Manfred Bauer, Daniel Simon Maier

Bild: ipopba/iStock/Getty Images

Eine der bei GNSS auftretenden Bedrohungen besteht darin, die Empfänger dahingehend zu manipulieren gefälschte, zu einer falschen Navigationslösung führende Navigationssignale zu verwenden: Das Spoofen.

Authentifizierung: ein Verfahren zur Bekämpfung von Spoofing für Galileo und GPS.

Gegen das Spoofen kann man sich auf unterschiedliche Weise wehren (s. z.B. Bauer 2019, Folien 32 – 34, Fernández-Hernández u.a. 2019). Fast immer sind die dabei anzuwendenden Verfahren nur mit hochwertigen Equipment (Empfänger, Antennen) möglich, kommen also für den Massenmarkt kaum in Frage. Aus diesem Grunde werden bei GPS und Galileo derzeit Verfahren entwickelt, die auch einfache Empfänger in die Lage versetzen sollen Spoofing-Angriffe zu erkennen.

Bei diesen Verfahren werden den GNSS-Signalen spezielle, zur Spoofing-Abwehr konzipierte Informationen hinzugefügt - die GNSS-Betreiber treffen also vorbeugende Maßnahmen. Die Signale werden so gestaltet, dass mit relativ geringem Aufwand auch Massenmarkt-Empfänger Spoofing-Angriffe abwehren können. Dabei werden zwei Grundverfahren angewendet: Authentifizierung der Navigationsnachricht, Authentifizierung des Spreizcode. Die Maßnahmen können unter der Überschrift »Bereitstellung von Informationen zur Authentifizierung der GNSS-Signale« zusammengefasst werden.

Durch eine Authentifizierung soll im Fall der GNSS der Nachweis erbracht werden, dass das GNSS-Signal vom Systembetreiber selbst kommt, nicht also von einem Dritten – einem Angreifer.

Die bei GPS und Galileo angewendeten Konzepte zur Authentifizierung sind unterschiedlich.

Galileo entwickelt:

1. Für den Open Service auf E1: Navigation Message Authentication (OS-NMA) – kostenfrei.
2. Für den Commercial Service auf E6: Commercial Authentication Service (CAS) – kostenpflichtig.

GPS entwickelt:

- Für das zivile GPS L1C Signal: Chips Message Robust Authentication (Chimera) – kostenfrei.

Gemeinsam ist den Konzepten OS-NMA und Chimera, dass Empfänger, deren Firmware die Authentifizierungsverfahren nicht implementiert haben, ohne Einschränkungen wie bisher genutzt werden können.
Die wesentlichen Unterschiede bei den angewandten Authentifizierungsverfahren sind:

- Galileo: Die Navigationsnachricht wird authentifiziert,
- GPS: Der gesendete Spreizcode und die Navigationsnachricht werden authentifiziert.

Ein Schutz gegen das verzögerte Weiterleiten eines GNSS-Signals (Meaconing) und das Stören eines GNSS-Signals wird bei den Verfahren nicht erreicht.

Bei beiden Verfahren werden die zur Authentifizierung benötigten Informationen regelmäßig auf nicht vorhersehbare Weise geändert. Ein Angreifer kann diese Information nicht extrapolieren. Die Authentizität des Signals bleibt damit immer in regelmäßigen Abständen überprüfbar – wenn auch mit einer gewissen Verzögerung.

Zum Verständnis der geplanten Verfahren ist es nötig, einige grundlegende Begriffe aus der Kryptografie zu kennen. Diese Begriffe sollen im folgenden Abschnitt in dem Umfang erläutert werden, wie es zum Verständnis nötig erscheint. Leser die an einer tieferen Darstellung interessiert sind, seien auf Paar, Pelzl 2016 hingewiesen.

Einige Begriffe aus der Kryptografie

Chiffre

Eine Chiffre ist ein Verfahren, mit dessen Hilfe eine Nachricht chiffriert (verschlüsselt) bzw. dechiffriert (entschlüsselt) wird. Ein Synonym für Chiffre ist »kryptografischer Algorithmus«.

Schlüssel (Key)

Der Schlüssel einer Chiffre ist die Information, mit der die Parameter einer Chiffre (eines kryptografischen Algorithmus) festgelegt werden.

Zum Beispiel ist die Zahl 3 ein möglicher Schlüssel für die Cäsar-Chiffre die von Julius Cäsar zur militärischen Kommunikation verwendet wurde. Dabei wird jeder Buchstabe des Alphabets um den Betrag des Schlüssels im Alphabet nach hinten verschoben. Mit dem Schlüssel „3“ wird aus A also D, aus B wird E und so weiter. Der kryptographische Algorithmus ist die Cäsar-Chiffre, der Schlüssel die Zahl 3. Moderne kryptographische Algorithmen haben mehr als einen Parameter und vor allem mehrere unterschiedliche Schlüssel.

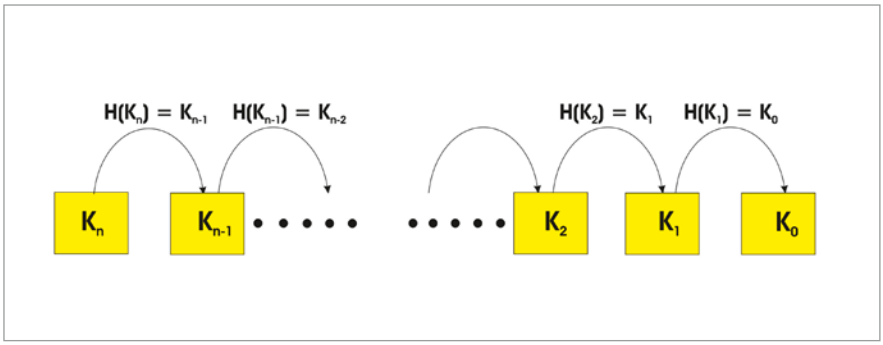

Bild 1: Bildung einer Hash-Kette

Hash Funktion

Die Hashfunktion H (deutsche Bezeichnung „Streufunktion“) ist eine mathematische Formel, die einen Klartext (Message M) beliebiger Länge in einen Block verschlüsselter Daten fest vorgegebener Länge umwandelt. Sie kann zur Erzeugung der unten beschriebenen digitalen Signatur verwendet werden.
Eine Hash Funktion sollte folgende Eigenschaften haben:

- Keine-Injektivität (Unumkehrbarkeit)
  Es soll nicht möglich sein, ihre Umkehrfunktion zu berechnen. Es soll also nicht möglich sein, aus dem Ergebnis der Hash-Funktion auf die Daten zu schließen, die auf die Hashfunktion angewendet wurde.
- Kollisionsfreiheit
  Es soll nicht passieren können, dass die Anwendung der Hash-Funktion auf zwei auch nur geringfügig unterschiedliche Klartexte zum selben Ergebnis kommt, in der Sprache der Kryptografie: dass es zu einer Kollision kommt.

Es gibt unterschiedliche Arten von Hash-Funktionen

- Unverschlüsselte Hash-Funktion (unkeyed Hash-Function)
- Verschlüsselte Hash-Funktion (keyed Hash-Function)

Im folgendem soll lediglich die verschlüsselte Hash-Funktion H(K) betrachtet und deren Verwendung beschrieben werden.

Eine verschlüsselte Hash Funktion HK benötigt zwei Eingabeparameter: den Klartext M und einen Schlüssel K. In Abhängigkeit von diesen Parametern – beide Parameter sind Datensätze unterschiedlicher Länge – wird ein neuer Datensatz D erzeugt, der in der Regel kürzer ist als der Klartext. Der neue Datensatz D wird auch als Fingerabdruck bezeichnet.

Hash Kette (Hash Chain)

Eine spezielle Anwendung der Hash Funktion führt zur Hash Kette. Das Verfahren wird bei Galileo benutzt, um Schlüssel zu erzeugen, die offen übersendet werden.
Das Verfahren läuft wie folgt ab:
Es wird ein Ursprungsschlüssel K_n (Seed Key) definiert.

Aus dem Ursprungsschlüssel K_n werden unter Anwendung einer Hash-Funktion $H(K_n)$ - z.B. Secure Hash Algorithm-256 (SHA-256), ein vom US-amerikanischen National Institute of Standards and Technology entwickelter und publizierter Algorithmus – Folgeschlüssel abgeleitet (s. Bild 1). Die n-malige Anwendung führt zu einem Schlüssel mit der Bezeichnung „Root-Key K_0“
In formelmäßiger Darstellung $K_0 = H^n (K_N)$ und als Beispiel $K_0 = H^3 = H \{ H [H (K_n)] \}$

Öffentlicher Schlüssel (public Key) vs. privater Schlüssel (private Key)

Wenn zwei Teilnehmer Informationen vertraulich austauschen wollen, können sie sich auf eine Chiffre und einen dazu passenden Schlüssel – einen privaten Schlüssel (private Key) – einigen. Ein Problem bei dieser symmetrischen Kryptografie besteht darin den privaten Schlüssel sicher auszutauschen und vor dritten geheim zu halten. Beide Seiten müssen den Schlüssel sicher gegenüber dritten verwahren.

Wenn ein sicherer Informationsaustausch zwischen einem Sender – in diesem Fall einem GNSS-Satelliten – und einer unbegrenzten Anzahl von Empfängern, die als

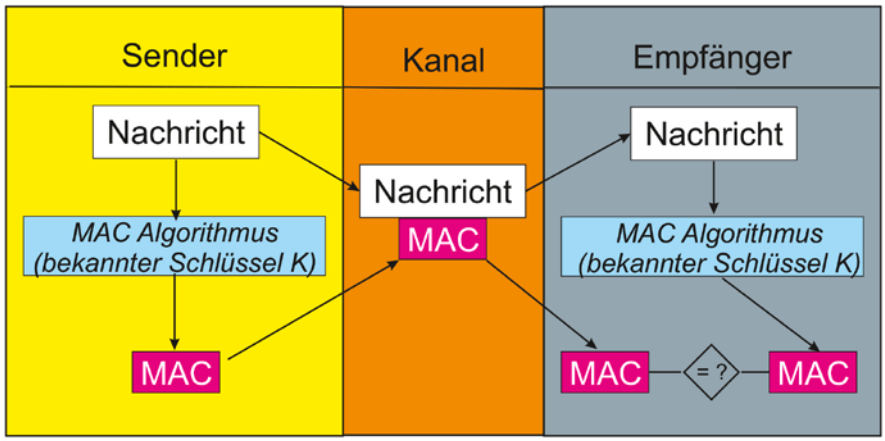

Bild 2: Funktionsweise eines Message Authentication Code (MAC) (nach Wikipedia)

unsicher einzustufen sind, organisiert werden soll – hier den Nutzern der GNSS -, steht ein anderes Verfahren zur Verfügung, das Verfahren der asymmetrischen Kryptografie. Bei der asymmetrischen Kryptografie der GNSS verwendet der Satellit einen geheimen privaten Schlüssel (private Key), die Empfänger nutzen einen öffentlichen Schlüssel (Public-Key). Der öffentliche Schlüssel ist prinzipiell nicht geheim. Geheim ist lediglich der Private-Key des Satelliten. Das von Diffie und Hellman (1976) entwickelte Verfahren ist so konzipiert, dass eine mit einem Private-Key verschlüsselte Nachricht nur dann mit dem Public-Key gelesen werden kann, wenn Private- und Public-Key aufeinander abgestimmt sind. Eine gesendete Nachricht kann nur dann mit dem Public-Key gelesen werden, wenn sie mit dem zugehörigem Private-Key verschlüsselt wurde.

Paar, Pelzl (2016) wählen für das Verfahren folgende Analogie: Es gibt einen Safe, in den nur mit einem privaten Schlüssel Nachrichten hineingelegt werden können und nur mit einem öffentlichen Schlüssel Nachrichten entnommen werden können.

Message Authentication Code (MAC)

Mit Hilfe eines MAC soll der Empfänger einer Nachricht M die Möglichkeit erhalten zu prüfen, dass er die Nachricht so wie gesendet empfangen hat.

Dazu wird vom Sender eine kryptografisch bestimmte Bit-Folge (Prüfsumme) – der MAC - unter Verwendung der Nachricht M, eines bekannten Schlüssels K und eines bekannten Algorithmus berechnet. Es gilt also MAC=f (K,M). Der MAC wird an die Nachricht angehängt. Der Empfänger erhält die Nachricht M und den MAC. Er berechnet seinerseits aus der Nachricht M, des ihm bekannten Schlüssels K und des ihm ebenfalls bekannten Algorithmus den MAC und prüft ob gerechneter und empfangener MAC identisch sind (s. Bild 2).

MAC ist ein Beispiel für symmetrische Kryptografie.

Digitale Signatur

Mit Hilfe einer Signatur soll in erster Linie erreicht werden, dass der Empfänger einer Nachricht sicher ist, dass die Nachricht vom behaupteten Sender stammt.

Die digitale Signatur Sig besteht ähnlich wie ein MAC aus einem Datensatz, der aus dem zu übertragenem Klartext errechnet wird. Für die Berechnung von Sig benötigt der Sender den privaten (geheimen) Schlüssel K_{pr}. Die Signatur Sig wird zusammen mit dem Klartext – der Message M – übersandt (s. Bild 3).

Der Nachweis für die Übermittlung vom behaupteten Sender wird dadurch erbracht, dass der Empfänger auf die empfangene Signatur Sig und die Message einen Verifikationsalgorithmus anwendet. Dazu wird ein öffentlicher Schlüssel angewendet. Der Verifikationsalgorithmus liefert nur die binäre Aussage „wahr“ oder „falsch“. Wenn M tatsächlich mit dem privaten Schlüssel signiert wurde, der zu dem öffentlichen Verifikationsschlüssel gehört, ist die Ausgabe des Verifikationsalgorithmus „richtig“, sonst ist die Ausgabe „falsch“.

Digitale Signatur ist ein Beispiel für asymmetrische Kryptografie.

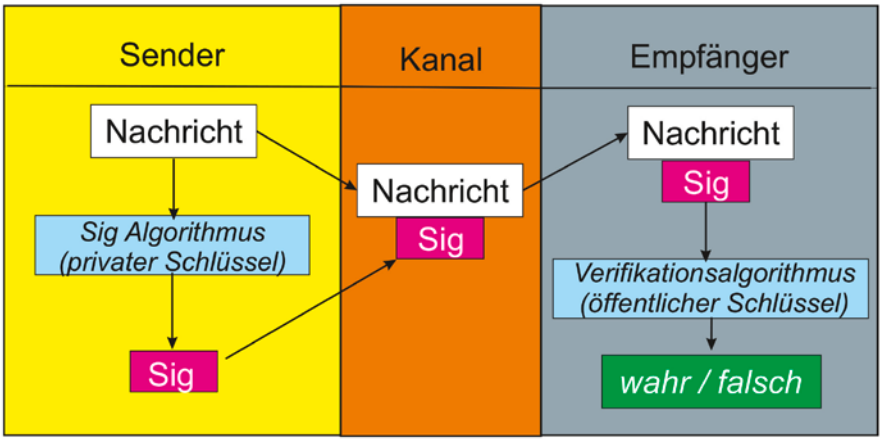

Bild 3: Funktionsweise einer digitalen Signatur

Timed Efficient Stream Loss-tolerant Authentication (TESLA)

TESLA ist ein Algorithmus zur Authentifizierung von Nachrichten. Die Entwicklung von TESLA geht zurück auf Arbeiten, die beginnend zur Jahrtausendwende an der Kalifornischen Universität Berkley und bei IBM durchgeführt wurden (Perrig u.a. 2000, Perrig u.a. 2002).

Auch bei TESLA werden Nachrichten zusammen mit MAC's gesendet. Der wichtigste Unterschied zwischen reinen MAC's und TESLA ist dass die Schlüssel, die zur Berechnung der MAC's verwendet werden,

a. sich regelmäßig ändern,
b. über einen Mechanismus regelmäßig authentifiziert werden müssen.

Der Empfänger kann also mit empfangenem MAC und immer wieder neu empfangenem und neu authentifiziertem Schlüssel (Key) die empfangene Nachricht auf Authentizität überprüfen (s. Bild 4).

Die Überprüfung der verwendeten Schlüssel geschieht unter Verwendung einer Hash-Kette zur Bereitstellung einer Schlüsselkette (s. Bild 1). Die Schlüsselkette generiert der Sender zu Beginn des TESLA-

Bild 4: Grundprinzip TESLA

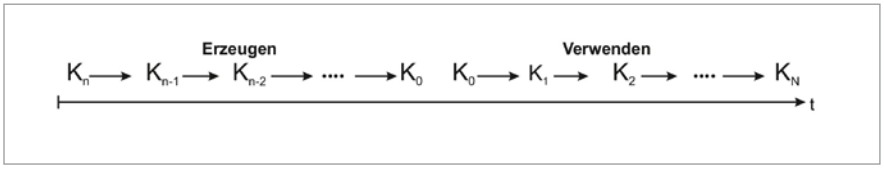

Bild 5: Erzeugung und Verwendung der Schlüssel bei TESLA

Prozesses und speichert sie ab.
Die Schlüsselkette K_n, K_{n-1}, K_{n-2}, …….K_1, K_0 wird in umgekehrter Reihenfolge K_0, K_1, ….. K_{n-2}, K_{n-1}, K_n verwendet und somit auch in dieser Reihenfolge ausgesendet (s. Bild 5). K_0 kann zu seiner Sicherheit signiert gesendet werden.

Im Empfänger wird nach dem Empfang eines Schlüssels K_L, durch L maliges Anwenden der Hashfunktion H der Schlüssel K_0 abgeleitet. Dabei ergeben sich alle Schlüssel zwischen K_{L-1} und K_0. Wenn der so gerechnete Schlüssel K_0 mit dem zu Beginn des Prozesses gesendetem Schlüssel und im Empfänger gespeicherten Schlüssel K_0 übereinstimmt, kann davon ausgegangen werden, dass der aktuelle Schlüssel K_L authentisch ist. Dann kann mit dem aktuellen Schlüssel K_L und der empfangenen Nachricht ein MAC gerechnet werden und mit dem empfangenem MAC verglichen werden.

Authentifizierung bei Galileo: "Galileo Open Service Navigation Message Authentication (OS-NMA)"

Einführung

2010 etablierte die EU die Mission Evolution Advisory Group, eine Gruppe von 25 Experten zur Weiterentwicklung von EGNOS und Galileo (European Commission 2010). 2013 empfahl diese Gruppe, für den Galileo Dienst Open-Service (OS) eine Authentifizierung zu ermöglichen; u.a. mit dem Ziel, die Authentifizierungsmöglichkeit zu einem wichtigen Unterscheidungsmerkmal in Bezug auf andere GNSS zu machen.

Die formale Entscheidung, Authentifizierungsverfahren einzuführen, traf die EC im Februar 2017 (European Commission 2017; Fernandez-Hernandez u.a. 2018). Spezifikationen für OS-NMA wurden von der Europäischen Kommission im November 2016 als Entwurf veröffentlicht (Version V.1.0).

Ziel der EC-Entscheidung war es, im Jahr 2018 ein von den Satelliten ausgestrahltes entsprechendes Signal zu testen. Unter anderem wegen der politischen Entwicklungen im Zusammenhang mit dem BREXIT hat sich die Einführung von OS-NMA verzögert. Mitte 2020 ist das OS-NMA-Signal noch im Experimentierstadium. Dennoch kann davon ausgegangen werden, dass das Verfahren über kurz oder lang eingeführt wird. Es soll daher in den folgenden Abschnitten beschrieben werden. Die theoretischen Grundlagen des Konzepts finden interessierte Leser bei Fernández-Hernández (2015), in dessen Arbeit die Weiterführung des von Perrig u.a. (2002) entwickelten TESLA-Verfahrens eine zentrale Rolle spielt.

OS-NMA Daten

Die für OS-NMA erforderlichen Daten sollen im Kanal B des offenen Signals E1 in der I/NAV Nachricht bereitgestellt werden. Die dort verfügbare Navigationsnachricht wird wie bisher unverschlüsselt ausgestrahlt. Zur Authentifizierung der Navigationsnachricht

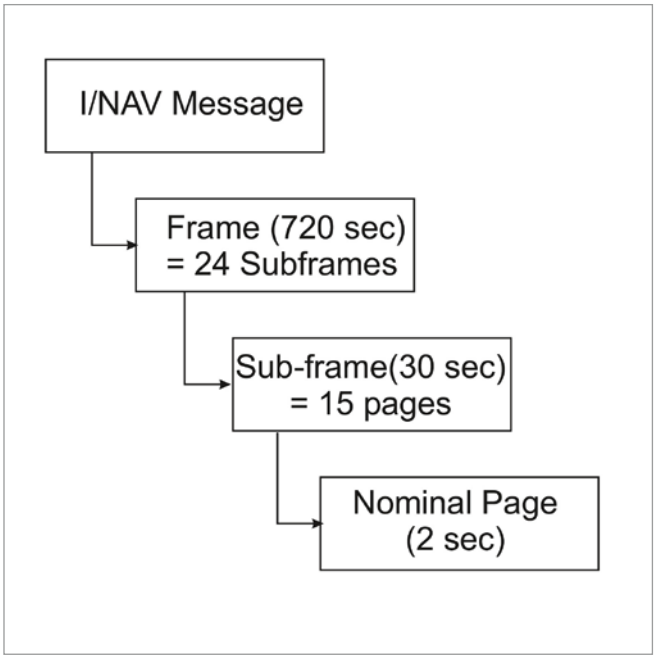

BIld 6: Struktur der nominellen Galileo I/NAV Message

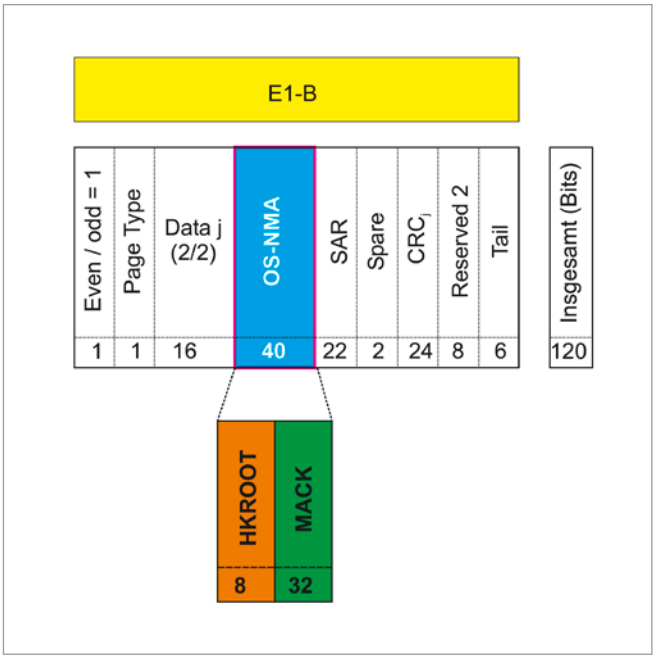

Bild 7: Teil 1 einer I/NAV Page mit OS-NMA Bereich

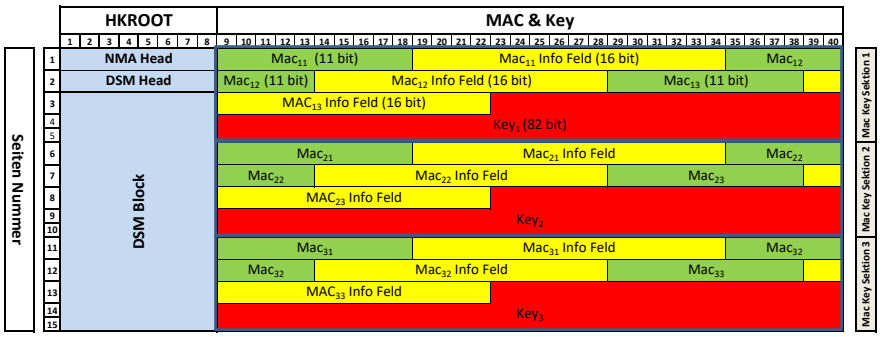

Bild 8 Anordnung (Struktur) der OS-NMA Informationen in einem Subframe

werden folgende Daten zusätzlich zur Navigationsnachricht ausgestrahlt:

- MAC's – berechnet aus den Navigationsnachrichten,
- Schlüssel zur Chiffrierung bzw. Dechiffrierung der MAC's ,
- Eine Signatur für den Root Key (s. Kapitel 3) der verwendeten Hash-Kette.

In Kap. 3 wurde dargestellt, dass MAC zu den Verfahren der symmetrischen Kryptografie gehört, die Signatur zu den Verfahren der asymmetrischen Kryptografie. OS-NMA verwendet demnach eine Kombination aus beiden Verfahren, die Hybrid-Kryptografie.

Struktur der OS-NMA Daten

Die I/NAV Nachricht des Galileo E1 Signals besteht aus Frames, Subframes und Pages (s. Bild 6).

Alle 30 Sekunden wird ein aus 15 Pages bestehender Subframe gesendet, alle 2 Sekunden eine I/NAV Page. Zu einer Page gehören zwei Teile von je 120 Bits. Jeder Subframe enthält wie bisher alle zur Durchführung einer Navigationslösung notwendigen Informationen (die Navigationsnachricht M).

In jedem ersten Teil der 15 Seiten (pages) gibt es zusätzlich die bisher nicht genutzten Bereiche „Reserved 1" (40 Bits) und „Reserved 2" (8 Bits). Die 40-Bits des „Reserved 1"- Bereichs werden für die Bereitstellung der für die Authentifizierung nötigen Daten in Anspruch genommen. (s. Bild 7). Pro Subframe stehen damit 15 ·40=600 Bits für die Authentifizierung erforderlichen Daten zur Verfügung.

Der OS-NMA Bereich ist in zwei Sektionen gegliedert. (s. Bild 7 und 8):

1. HKROOT Section (Headers and KROOT, 8 Bits pro Seite)
   - Enthält im wesentlichem den signierten Ursprungsschlüssel (Root-Key K_0) und seine Signatur (DSM (Digital Signature Message)).
2. MAC & Key Section (MACK, 32 Bits pro Seite)
   - Enthält im wesentlichem MACs und Schlüssel (Keys) zur Authentifizierung der Navigationsdaten.

Pro Subframe gehören damit zur HKROOT Section 15 x 8 Bits =120 Bits , zu MACK & Key Section 15 x 32 Bits = 480 Bits. Zu OS-NMA gehören zusammen also 600 Bits pro Subframe.

Bild 8 zeigt eine mögliche OS-NMA-Implementierung in detaillierter Darstellung (Binda 2018). Man kann daraus u.a. entnehmen, dass zur Authentifizierung von drei MACs ein gemeinsamer Key genutzt wird. Der entsprechende Key steht erst nach Aussendung von fünf Seiten zur Verfügung, also nach Ausendung der drei MAC's. Einzelheiten für mögliche Implementierungen sind in der OS-NMA Spezifikation geregelt. Für das weitere Verständnis ist es wichtig zu wissen, dass zur Aussendung der vollständigen HKROOT K-Root Section mindestens sechs Subframes benötigt werden (Fernández Hernández 2015).

Auswertung der OS-NMA Daten mit TESLA-Algorithmus

Voraussetzung für die Authentifizierung ist, dass der Empfänger den bei OS-NMA signiert gesendeten Root-Key (K_0) zur Verwendung nach dem weiter oben beschriebenen TESLA-Algorithmus entschlüsselt hat. K_0 und seine Signatur – die Digital Signature Message (DSM) - werden im HKROOT gesendet. Da die DSM über sechs Subframes verteilt ist, kann es bis zu 180 Sekunden dauern, bis der Root-Key zur Verfügung steht. Dies gilt allerdings nur beim erstmaligen Gebrauch des Empfängers oder wenn der Empfänger über mehrere Monate nicht in Gebrauch war. Im regulären Betrieb erlaubt OS-NMA eine Authentifizierung der Nachricht alle 10 Sekunden.

Die Signierung von K_0 erfolgt unter Verwendung des Elliptic Curve Digital Signature Algorithm (ECDSA) (Institute of National Standards and Technology, 2013). Der Satellit nutzt dazu einen geheimen Private-Key, der Empfänger zur Authentifizierung von K_0 den dazu passenden Public-Key. Er kann dem Empfänger auf unterschiedlichen Wegen zur Verfügung gestellt werden kann, z.B. durch dessen Implementierung in der Empfänger-Firmware, aber auch mit Hilfe einer sicheren Internetverbindung.

Die Berechnung der benötigten Hash-Kette wird im Bodensegment durchgeführt und mit der Navigationsnachricht auf den Satelliten hoch geladen.

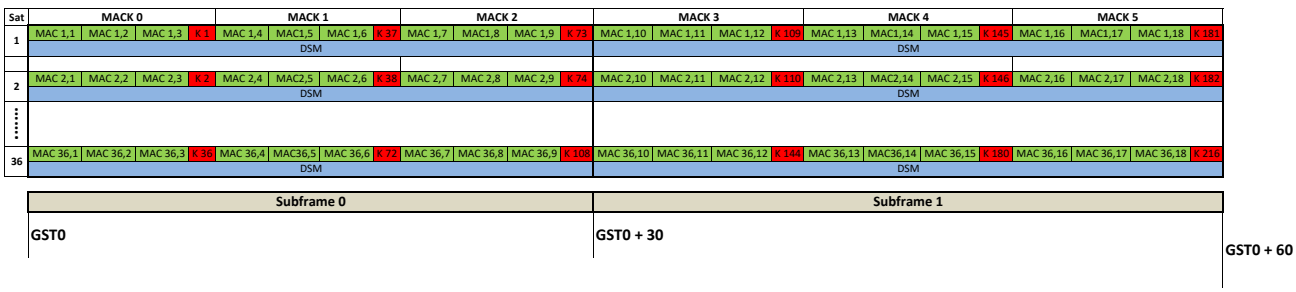

Bild 9: Verwendung der Schlüssel in den ersten beiden Subframes

1	Der Empfänger empfängt die Navigationsnachricht sowie die daraus mit einem bekannten kryptographischen Algorithmus abgeleiteten drei MACs.
2	Der Empfänger empfängt den Schlüssel K_L mit deren Hilfe die MACs generiert wurden.
3	Der Empfänger authentifiziert K_L durch die erforderliche Anzahl von Anwendungen der Hash Funktion. Er rechnet also $K'_0 = H^{L-1}(K_L)$. Wenn gilt $K'_0 = K_0$,ist der empfangenen Schlüssel K_L authentisch.
4	Der Empfänger berechnet mit Hilfe des Schlüssels K_L unter Verwendung der Navigationsnachricht *MAC'*s und vergleicht sie mit den empfangenen MACs. Wenn sich ergibt, dass im Empfänger gerechnete und empfangene MACs identisch sind, sind die empfangenen Navigationsnachrichten authentisch.

Tabelle 1: Ablauf einer OS-NMA Authentifizierung

Die Länge der für alle Satelliten gleichen Schlüsselkette ist definiert durch die Satellitenanzahl, die Schlüsselrate (die Zeit zwischen zwei Schlüsseln) und die gewünschte Gültigkeitsdauer der Schlüsselkette. Nach Ziffer 6.6 der OS-NMA Spezifikation soll eine Kette nach Erzeugung von 2^{25} bis 2^{26} Schlüsseln durch eine neue Kette ersetzt werden. Bei 2^{25} = 33.554.432 Schlüsseln ergeben sich bei 36 Satelliten 932.068 Schlüssel für einen Satelliten. Unter Berücksichtigung der Übertragungszeit von 10 Sekunden für einen Schlüssel (s. Bild 7) erhält man eine Periode von rd. 4 Monaten für die Schlüsselkette des Satelliten.

Bild 9 zeigt wie bei Verwendung von 36 Satelliten die Schlüssel auf die Satelliten in Abhängigkeit von der Galileo-System-Time (GST) verteilt werden.

Entsprechend dem TESLA-Algorithmus wird im Empfänger nach dem Empfang eines Schlüssels K_L, durch L maliges Anwenden der Hashfunktion H der Schlüssel K_0 abgeleitet. Da nicht ausgeschlossen werden kann, dass die Anzahl der erforderlichen Hash-Operationen die Rechenkapazität eines Empfängers überfordert, werden in der HK-Root-Section mit fortschreitender Zeit der Ursprungsschlüssel K_0 durch sogenannte „ floating KROOTs“ ersetzt. Sie sollen die Authentifizierung für die Empfänger erleichtern, die erst in der Mitte einer TESLA-Kette eingeschaltet werden.

Die Schlüssel K_1 bis K_L können erst dann verwendet werden wenn, wenn der Root Key K_0 mit seiner Signatur – seiner Digital Signature Message (DSM) – zur Verfügung steht. Der weitere Ablauf soll der Einfachheit halber für den Fall geschildert werden, dass der Satellit den signierten Root-Key K_0 bereist empfangen sowie verifiziert hat Dann gestaltet sich das weitere Prozedere wie in Tabelle 1 skizziert (s. dazu auch Bild 4).

Authentifizierung bei GPS: “Chips Message Robust Authentication (Chimera)” für das zivile GPS L1C Signal

Einführung

Auch die USA planen zukünftig die Authentifizierung eines zivilen GPS-Signals zu ermöglichen – des L1C Signals. Die dazu erforderlichen Entwicklungsarbeiten werden derzeit vom US-amerikanischen Air Force Research Laboratory (AFRL) durchgeführt. Ein entsprechendes Signal soll der experimentelle NTS-3 Satellite (Navigation Technology Satellite 3) senden. Der Start des Satelliten ist für das Jahr 2023 geplant (Divis, 2019).

Die grundlegende Idee für das anzuwendende Authentifizierungsverfahren geht auf ein im Jahr 2003 veröffentlichtes Papier von Logan Scott zurück (Scott 2003). Scott schlägt vor den Nachweis erbringen zu lassen dass nicht nur die gesendete Navigationsnachricht geprüft wird sondern zusätzlich auch der gesendete Spreizcode. Es werden also die Authentizität der Nachricht und des Spreizcodes geprüft. Unter dem Eindruck von immer mehr erfolgreichen Spoofing-Angriffen werden die Ideen von Scott derzeit wieder aufgegriffen (Scott, 2019).

Im April 2019 hat AFRL eine Interface Spezifikation für das geplante Authentifizierungsverfahren Chimera herausgegeben (IS-AGT-100). Sie kann auf der Internetseite von Logan Scott (http://www.gpsexpert.net/chimera-specification) eingesehen und heruntergeladen werden.

Zum Verständnis des Chimera-Verfahrens ist es erforderlich, sich einige Aspekte des L1C-Signals und des dazu gehörenden CNAV-2 Datenformats vor Augen zu führen. (Einzelheiten s. „IS-GPS-800E“ vom 22 Mai 2018).

Das GPS-L1C-Signal

Das L1C-Signal besteht aus einem Daten- und einem Pilot-Signal. Die Signale verwenden unterschiedliche Codes und geringfügig unterschiedliche Modulationsverfahren. Zur Authentifizierung nutzt Chimera beide Signalkomponenten. Im Datensignal wird Daten-Authentifizierung ermöglicht, im Pilotsignal Spreizcode Authentifizierung

- $L1C_D$-Signal (Datensignal).
  Das Datensignal ist in Messages von 18 Sekunden Länge unterteilt (s. Bild 10). Jede Message besteht aus drei Subframes unterschiedlicher Länge (Subframe 1: 9 Bits; Subframe 2: 600 Bits; Subframe 3: 274 Bits). In Zusammenhang mit Chimera ist Subframe 3 von besonderer Bedeutung.

Zu Subframe 3 jeder Message gehört eine Seite (page) mit abwechselnden Inhalten. Festgelegt ist in IS-AGT-100, dass Seiten mit den bisher nicht genutzten Nummern 8 und 9 von Chimera genutzt werden. Dort wird die zur Authentifikation erforderlichen digitale Signatur abgespeichert. Für die Aussendung der Seiten mit den Nummern 1 bis 7 gibt es keine festgelegte Reihenfolge

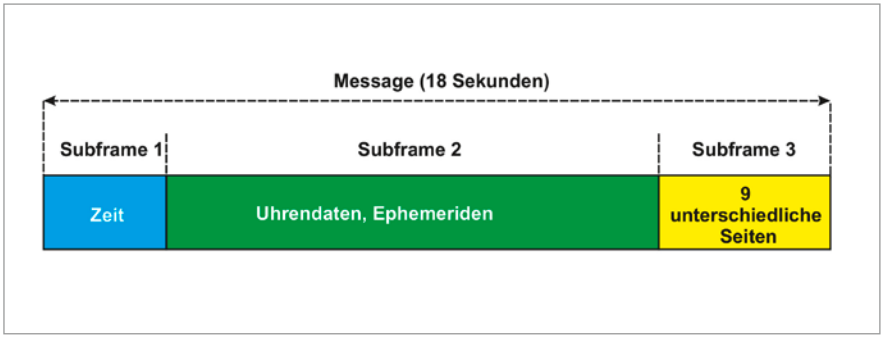

Bild 10: $L1C_D$-Signal

Bild 11: Chimera Epoche

(s. Ziffer 3.5.5.1 IS-GPS-800). Für Messages mit Seiten Nr. 8 und Nr. 9 des Subframe 3 sind im IS-AGT-100 Ziffer 3.4.5 bestimmte Positionen innerhalb einer Chimera-Epoche festgelegt (s. Bild 11 bzw. Abschnitt 5.3).

- $L1C_P$ Signal (Pilot-Signal)
  Das gegenüber dem Datensignal mit 3fach größerer Energie ausgestrahlte Pilot-Signal ist datenfrei. Es ist lediglich mit einem Spreizcode moduliert. Bei Chimera wird der $L1C_P$ Spreizcode zu seiner Authentifizierung geringfügig modifiziert.

5.3 Die Chimera Epoche

Zur weiteren Beschreibung muss der Begriff der Chimera-Epoche eingeführt werden. Sie wird aus 10 Messages des L1C Datensignals gebildet, dauert also 180 Sekunden (s. Bild 11).

Teile der innerhalb einer Epoche ausgestrahlten Daten werden vor ihrer Aussendung verkettet und die gebildete Kette mit einem privaten Schlüssel signiert (s. Ziffer 20.2.2 IS-AGT-100). Die entsprechende Signatur wird in die Seiten 8 und 9 der Subframes 3 eingefügt. Zu der innerhalb der Epoche zuletzt ausgestrahlten Message gehört für deren Subframe 3 immer die Seite 9. Die Message mit Seite 8 des Subframe 3 kann an beliebiger Stelle der verbleibenden 9 Intervalle von je 18 Sekunden ausgestrahlt werden. Erst mit der zuletzt ausgesandete Message einer Chimera Epoche mit ihrer Seite 9 können mit Hilfe der nunmehr bekannten Signatur:

1. die Dateninhalte authentifiziert werden,
2. Informationen darüber gewonnen werden, an welchen Segmenten des $L1C_P$ Spreizcodes durch Einfügen von Spread Spectrum Security Codes (SSSC) Veränderungen vorgenommen wurden.

Die entsprechenden $L1C_P$ Segmente bestehen aus 33 Chips, von denen vier Chips unverändert bleiben.

Die Anzahl von 33 Chips für jedes Segment ergibt sich daraus dass der in 10 ms ausgestrahlte L1C Pilot-Code aus 10.230 Chips besteht, der sich in 310 Segmente von je 33 Chips aufteilen lässt. Von diesen je 33 Chips sind 4 Chips an festgelegten Stellen BOC (6,1) moduliert, die anderen 29 Chips des Segments BOC(1,1) moduliert (s. Bild 12). Nur BOC(1,1) modulierte Chips können durch Spread Spectrum Security-Codes (SSSC) ersetzt werden, die BOC(6,1) modulierten Chips bleiben unverändert.

Die SSSC verschlechtern zwar die Korrelationseigenschaft des Pilot-Signals, dies aber nur in vernachlässigbarem Umfang. Sie sind vergleichbar Wasserzeichen eines Geldscheins. Empfänger, die Chimera nicht implementiert haben, können mit diesen Wasserzeichen das Pilotsignal zur Durchführung des Korrelationsprozesses ohne nennenswerte Einschränkungen weiter verwenden. Die SSSC sind aber nachweisbar, wenn man weiß, wo sie eingefügt wurden.

Auswertung der Chimera Daten – Authentifizierung

Am Ende einer Chimera-Epoche (nach 180 Sekunden) stehen folgende Informationen dem Empfänger zur Verfügung:

1. Die mit einem private Key erstellte und dem zugehörigen public Key verifizierte Signatur zur Datenauthentifizierung (entnommen den Seiten 8, 9 der Subframes 3 des Datensignals),
2. Die Position und Anzahl der 33 Chips langen Segmente mit SSSC im Spreizcode des Pilotkanals (abgeleitet mit Hash-Algorithmus aus der Signatur, übermittelt im Datenkanal),
3. Die Bit-Folge der jeweiligen SSSC (abgeleitet mit Hash-Algorithmus aus der Signatur).

Die entsprechenden Informationen sind für jeden Satelliten anders.

Der Empfänger kann damit unter Verwendung des public Key prüfen:

1. Sind die Daten authentisch? – dazu wird die Signatur verwendet,
2. Sind die Veränderungen im Spreizcode des Pilotkanals so vorgenommen wie aus der Signatur und den zugehörigen Hash-Algorithmen ableitbar? Wurden die SSSC entsprechend eingefügt?

Voraussetzung für die Überprüfung der Richtigkeit der SSSC-Einfügung in das Pilotsignal ist, dass der Empfänger zuvor am Ausgang des A/D-Wandlers Abschnitte der ankommenden Rohdaten abgespeichert hat. Wenn er Abschnitte von 0,1 Sekunden Länge abspeichert, hat er den 10 ms dau-

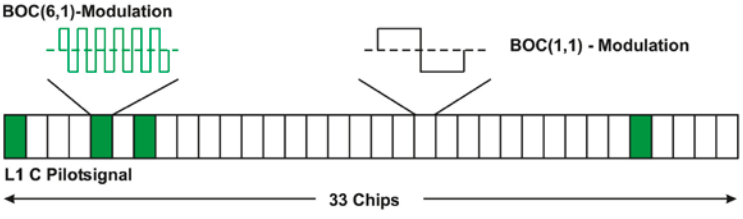

Bild 12: Ein Segment des $L1C_P$ Pilotsignals

ernden Code 10 mal hintereinander abgespeichert. Je nach A/D-Wandler fallen unterschiedliche Datenmengen an. Bei einem 2 Bit Analog/Digitalwandler mit einer Sample Rate von10 MHz wird bei einem Aufzeichnungsintervall von 0,1 Sekunden eine Speicherkapazität 0,25 Mbytes benötigt (Anderson u.a. 2017). Im Empfänger können die empfangenen und entsprechend gespeicherten Codeteile mit dem Sollspreizcode korreliert werden. So lässt sich erkennen, ob der ausgestrahlte Code authentisch ist und alle drei Minuten eine Authentifizierung durchführen. Die relativ kleinen Korrelationsabschnitte – die entsprechenden Segmente – führen dazu, dass die Detektion der Korrelationsspitze schwierig und damit fehleranfällig ist. Die Spreizkode Authentifizierung ist also nicht so robust wie die Nachrichten Authentifizierung. Des Weiteren müssen die GNSS Empfänger mit entsprechender Speicherkapazität ausgestattet werden, was eine hardwareseitige Änderung der Empfänger erfordert.

Leser die an einer detailreicheren Darstellung des Chimera-Verfahrens interessiert sind, seien auf das Papier von Anderson u.a. 2017 hingewiesen.

Zusammenfassung

Mit OS-NMA und Chimera werden für Galileo und GPS über kurz oder lang Verfahren zur Verfügung stehen, die das Spoofen dieser GNSS zumindest erschweren. Über ähnliche Verfahren wird auch für BDS nachgedacht (Wu u.a. 2020). Es gibt andere Verfahren gegen Spoofing (Fernández-Hernández, 2019, Bauer 2019). Für die hier beschriebenen Verfahren sind die technischen Voraussetzungen zu ihrer Realisierung nicht besonders anspruchsvoll. Sie stehen bis auf geringfügige Änderungen in der Firmware der Satellitenempfänger den GNSS-Nutzern kostenfrei zur Verfügung. Michael Ritter, Präsident von Hexagon's Autonomy & Positioning Division wird von Gutierez (2020) mit dem Satz zitiert: „When I talk to people about Galileo, Authentication is the real seller".

Literatur:

Air Force Research Laboratory. Space Vehicles Directorate. Advanced GPS Technology (2019): Interface specification IS-AGT-100

Anderson, J.M., Carroll, K.L., Nathan P., DeVilbiss, N.P., Gillis, J.T., Joanna C. Hinks, J.C., O'Hanlon, B.W., Rushanan. J.J., Scott, L. u. Yazdi, L.R. (2017): Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals. 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2017), Portland, Oregon, September 25-29, 2017

Bauer, M. (2019): Jamming, Spoofing, Meaconing. Ein Problem für Autonomeous Surface Vehicles (ASV) in der Hydrographie ? HPA Workshop 21. August 2019 "Why to use an ASV for Hydrography". https://daten.turla.de/index.php/s/Apy3JdxdM5k7XIP

Binda, S. (2018): Galileo Open Service Navigation Message,Workshop GNSS Interferentie en Authenticatie, Haarlem, NL. http://www.navnin.nl/new/wp-content/uploads/2018/02/WSIA-4-ESA-NIN-Workshop-Galileo-NMA.pdf

Diffie, W. and Hellman, M. E. (1976) New Directions in Cryptography, IEEE Transactions on Information Theory (22:6), pp. 644-654. https://ee.stanford.edu/~hellman/publications/24.pdf

Divis, A. (2019): New Chimera Signal Enhancement Could Spoof-Proof GPS Receivers. Inside GNSS. Global Navigation Satellite Systems Engineering, Policy, and Design https://insidegnss.com/new-chimera-signal-enhancement-could-spoof-proof-gps-receivers/

European Commission (2010): Commission Decision of 6 October 2010 setting up the group of experts on the mission evolution of the European navigation satellite systems, the 'Mission Evolution Advisory Group'

European Commission (2016): Galileo Navigation Message Authentication Specification for Signal-in-Space Testing – v.1.0 (https://www.gsa.europa.eu/sites/default/files/calls_for_proposals/annex_1-rd4.pdf)

European Commission (2017): COMMISSION IMPLEMENTING DECISION (EU) 2017/224 of 8 February 2017 (CS Implemeting Act), 2017.

Fernández Hernández, I. (2015): Snapshot and Authentication Techniques for Satellite Navigation. Dissertation Aalborg University Dänemark

Fernandez-Hernández, I., Vecchione, G. u. . Díaz-Pulido, F. (2018): Galileo Authentication: A Programme and Policy Perspective. 69th International Astronautical Congress, Bremen, Germany.

Fernández-Hernández, I., Walter, T., Alexander, K., Clark, B., Châtre, E., Hegarty, Ch., Appel, M. u. Meurer, M. (2019): Increasing International Civil Aviation Resilience: A Proposal for Nomenclature, Categorization and Treatment of New Interference Threats. International Technical Meeting of the Institute of Navigation. Reston, Virginia, USA. https://elib.dlr.de/127030/

Gutierrez, P. (2020): Galileo to Transmit Open Service Authentication. Inside GNSS January / February 2020

Paar, C., Pelzl, J. (2016); Kryptografie verständlich. Ein Lehrbuch für Studierende und Anwender. Springer Vieweg.

Perrig, A., Canetti, R., Tygar, J.D. u. Song,D. (2000). Efficient authentication and signature of multicast streams over lossy channels. In Proceedings of the IEEE Symposium on Research in Security and Privacy, pages 56–73, May 2000. https://netsec.ethz.ch/publications/papers/stream.pdf

Perrig, A., Canetti , R., Tygar, J.D. u. Song, D. (2002): The TESLA Broadcast Authentication Protocol. In CryptoBytes, 5:2, Summer/Fall 2002, pp. 2-13 https://people.eecs.berkeley.edu/~tygar/papers/TESLA_broadcast_authentication_protocol.pdf

Scott, L. (2003): Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems. ION GPS/GNSS 2003, 9-12 September 2003, Portland, OR

Scott, L. (2019): The Role of Civil Signal Authentication in Trustable System. Presentation to PNT Advisory Board, June 2016

Wu, Z. Zhang, Y. und Liu, R. (2020): BD-II NMA&SSI: An Scheme of Anti-Spoofing and Open BeiDou II D2 Navigation Message Authentication. IEEE Access, Volume 8, 2020.

Prof. (i.R.) Dipl.-Ing. Manfred Bauer
Pfarrstrasse 13
22149 Hamburg
E-Mail: m.bauer-hh@t-online.de
www.Vermessung-und-Ortung-mit-Satelliten.de

M. Sc. Daniel Simon Maier
Universität der
Bundeswehr München
(Institute of Space Technology
and Space Applications (ISTA))
Werner-Heisenberg-Weg 39
85577 Neubiberg
E-Mail: daniel.maier@unibw.de