la segunda clave de la cadena. Esta operación se vuelve a repetir un número definido de veces hasta obtener una cadena de claves.

Cada una de estas claves se utilizará para codificar un mensaje de navegación. En este caso, en lugar de codificar un mensaje de navegación completo, por simplicidad se ha optado por la codificación de una cadena de caracteres.

Para la codificación se aplica una función HMAC, la cual se aplica sobre un par de elementos: clave y mensaje. De esta forma, a partir de $Clave_i$ y de $Mensaje_i$, se obtiene como resultado MAC_i. Este método se repite con todos los mensajes generados, de forma que a cada par de $Mensaje_i$ y $Clave_i$ le corresponde una MAC_i.

Como se ha visto en el apartado de OSNMA, una vez generada la cadena de claves y se han obtenido los MACs, el transmisor está listo para comenzar a enviar los paquetes.

La diferencia entre el paquete transmitido que contiene un mensaje y el paquete que contiene la clave para ese mensaje será 1, de forma que $d = 1$. Los paquetes que se transmiten están formados por MAC_i, $Mensaje_i$ y $Clave_{i-1}$. De esta forma, cuando el usuario recibe el $Paquete_i$, no recibe la $Clave_i$ correspondiente al mensaje recibido en ese mismo paquete, sino la $Clave_{i-1}$ correspondiente al mensaje $Mensaje_{i-1}$, incluido en el paquete recibido anteriormente, $Paquete_{i-1}$. El usuario no podrá autenticar $Mensaje_i$ hasta recibir el próximo paquete $Paquete_{i+1}$, que contendrá la $Clave_i$.

Una vez recibido $Mensaje_i$, MAC_i, y un paquete después, $Clave_i$, el usuario podrá comprobar la autenticidad del $Paquete_i$. Para ello, utilizará el par de elementos recibidos $Clave_i$ y $Mensaje_i$ para aplicar la misma función MAC, generando su propio $*MAC_i$ a partir de elementos recibidos. En el caso de que el MAC recibido MAC_i sea el mismo que el MAC generado en el usuario $*MAC_i$, el $Paquete_i$ ha sido autenticado, y por tanto también el $Mensaje_i$.

El canal de comunicación a través del cual se transmiten los paquetes puede generar alteraciones en el mensaje transmitido o incluir la presencia de un *spoofer*, cuya finalidad es transmitir un mensaje de navegación falseado para que el usuario obtenga un posicionamiento erróneo. Para simular este entorno se ha optado por modificar los mensajes incluidos dentro de los paquetes recibidos, de forma análoga a como lo haría un atacante o un canal que modifica el mensaje que se transmite a través suyo. De esta forma, si el mensaje recibido no es el mismo que el mensaje transmitido original que se utilizó para generar el MAC original, el MAC reconstruido en el receptor será distinto al recibido, y no se podrá autenticar el mensaje.

Debido al alcance del proyecto y a la decisión de realizar una versión simplificada de la implementación, no se hará uso de una entidad de certificación PKI ni del uso de firmas digitales para verificar claves, como sí se hace en el sistema OSNMA implementado en Galileo. En este caso, los mensajes serán verificados mediante la comprobación de MACs, claves y mensajes que se ha detallado.

Diseño de los bloques desarrollados

El primer bloque funcional será el bloque generador de claves. Se genera una secuencia de 8 bits aleatoria, a la cual se le aplicará la función hash de generación de claves. Se opta por utilizar una función hash que aparece en la especificación de OSNMA [10] como una de las posibilidades a elegir en las implementaciones del sistema: SHA-256. De esta forma, cada una de las claves generadas estará formada por 256 bits. En esta prueba de concepto se opta por no truncar la salida de la función hash debido a que no hay necesidad de ello por contar con capacidad de cómputo suficiente y tratarse de una implementación simplificada. Por tanto, se tomará el mayor tamaño de clave que se permite en la especificación de OSNMA [10].

El segundo bloque será el bloque generador de mensajes de navegación transmitidos. Se crearán una serie de variables que contienen una cadena de caracteres. El contenido de los mensajes será distinto en cada uno de ellos, y no podrá ser modificado por el usuario, aunque sí podrá conocer es el contenido de los mismos.

El tercer bloque será el bloque generador de MACs. Aplicando la función MAC a una pareja formada por una clave y un mensaje generados en los bloques anteriores, se obtiene una MAC inequívoca que identifica ambos elementos generadores de MAC. Se ha optado por utilizar una de las funciones MAC presentes en la especificación del sistema OSNMA [10], HMAC-SHA-256. La salida de la función MAC no se truncará, obteniendo una longitud de MAC de 256 bits.

El cuarto bloque será el generador de mensajes recibidos. El usuario podrá saber cuál es el mensaje que se generó en el transmisor, y definir cuál es el mensaje que finalmente se ha recibido. De esta forma, el usuario decide si en el canal existen perturbaciones que modifiquen el mensaje durante su transmisión o si se da la presencia de un atacante que envía un mensaje falseado. Si el usuario genera un mensaje igual al que se ha transmitido, el canal simulado no ofrece perturbaciones y el receptor podrá autenticar el mensaje recibido de forma

satisfactoria y se dará por bueno. En el caso de que el usuario modifique el mensaje recibido, el receptor no podrá autenticar el mismo y se tomará el mensaje como alterado o no genuino.

El quinto bloque será el generador de MACs de referencia. Se realiza utilizando los paquetes recibidos, dentro de los cuales estarán los mensajes recibidos, hayan sido modificados por el usuario o no. El receptor generará un nuevo MAC con ellos y con la clave recibida.

El sexto bloque será el comparador de MACs. El receptor tratará de autenticar los mensajes mediante la comparación del MAC recibido y del MAC generado en el receptor utilizando el mensaje de navegación recibido. Si ambos MACS coinciden, entonces tanto la clave como el mensaje de navegación coinciden en ambos casos, y el mensaje se dará por autenticado, dando por bueno su contenido.

Diseño del algoritmo

A continuación, se detalla el diagrama donde se representan los bloques funcionales, los flujos de datos generados y la aplicación de funciones a las mismas.

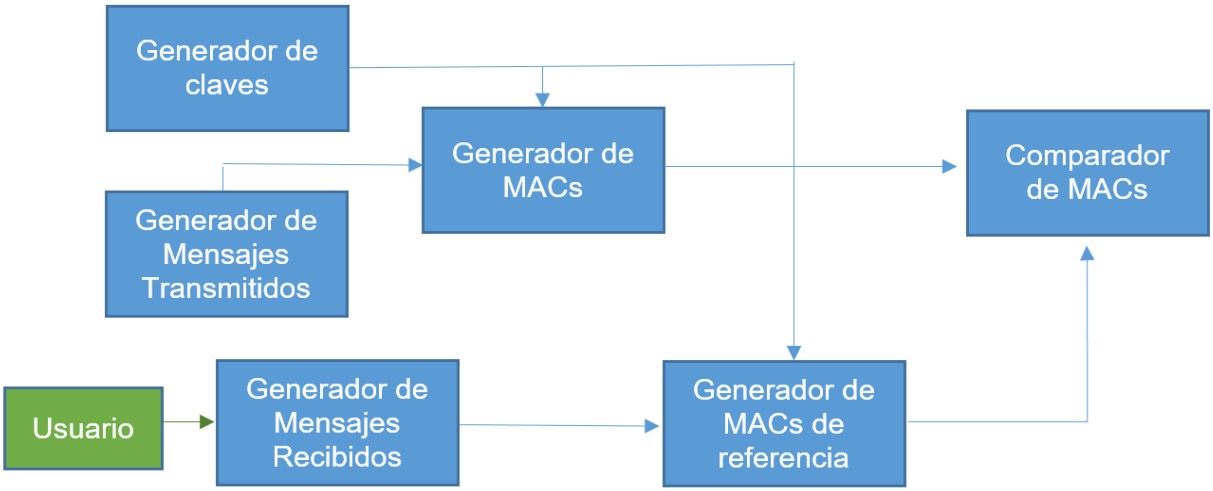

Ilustración 27. Diagrama de bloques del software desarrollado.

Herramientas utilizadas

Generador de seed para las claves

Se utiliza la clase randi() [19] de Matlab para generar un número aleatorio de 8 bits. Se introduce el rango entre 1 y 256.

Generador de SHA256: SHA256Managed Class

La clase SHA256Managed [20] pertenece al espacio de nombres System.Security.Cryptography [21], la cual ofrece diferentes servicios criptográficos. Permite generar un hash a partir del valor de entrada introducido.

La función SHA256 genera un hash de 256 bits de longitud. Puesto que Matlab no permite una variable de tal longitud, el hash se divide en 32 Bytes.

HMAC Hash Message Authentication Code Function

Se trata de una función generada en Matlab (HMAC.m) [22] para calcular el HMAC. Permite utilizar cuatro funciones hash distintas. En esta implementación se utiliza SHA256 ya que se trata de una de las funciones permitidas en las especificación de OSNMA [10]. Son necesarias tres entradas para utilizar la función:

- Clave: Clave secreta introducida, de tipo char.
- Mensaje: Mensaje de entrada, de tipo char.
- Método: Función hash a utilizar. Pueden ser:
  'SHA-1', 'SHA-256', 'SHA-384', 'SHA-512'

Isequal function

Función Matlab [23]. Compara elementos para comparar si son iguales o no. Si son iguales, la función devuelve 1 (true), mientras que en el caso de que los elementos introducidos no sean iguales, devuelve 0 (false).

Bloques desarrollados

A continuación, se detalla una parte significativa de cada uno de los bloques de código que se han desarrollado en la implementación. La finalidad de este apartado es facilitar la comprensión de la funcionalidad del código.

Bloque generador de claves

```
seed=randi(2^8,1); %Genera el seed aleatorio de 8 bits

sha256hasher = System.Security.Cryptography.SHA256Managed;
sha256_5 = uint8(sha256hasher.ComputeHash(seed));
sha256_4 =
uint8(sha256hasher.ComputeHash(uint8(sha256_5)));
```

Bloque generador de mensajes transmitidos

```
mensaje_1 = 'Mensaje1';
mensaje_2 = 'Mensaje2';
```

Bloque generador de MACS

```
hmac_1 = HMAC(sha256_1,mensaje_1,'SHA-256');
hmac_2 = HMAC(sha256_2,mensaje_2,'SHA-256');
hmac_3 = HMAC(sha256_3,mensaje_3,'SHA-256');
hmac_4 = HMAC(sha256_4,mensaje_4,'SHA-256');
hmac_5 = HMAC(sha256_5,mensaje_5,'SHA-256');
```

Bloque generador de mensajes recibidos

```
% Paquete 1 recibido
% Generador de Mensaje1 recibido
p1_m = input('Ingrese un Mensaje1 recibido
(enviado:Mensaje1): ','s');
p1_hmac = hmac_1;
p1_key = sha256_0;

% Paquete 2 recibido
% Generador de Mensaje2 recibido
p2_m = input('Ingrese un Mensaje2 recibido
(enviado:Mensaje2): ','s');
p2_hmac = hmac_2;
p2_key = sha256_1;
```

Bloque generador de MACS de referencia

```
% Generador de MAC de referencia de Mensaje 1
hmac_check_1 = HMAC(p2_key,p1_m,'SHA-256');
% Generador de MAC de referencia de Mensaje 2
hmac_check_2 = HMAC(p3_key,p2_m,'SHA-256');
```

Bloque generador de MACS de referencia

```
%Comparacion de MAC de Mensaje 1 transmitido y MAC de
referencia de Paquete 1
a_1 = isequal(hmac_check_1, p1_hmac);

%Comparacion de Mensaje 1 transmitido y Mensaje 1 recibido
b_1 = isequal(p1_m, mensaje_1);

%Comparacion de Clave 1 transmitida y Clave 1 recibida
c_1 = isequal(p2_key, sha256_1);



%Comparacion de MAC de Mensaje 2 transmitido y MAC de
referencia de Paquete
%2
a_2 = isequal(hmac_check_2, p2_hmac);

%Comparacion de Mensaje 2 transmitido y Mensaje 2 recibido
b_2 = isequal(p2_m, mensaje_2);

%Comparacion de Clave 2 transmitida y Clave 2 recibida
c_2 = isequal(p3_key, sha256_2);
```

Requisitos para la ejecución del software implementado.

El código desarrollado está escrito en Matlab, y puede ejecutarse mediante la herramienta del mismo nombre. La mayoría de las funciones y estructuras utilizadas están incluidas dentro del paquete base de Matlab. Sin embargo, existen algunas funciones y librerías que será necesario incluir dentro del path de ejecución de la herramienta.

En el código implementado, aparece esta línea:

```
%addpath ('C:\Ruta\de\Import')
```

Será necesario descomentar la línea de código e incluir los siguientes archivos dentro de la ruta que aparece en la línea comentada.

HMAC.m

La función permite realizar funciones HMAC mediante la inclusión de un mensaje y la clave para generarla. Puede descargarse del repositorio Mathworks del autor. [22]

DataHash.m

La función permite realizar funciones hash SHA o MD5 de diversas estructuras en Matlab. Puede descargarse del repositorio Mathworks del autor. [24]

6. Comparativa entre OSNMA de Galileo y Chimera de GPS.

OSNMA

Se trata de un esquema de autenticación NMA basado en la encriptación del mensaje de navegación de Galileo. Está basado en el protocolo de autenticación TESLA, utilizado en diversos ámbitos tanto de Information Technology (IT) como de Operational Technology (OT).

La implementación del sistema OSNMA en Galileo ofrece una serie de **ventajas**.

- Ofrece un rendimiento similar a usuarios que utilicen autenticación de datos que a los usuarios de Galileo que no utilizan OSNMA.

- Maximiza la disponibilidad y la robustez en la señal Open-Service E1b, aprovechando el campo 'Reserved 1' de la estructura de los mensajes I/NAV, mediante una implementación retrocompatible que implica un nivel de cambios mínimo en la estructura desplegada.

- La autenticación se hace únicamente en base al mensaje de navegación y no a nivel de código de ensanchamiento, lo que reduce los requisitos de computación necesarios. Estos requisitos son aceptables en la gran mayoría de sistemas receptores desplegados en la actualidad. Además, permite la adecuación a las necesidades de cada caso de uso mediante la variación de parámetros como la longitud de claves y MACS, la adición de datos aleatorios 'salt' a las funciones hash o la elección de diferentes funciones hash y HMAC.

- La utilización de una misma cadena de claves para todos los satélites desplegados evita que un *spoofer* pueda obtener la clave de un satélite a partir de la clave generada por otro satélite. Además, al utilizar una única cadena se reduce la complejidad del receptor al disminuir la capacidad de cómputo necesaria en el mismo.

- El uso de autenticación cruzada entre satélites permite la implementación del sistema OSNMA a corto plazo en la generación de satélites desplegados en la actualidad, incluyendo aquellos que no tienen conectividad con tierra.

- Permite la implementación de una metodología para evitar ataques de repetición basada en la inclusión de símbolos impredecibles y el posicionamiento de los mismos dentro de la estructura de mensaje transmitido.

Dentro de las limitaciones, puesto que OSNMA está basado en TESLA, también se ve afectado por los requisitos de seguridad de este algoritmo, analizados

previamente en este Trabajo. Se describen algunas **vulnerabilidades criptográficas** del sistema.

- El nivel de sincronización entre emisor y receptor está relacionado con el tiempo de espera entre la publicación de claves. Para reducir la ventana de ataque durante el cual el sistema es vulnerable es necesario acotar el tiempo de espera de recepción de claves.

- Necesidad de un método de autenticación por clave simétrica basado en un algoritmo robusto para evitar que un atacante pueda generar una cadena de claves falsa y firmarla como auténtica.

- Al utilizar una única cadena de claves para todos los satélites, se reduce el número de bits impredecibles. Esto afecta a la seguridad que proporciona el sistema ante ataques de repetición, disminuyendo el número de bits impredecibles para implementar técnicas de defensa *anti-replay*.

- Mediante ataques de fuerza bruza basados en la computación se podría obtener la cadena de claves. Para aumentar la dureza criptográfica ante ataques de pre-computación se puede optar por añadir información conocida por el receptor al proceso de creación de hashes, como la variable *Galileo System Time* (GST).

- El sistema de autenticación cruzada entre satélites permite que los datos de navegación de los satélites no conectados al segmento de tierra se transmitan en texto plano para que los satélites sí conectados los vuelvan a transmitir de forma segura. Existe la posibilidad de que estos mensajes no autenticados sean víctimas de un ataque de repetición o *replay-attack*.

RESUMEN

OSNMA está basado en el protocolo TESLA, modificado para utilizar una única cadena de claves para todos los satélites. Esto permite aumentar la disponibilidad, a la vez que aumenta la robustez ante ataques y pérdida de datos. Las prestaciones obtenidas con OSNMA son muy similares a las obtenidas sin método de autenticación. Al contar OSNMA con una gran cantidad de bits impredecibles, hace menos efectivos los ataques basados en repetición y los de estimación de código. Por otro lado, la autenticación cruzada reduce la complejidad necesaria en los receptores, a la vez que permite que todos los satélites desplegados en la actualidad puedan autenticar sus mensajes, incluidos aquellos que no tienen conectividad con el segmento de tierra.

CHIMERA

La principal característica diferenciadora de este sistema para GPS es la utilización de un esquema de enlazado de tiempo o *time-binding* para enlazar el mensaje de navegación y el código de ensanchamiento. Esto lo hace muy robusto ante algunos métodos de ataque a cambio de aumentar su complejidad.

La implementación de Chimera sobre una señal conlleva una serie de ventajas:

- Puesto que se basa en la encriptación del código de ensanchamiento, consigue un mayor de protección frente a ataques de estimación o pre-computación que los sistemas que se basan únicamente en autenticación de mensaje.

- Tiene una alta probabilidad de detectar si el sistema se encuentra bajo un ataque de spoofing de una señal no genuina, siempre que se haya pasado a la fase de seguimiento.

- Mejora la robustez de la señal L1C de GPS, utilizando dos páginas de la subtrama 3 del mensaje de datos CNAV-2 para transmitir las firmas digitales.

- Cuenta con dos protocolos distintos que aplican a diferentes tipos de usuarios: aquellos que tienen acceso a una conexión *out-of-band* utilizan el llamado *fast-channel*, mientras que aquellos que únicamente cuentan con la conexión con el sistema GPS harán uso del protocolo *slow-channel*. Estos protocolos pueden convivir en una misma señal de forma simultánea.

- Permite la adaptación a distintas necesidades y campos de aplicación mediante la configuración de características propias del esquema, como el DC de inserción de marcadores. Además, se puede especificar el número de marcadores de forma independiente para *slow-channel* y *fast-channel*.

- Debido a que la inserción de marcadores en el código de ensanchamiento se realiza mediante una deriva criptográfica a nivel de chip, está diseñado para disminuir la probabilidad de éxito de ataques de Denial of Service (DoS) o denegación de servicio, o de modulación de potencia. Además, ataques como SCER que se basan en la estimación de datos futuros no son efectivos ante un sistema que utiliza LDPC en las firmas digitales para el protocolo *fast-channel*.

Por otro lado, aplicar el método Chimera en una señal implica algunas limitaciones:

- El sistema de autenticación Chimera se utiliza una vez el usuario ha pasado a la fase de seguimiento de señal, en un momento en que conoce

Doppler y la fase de código. Por tanto, no es capaz de prevenir que un usuario se enganche a una señal que ha sido comprometida.

- Para el correcto funcionamiento del esquema es necesario desplegar un sistema PKI de distribución de claves y certificados. La implementación sobre una señal conlleva que todos los receptores realicen una conexión al menos una vez al año, para obtener el par de claves públicas y privadas del PKI. Esto les permitirá hacer uso de las firmas digitales en las que se basa el posicionamiento de marcadores.

- Para utilizar el protocolo *fast-channel*, el receptor necesita contar de forma continua con un canal de comunicación *out-of-band*, alternativo al sistema GPS.

- El protocolo *slow-channel* es susceptible de recibir un ataque de denegación de autenticación si se busca realizar una colisión de datos contra la segunda parte de la firma digital, debido a la periodicidad de la misma y a que siempre se encuentra en la misma página de la subtrama 3.

- La localización de los marcadores es desconocida para el usuario hasta que haya recibido la *marker key*. Ambos tipos de protocolos requieren que el usuario no tenga acceso a las *marker generation keys* hasta que sean publicadas, esto es, hasta el final del Chimera epoch. Por esto, el receptor deberá almacenar datos *raw* para autenticarlos más adelante, cuando obtenga las *marker keys*. Esto conlleva unos requerimientos de capacidad de computación y memoria de almacenamiento mayores que el resto de sistemas de autenticación. Para la implementación sobre la señal L1C, la capacidad de almacenamiento requerida en un receptor será de alrededor de 2.50 Mbytes por cada segundo de seguimiento de la señal que implemente el sistema Chimera.

- Los requerimientos técnicos que aplican al receptor relacionados con la capacidad de computación y de almacenamiento hacen a Chimera un sistema difícil de implementar en una gran cantidad de sistemas embebidos de bajo consumo como dispositivos wearables o redes de sensores inalámbricas.

- La implementación de Chimera en un sistema de transmisión conlleva pérdidas por correlación en la recepción de la señal que se está siguiendo. Estas pérdidas están relacionadas con el número de marcadores introducidos o *Duty-Cycle* de inserción. Para mantener el nivel de correlación esperado y no perder prestaciones en el posicionamiento y la autenticación, será necesario aumentar la potencia de transmisión en los sistemas de emisión de señales. Esto implica un aumento de la energía necesaria en el funcionamiento de los sistemas de transmisión de señales que implementen Chimera como sistema de autenticación, así como un aumento en los gastos de operación *OpEx* de los mismos.

RESUMEN

El sistema de autenticación Chimera Ofrece una gran fiabilidad ante ataques de estimación y posterior emisión de posicionamiento falseado. Sin embargo, no permite por sí mismo proteger al usuario ante otro tipo de ataques como los basados en de repetición o *Meaconing*. Por tanto, por sí sólo no es el sistema definitivo de defensa ante ataques de *spoofing*, por lo que sería conveniente utilizando en combinación con otros métodos de defensa como los RPM para detección de anomalías en el nivel de potencia de la señal.

Su implementación sobre una señal conlleva un aumento en la complejidad del sistema y de los receptores, lo cual lo hace inviable en algunos de los sistemas de bajo consumo que existen en la actualidad.

7. Conclusiones

Objetivos y alcance del proyecto.

En este Trabajo de Fin de Máster se ha analizado el fenómeno del *spoofing* contra las señales que forman los diferentes sistemas GNSS. Se enumeran las principales técnicas de ataque que se utilizan para falsear señales y hacerlas pasar por auténticas, en relación con los sistemas necesarios para llevarlos a cabo. Además, se tratan los diversos métodos de defensa disponibles para hacer frente a estos ataques, especificando las condiciones y las técnicas de ataque ante los que son más efectivos.

Las técnicas criptográficas son uno de los métodos de defensa avanzada más utilizados. Puede optarse por la encriptación de diversos elementos que conforman el canal de comunicación del sistema GNSS. El método de autenticación del mensaje de navegación (NMA) tiene como objetivo asegurar al usuario que el posicionamiento que está obteniendo proviene de una señal auténtica, transmitida por un satélite que pertenece a la constelación del GNSS en cuestión, además de que no ha sido modificado en ningún paso seguido en su transmisión.

El sistema OSNMA busca autenticar las señales utilizadas por los servicios *Open-Service* de Galileo. Para ello se utiliza un algoritmo basado en TESLA que genera y distribuye una cadena de claves común entre todos los satélites disponibles, mediante un sistema de *cross-authentication*. Esto aporta robustez al sistema, a la vez que permite que el sistema se pueda implementar actualmente en todos los satélites que ya han sido desplegados. También es necesario implementar un sistema de distribución de certificados PKI para la utilización de firmas digitales. Puesto que la especificación de OSNMA se ha realizado buscando que los datos necesarios para la autenticación están contenidos dentro de la estructura actual del mensaje de navegación de Galileo, no necesita de una modificación en el hardware del sistema.

Se ha analizado el sistema de autenticación diseñado para el sistema QZSS. Este sistema utiliza señales en las mismas bandas de frecuencia que el sistema GPS, de forma que puede aprovechar señales de QZSS para autenticar la señal de GPS. Para implementar este sistema no es necesario realizar cambios de hardware en los dispositivos receptores, pero sí en la estructura de los mensajes señal propia de QZSS L1SAIF. Requiere la implementación de un *Authentication Data Center (ADC)* para conectar con el sistema de control y poder ofrecer el servicio de autenticación.

Otra alternativa es la encriptación del código de ensanchamiento. Chimera es una aproximación para implementar un sistema de autenticación en GPS basada en este tipo de encriptación. Utilizando un método de time-binding, se enlaza el mensaje de navegación y el código de ensanchamiento de las señales GPS transmitidas en abierto, como es el caso de L1C. Esto aporta una mayor seguridad ante cierto tipo de ataques de *spoofing*, mientras que conlleva un aumento en la complejidad del sistema y en la capacidad de almacenamiento y procesamiento requeridos en los receptores.

Será necesario desplegar un sistema de distribución de certificados digitales PKI. En el sistema de autenticación existen dos protocolos diferentes que conviven de forma simultánea: uno destinado a usuarios que tienen acceso a una red de comunicación *out-of-band* y otro para los que solo tienen conexión con GPS.

Se ha diseñado e implementado una prueba de concepto simplificada para comprobar el funcionamiento del sistema de cadena de claves presente en el sistema de autenticación OSNMA de Galileo. Se ha optado por utilizar el lenguaje de programación Matlab para generar una cadena de claves, utilizando las mismas funciones hash que se aplican en el sistema OSNMA. A través del software desarrollado, el usuario puede modificar el mensaje de navegación recibido para comprobar que el sistema es capaz de detectar que el mensaje recibido no es el mismo que el transmitido, obteniéndose un fallo en la autenticación. También puede comprobarse que el sistema es capaz de detectar un mensaje que no procede de una fuente fiable, aunque el contenido del mensaje sea el mismo que el original. Esto se realiza mediante la comprobación de las MAC generadas a partir de las claves y mensajes recibidos.

Por último, se ha realizado una comparativa de los dos sistemas de autenticación que se han analizado en mayor profundidad en este Trabajo de Fin de Máster: OSNMA de Galileo y Chimera de GPS. Para ello se han desglosado las características y funcionalidades más importantes de cada uno, poniendo el foco tanto en las ventajas que supone su implementación como en las limitaciones y requerimientos que conlleva su uso.

Planificación del Trabajo

En los primeros pasos del Trabajo se realizó una planificación con tareas, apartados y tiempos asignados en función de la dificultad de búsqueda y síntesis de información. Las tareas de búsqueda de información se extendieron más allá del tiempo planificado en un principio, por lo que se ha necesitado comprimir el tiempo dedicado a algunas de las tareas restantes.

La síntesis de los métodos de ataque mediante *spoofing* tomó más tiempo del esperado, así como la búsqueda de información de los métodos de defensa y aspectos generales de la defensa basada en métodos criptográficos.

El tiempo y recursos dedicados a los apartados de los métodos de autenticación basados en encriptación del mensaje de navegación coincidieron en gran medida con los planificados. Puesto que OSNMA era el aspecto a desarrollar en mayor profundidad, se le asignaron unos recursos mayores que al resto de tareas. Por otro lado, se consiguió reducir el tiempo destinado a QZSS con respecto al planificado, lo que alivió la compresión de tareas generada por la extensión de los primeros puntos.

El tiempo dedicado al diseño e implementación de la prueba de concepto fueron mayores de lo esperado y se realizó de forma simultánea a otras tareas de búsqueda de información. Se superó el tiempo que se había asignado a esta tarea y fue necesario reducir el alcance, simplificando la prueba de concepto obtenida.

Se dedicó una cantidad de recursos mayor de lo esperado al apartado de autenticación por encriptación del código de ensanchamiento. Para cumplir los plazos de entrega, fue necesario añadir más horas de trabajo dentro de los días estimados en la planificación.

El tiempo dedicado a la comparativa y a las conclusiones coincidieron en gran medida con el planificado. Sería deseable haber obtenido una comparativa más exhaustiva en apartados como requerimientos hardware o económicos de los sistemas analizados.

Líneas de Trabajo Futuro

- Como líneas de trabajo futuro, se puede desarrollar una prueba de concepto de OSNMA más detallada, donde se utilice un sistema de distribución de certificados digitales PKI para firmar las claves pertenecientes a la cadena de claves. Podría ampliarse el detalle de los mensajes de navegación utilizados, especificando los apartados H-K-Root y MAC-K, y comprobando su integridad mediante CRC. Mediante esta implementación podría incluirse el identificador del satélite para testear el funcionamiento de la cadena de claves única utilizada en *cross-authentication*.

- Se puede implementar una prueba de concepto que utilice una cadena de claves cuya longitud sea definida por el usuario en lugar de fijada por el desarrollador de antemano. Esta implementación se puede utilizar para realizar una comparativa de prestaciones obtenidas en función de la longitud de la cadena de claves utilizada.

- Otra de las posibles líneas de trabajo futuro es la realización de una comparativa de requerimientos hardware en detalle para la implementación de los sistemas OSNMA y Chimera. Estos requerimientos conllevan una inversión económica en términos de costes de implementación (*CapEx*) y costes de operación (*OpEx*) que puede ser medida y detallada para comparar ambos sistemas en términos económicos, en relación con el nivel de seguridad alcanzado al implementar el sistema de autenticación.

Glosario

GNSS	Global Navegation Satellite System
ONSMA	Open-Service Navigation Message Authentication
RAIM	Receiver Autonomous Integrity Monitoring
SCER	Security Code Estimation and Replay
RPM	Received Power Monitoring
NMA	Navigation Message Authentication
OS	Open Service
CS	Commercial Service
SOL	Safety of Life
SAR	Search and Rescue
TESLA	Timed Efficient Stream Loss-tolerant Authentication
MAC	Message Authentication Code
TTFF	Time To First Fix
TTFAF	Time To First Authenticated Fix
AER	Authentication Error Rate
BER	Bit Error Rate
TBA	Time Between Authentications
MPT	Maximum Predictable Time
USR	Unpredictable Symbol Ratio
PVT	Position Velocity Time
OTA	Over-The-Air
DSM	Digital Signature Message
IOD	Issue-Of-Data
ADKD	Authentication Data & Key Delay
SCA	Spreading Code-level Authentication
QZSS	Quasi-Zenith Satellite System
MSAS	Multi-functional Satellite Augmentation System
EGNOS	European Geostationary Navigation Overlay Service
GAGAN	GPS Aided Geo Augmented Navigation
SBAS	Satellite Based Augmentation System
PKI	Public Key Infrastructure
LDPC	Low Density Parity Check
ADC	Authentication Data Center, Analog Digital Converter
CDMA	Code Division Multiple Access
GST	Galileo System Time
CHIMERA	Chips-Message Robust Authentication
IS	Interface Specification
BOC	Binary Offset Carrier
PRN	PseudoRandom Noise

Bibliografía

[1] Skydel, «Skydel. Test Setup for Vehicle Spoofing Mitigation,» [En línea]. Available: https://www.skydelsolutions.com/en/resources/app-notes/app-note-vehicle-spoofing/.

[2] M. L. Psiaki and T. E. Humphreys, «"GNSS Spoofing and Detection",» *Proceedings of the IEEE,* vol. 104, nº 6, pp. 1258-1270, May 2012.

[3] José A. López-Salcedo, José López Vicario, Gonzalo Seco Granados, «El sistema GPS. PID_00252620.,» de *Sistemas de Radionavegación. Módulo 3.*, UOC.

[4] D. A. J. Fenn, «MIT-OpenCourseWare,» [En línea]. Available: https://ocw.mit.edu/resources/res-ll-002-adaptive-antennas-and-phased-arrays-spring-2010/.

[5] R. M. Ferré, «Analysis of GNSS replay-attack detectors exploiting unpredictable symbols,» *Thesis Advisor: Gonzalo Seco Granados, Universitat Autònoma de Barcelona, Department of Telecommunications and Systems Engineering,* Febrero 2018.

[6] Dhruvil M Modi,, «Quora. What's the difference between symmetric and public-key cryptography?,» [En línea]. Available: https://www.quora.com/Whats-the-difference-between-symmetric-and-public-key-cryptography.

[7] C. O'Driscoll, «What is navigation message authentication?,» *Inside GNSS,* pp. 26-31, Ene/Feb 2018.

[8] Raja Datta, Ningrinla Marchang, «A Secure On Demand Routing Protocol for Ad Hoc Networks (ARIADNE),» de *Handbook on Securing Cyber-Physical Critical Infrastructure*, 2012, pp. 147-190.

[9] Andrew Neish, Todd Walter, Per Enge, «Byte-Sized Security: Data Authentication for SBAS,» *GPS Laboratory, Stanford University.*

[10] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simón, I. Rodríguez, J. D. Calle, «A Navigation Message Authentication Proposal for the Galileo Open Service,» *Navigation,* vol. 63, nº 1, pp. 85-102, 01 Abril 2016.

[11] G. GNSS, «I/NAV Message,» [En línea]. Available: https://galileognss.eu/inav-message/.

[12] «Galileo Open Service Signal in Space Interface Control Document,» 2016.

[13] I. Fernandez, V. Rijmen, T. Ashur, P. Walker, G. Seco, J. Simon, C. Sarto, D. Burkey, O. Pozzobon, «Galileo Navigation Message Authentication Specification for Signal-In-Space Testing,» Noviembre 2016. [En línea]. Available: https://www.gsa.europa.eu/sites/default/files/procurement/annex_i_-_tender_specifications_1.zip.

[14] I. Fernández-Hernández, G. Seco-Granados, «Galileo NMA Signal Unpredictability and Anti-Replay Protection,» *IEEE,* 2016.

[15] Koichi CHINO, Dinesh MANANDHAR, Ryosuke SHIBASAKI, «Authentication Technology using QZSS,» *The University of Tokyo, Center for Spatial Information Science ,* 2014.

[16] J. L. Anderson, K. L. Carroll, N. P. DeVilbiss, J. T. Gillis, J. C. Hinks, B. W. O'Hanlon, J. J. Rushanan, L. Scott, R. A. Yazdi, R.A, «Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals,» ION GNSS+, 2017.

[17] AIR FORCE RESEARCH LABORATORY SPACE VEHICLES DIRECTORATE ADVANCED GPS TECHNOLOGY, «Chips Message Robust Authentication (Chimera) Enhancement for the L1C Signal: Space Segment/User Segment Interface,» de *INTERFACE SPECIFICATION IS-AGT-100*, 17-Apr-2019.

[18] Anna Poltronieri, Gianluca Caparra, Nicola Laurenti, «Analysis of the Chimera Time-Binding Scheme for Authenticating GPS L1C,» *Department of Information Engineering, University of Padova,* Dec 2018.

[19] Mathworks, «randi function,» [En línea]. Available: https://es.mathworks.com/help/matlab/ref/randi.html.

[20] Microsoft .NET, «SHA256Managed Class,» [En línea]. Available: https://docs.microsoft.com/es-es/dotnet/api/system.security.cryptography.sha256managed.

[21] Microsoft .NET, «System.Security.Cryptography Namespace,» [En línea]. Available: https://docs.microsoft.com/es-es/dotnet/api/system.security.cryptography?view=netframework-4.8.

[22] P. G. Wang, « HMAC Hash Message Authentication Code Function,» [En línea]. Available: https://es.mathworks.com/matlabcentral/fileexchange/46182-hmac-hash-message-authentication-code-function.

[23] Mathworks, «isequal function,» [En línea]. Available: https://es.mathworks.com/help/matlab/ref/isequal.html.

[24] Jan, « DataHash,» [En línea]. Available: https://es.mathworks.com/matlabcentral/fileexchange/31272-datahash.

Anexos

Index	Sector Pattern
0	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
1	1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
2	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1
3	1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
4	0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1
5	1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1
6	0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0
7	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
8	0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1
9	1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1
10	0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0
11	1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1
12	0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0
13	1 0 0 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 0
14	0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1
15	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
17	1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
18	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
19	1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
20	0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0
21	1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
22	0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1
23	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
24	0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0
25	1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0
26	0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1
27	1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
28	0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1
29	1 0 0 0 0 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1
30	0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0

Anexo 1. Tabla de selección de *sector patterns* en Chimera. [17]