What is navigation message authentication?

GNSS Solutions is a regular column featuring questions and answers about technical aspects of GNSS. Readers are invited to send their questions to the columnist, **Dr. Mark Petovello**, Department of Geomatics Engineering, University of Calgary, who will find experts to answer them. His e-mail address can be found with his biography below.

MARK PETOVELLO is a professor (on leave) at the University of Calgary. He has been actively involved in many aspects of positioning and navigation since 1997 and has led several research and development efforts involving Global Navigation Satellite Systems (GNSS), software receivers, inertial navigation systems (INS) and other multi-sensor systems.
E-mail: mark.petovello@gmail.com

Proposals for Navigation Message Authentication (NMA) for both GPS and Galileo are becoming more common. This article describes NMA and its potential impact on future receivers and GNSS users.

As of today, all open civil GNSS signals are transmitted in the clear, conforming to interface specifications that are fully available in the public domain. Receivers will accept any input that conforms to the specifications and treat it as if it came from a GNSS satellite. Combined with the extremely low power levels of GNSS signals this makes it almost trivially simple to spoof a GNSS receiver.

As early as 2003, Logan Scott proposed a number of techniques that could be implemented at the satellite level to "harden" the civil GNSS signals against spoofing attacks. The first and most straightforward amongst these was NMA.

Message Authentication is a concept that has a long history in digital communications, the basic idea being that the receiver of a message would like to ensure that the message they receive: 1) is identical to the message that was transmitted; 2) was generated by a trusted source. NMA is, unsurprisingly, the application of the Message Authentication concept to the navigation messages generated by GNSS satellites.

So How Does NMA Work?

Message authentication has been referred to as the "second face" of cryptology, and it uses many of the same tools and techniques as the more well-known first face of cryptology: cryptography, or data secrecy. In message authentication the sender uses a secret key to generate an authentication signature from the original message. Both message and signature are then transmitted to the receiver, which uses a key (potentially different to that used by the transmitter) to verify that the message and authentication signature correspond.

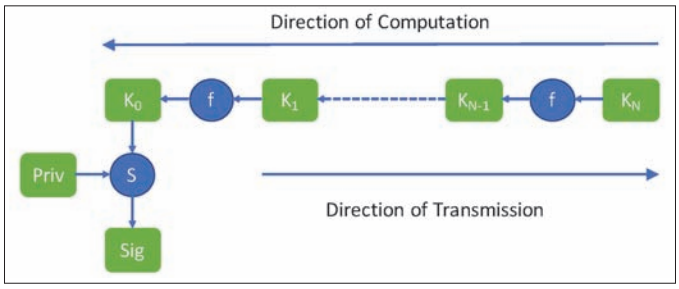
When the received message is authenticated the receiver can conclude that:

1. The transmitted and received message are the same
2. Only someone with access to the transmitter's secret key could have generated the authentication message

There are two different ways to generate authentication signatures:

1. Using symmetric key techniques in which both transmitter and receiver share a secret key
2. Using asymmetric key techniques in which the secret key is split into two parts, a "private" key, known only to the transmitter, and a public key which can be distributed publicly. The private key is used to generate the authentication message, while the public key is used in the verification step.

There are some issues associated with each of the two techniques. In the symmetric key case the most difficult issue is how to distribute the "private" key to all users, without also giving the spoofer access to this key. Similarly, for the asymmetric case, the receiver needs some mechanism to ensure that the "public" key does indeed come from the trusted transmitter (the GNSS system operator in the case of NMA). This



OS NMA chain generation: A chain of keys is generated by first generating a random "seed" key, denoted K_N. The one way function is applied to K_N to produce K_{N-1}, and this process is repeated until the "root" key K_0 is obtained. The root key is signed using the Galileo Private Key (Priv) and the digital signature algorithm (S) to generate the signed root key, Sig. The chain keys are used in the reverse order, i.e. in the order K_1, K_2, …, K_N. Knowing K_M one can confirm that it belongs to the chain with K_0 as root, but cannot determine any future keys K_L, L>M. The digital signature Sig is transmitted by the satellites, and each K_i is transmitted after the MACs that have been generated using it.

problem is usually solved using a Public Key Infrastructure (PKI) consisting of a trusted authority that manages the certification that public keys do indeed belong to the organization that claims them.

So it would appear that the asymmetric approach is superior, as the infrastructure is simplified and the "secret" key can remain secret. However, asymmetric encryption has two major drawbacks: firstly, it is much more computationally intensive than symmetric key encryption; secondly, much longer keys are required for the same level of security.

Interestingly, both symmetric and asymmetric NMA approaches have been proposed for GPS (on the new L1C signal) and Galileo (on the E1 Open Service signal), as discussed below.

The GPS Approach – Asymmetric NMA

The Chips-Message Robust Authentication (Chimera) is a hybrid NMA and spreading code authentication technique proposed for use with the GPS L1C signal. The NMA portion of this scheme is based on the asymmetric elliptic curve digital signature algorithm (ECDSA) P-224, which is a well-established standard. The public key is 448-bits long for an equivalent security of about 112 bits (i.e., it is equivalent to a 112-bit symmetric key system).

The Chimera proposal uses two Subframe 3 pages of the C/NAV message to transmit each digital signature, with a repetition rate of at most once every three minutes. In this way a receiver can verify that the navigation message is authentic every three minutes.

The ECDSA scheme is a well-established Federal Information Processing Standard (FIPS) standard and is implemented in most Open Source and commercially available cryptographic libraries, which simplifies the integration of the scheme into existing GNSS receivers.

Chimera requires receivers to have occasional access, via non-GPS channels, to infrastructure to provide authenticated GPS system public keys. This Public Key Infrastructure (PKI) is essential to any asymmetric cryptosystem, including the Transport Layer Security (TLS) system used in securing websites. In this system, each entity that wishes to provide an authenticated public key obtains a signed certificate from a trusted Certification Authority (CA). A user can then verify that the public key provided corresponds to that in the signed certificate. Reusing this certification process should be straightforward in the GNSS context.

The Galileo Approach – Hybrid Symmetric/Asymmetric NMA

The proposal for Galileo Open Service Navigation Message Authentication (OSNMA) differs from Chimera in that it is based on a hybrid symmetric/asymmetric key approach known as the Timed Efficient Streamed Loss-Tolerant Authentication (TESLA) scheme.

TESLA addresses the issue of symmetric key distribution as follows. First, a Message Authentication Code (MAC) is generated using the message and the private key. Both the message and the MAC are transmitted and then, sometime later, the private key is broadcast. This delayed release mechanism should ensure that the key used to generate the MAC is not known until after the message and MAC are already received. However, this does not prevent a spoofer from simply generating their own messages, keys *and* MACs and broadcasting them in a manner compliant with the specifications.

To address this latter issue, TESLA uses the concept of a *chain* of keys. An initial key K_0 is randomly selected. Each subsequent key in the chain K_{i+1} is generated from the previous key K_i using a one way function: $K_{i+1} = f(K_i)$. A one way function is a mathematical transformation that is easy to compute but very difficult to invert. Thus, given K_i it is easy to compute K_{i+1}, but given K_{i+1} it is computationally infeasible to establish K_i. In TESLA the system generates a chain of length N, then transmits the Nth key (called the root key) along with a digital signature generated using a standard asymmetric scheme, such as ECDSA. The chain keys are then used in reverse order to generate the MACs. Knowing the one-way function, the receiver can verify that each chain key is from the same chain as the digitally signed root key, but cannot predict "future" chain keys.

Once a TESLA chain has been established by asymmetric cryptographic means, the satellites begin transmitting messages, MACs and keys using the delayed release mechanism. The receiver extracts the messages and MACs and stores them until the key is received. The key is first checked to ensure that it is part of the TESLA chain in force using the known one-way function. If the key passes this test, it is then used to verify that the MAC and the message correspond.

There is one absolutely critical assumption that *must* be made for the TESLA-based scheme to work: the receiver must have an authenticated time synchronization that is at least better than the key delay (in TESLA nomenclature this is referred to as the security condition). Without this assurance, the receiver cannot be certain that the navigation message has not been generated by a spoofer that has already received the perfectly valid signing key from a live satellite signal. This naturally raises the question as to how the receiver can "bootstrap" — if it does not have authenticated coarse synchronization how does it go about achieving it?

The OSNMA proposal is currently in draft form and subject to change. As it stands it uses 40 bits every two seconds of the Galileo E1b I/NAV message. The data are grouped into subframes of 30 seconds duration, and each MAC is only 10 to 32 bits in length, while key sizes range from 80 to 256 bits. This enables OSNMA to sign many more messages per second than Chimera, enabling such features as cross-satellite and even cross-system authentication. The receiver side computation cost of this high authentication rate remains to be seen, as it includes both MAC computation and key validation through the evaluation of the one way function. Similar to Chimera, the OSNMA scheme is built from a number of standard cryptographic building blocks, facilitating its implementation in receivers. However, unlike ECDSA the TESLA scheme itself is not a standard implementation and therefore will require more effort to integrate into GNSS receivers.

As with Chimera, the problem of Public Key Infrastructure (PKI) arises for OSNMA. In this case, the receiver must have access to known, trusted public keys in order to authenticate the digitally signed root keys. At present the OSNMA proposal envisages two public key distribution mechanisms: 1) the receiver can be initialized with a valid Galileo system public key in the factory; 2) new public keys can be transmitted over the E1b navigation message, a process known as Over the Air Rekeying (OTAR). In the current draft neither of these techniques are defined, though signal bandwidth is available for the OTAR mechanism. Of course, with OTAR new higher level public keys are required in order to authenticate the root key signing public keys broadcast from the satellites. Ultimately a PKI similar to that used for authenticating public keys used by commercial websites must be put in place to support public key distribution for both OSNMA and Chimera.

What Does This Mean For Users?

NMA is not a panacea, and by itself does not solve the spoofing problem. In Scott's terminology, this is but Level 1 of a series of three system-side, signal level defenses. On the other hand, an NMA scheme that is correctly implemented in the receiver does constrain

the spoofer's "attack space" — essentially the spoofer is constrained to only broadcast valid navigation messages.

From the user's perspective, the desired outcome is most likely to obtain an authenticated PVT, or at the very least, to be able to detect a spoofing attack. NMA on its own is insufficient for PVT authentication since this requires both the navigation message and range measurements from each satellite, and NMA does not authenticate the range. However, certain types of spoofing attack are detectable when NMA is implemented.

Over the last number of months two different "spoofing" attacks have been reported in this magazine (see Additional Resources). The first, almost certainly a repeater, affected a number of ships in the Black Sea in June 2017. The second, not technically spoofing as it was unintentional, was due to a GNSS simulator operating in the exhibition hall at the ION GNSS+ 2017 conference in Portland, Oregon in September 2017.

NMA on its own would have been of no assistance whatsoever in the Black Sea case, assuming that this was in fact a repeater, as this simple attack involves re-broadcast of live GNSS signals in near real-time. The navigation message broadcast by the spoofer would have been identical to that broadcast by the satellites, and hence would have passed the authentication verification step. Such an attack can only be detected by careful monitoring of various signal parameters and looking for sudden, physically impossible jumps, a process known as consistency checking.

The simulator incident is an interesting case, as it shows really how vulnerable many receivers are to spoofing. The cause of the incident appears to have been an improperly terminated output port on a GNSS simulator that was running a demonstration in the exhibition hall. The energy radiated from this port was sufficient to capture the GNSS receivers in many attendees' smartphones. The simulated scenario was located somewhere in Europe and set in January 2014. Clearly, again simply monitoring the existing signal parameters available in the receivers should have provided some indication that there was a problem; jumping across continents or back in time is not something most receivers are capable of. But could NMA have helped in this situation? The short answer is yes, as the simulator would either have generated invalid signatures, or re-used out of date messages. However, in truth, the onus should really be on the receiver to detect sudden, physically impossible jumps in space or time.

In short, for the scenarios considered above, the most reliable test for spoofing is the receiver consistency check.

Summary

Navigation Message Authentication is coming to both Galileo and GPS in the next few years, which demonstrates that the system operators are serious about the civil spoofing threat. Other system level protection measures are also under development: Chimera implements secure spreading sequences for some signal level protection, while Galileo provides fully encrypted spreading codes in its Commercial Service.

NMA is but one tool in the arsenal that GNSS receivers can deploy against spoofing, but is certainly a step in the right direction. However, such efforts on the part of the system providers must be matched by the receiver manufacturers for there to be any benefit to the end user. Many receiver level defenses can, and should, be implemented today without waiting for the arrival of NMA.

Further Reading

Logan Scott's paper on how to harden GNSS receivers:

Scott, L. (2003) "Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems", *ION GNSS 2003*.

Details about the proposed GPS and Galileo NMA proposals:

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

Fernandez, I; Rijmen, V.; Ashur, T.: Walker, P.; Seco, G.; Simon, J.; Sarto, C.; Burkey, D.; Pozzobon, O., "Galileo Navigation Message Authentication Specification for Signal-In-Space Testing", Version 1.0, November 2016, available at: https://www.gsa.europa.eu/development-supply-and-testing-galileo-open-service-authentication-user-terminal-os-nma-gsa.

Additional Reading on the TESLA Scheme:

Perrig, A.; Canetti, R.; Tygar, J.D.; Song, D., "Efficient Authentication and Signing of Multicast Streams over Lossy Channels", *Proceedings of the IEEE Symposium on Security and Privacy, 2000*.

Additional Resources

[1] Goff, S. "Reports of Mass GPS Spoofing Attack in the Black Sea Strengthen Calls for PNT Backup", *Inside GNSS,* 24 July 2017, http://www.insidegnss.com/node/5555.

[2] Scott, L. "Spoofing Incident Report: An Illustration of Cascading Security Failure" *Inside GNSS,* 9 October 2017, http://www.insidegnss.com/node/5661

Biography

Cillian O'Driscoll received his Ph.D. degree from the Department of Electrical and Electronic Engineering, University College Cork, Ireland. He was a senior research engineer with the PLAN group at the University of Calgary from 2007 to 2010. He subsequently spent three years with the European Commission: first at the Joint Research Centre in Ispra, Italy, and later with the European GNSS Programmes Directorate in Brussels. From 2014 to 2017, Dr. O'Driscoll was a researcher in the INFANT Centre at University College Cork. He currently works as an independent consultant, specializing in all areas of GNSS signal processing.