Authenticating GNSS
Proofs against Spoofs
Part 2

GUENTER W. HEIN, FELIX KNEISSL, JOSE-ANGEL AVILA-RODRIGUEZ, AND STEFAN WALLNER

© iStockphoto.com/Cristian Nitu

Securing GNSS systems against unauthorized use and false signals (spoofing) is a matter of growing concern for GNSS operators and users. In this column, the second and final part of a series, the authors explore a variety of methods for user and signal authentication and discuss their application in GNSS.

The emergence of a multi-GNSS world will inevitably require the civil GNSS user community to address the issue of signal authentication: confirming that a pretended identity of a user or transmitted information is, in fact, real and correct.

This two-part column focuses on the concepts and methods for achieving authentication in GNSS operations. In the July/August issue, the column began by introducing some of the cryptographic concepts, terminology, and techniques used to develop and implement authentication methods in navigation systems in general.

This second and final instalment will discuss the possibilities of navigation message authentication, and examine public and private spreading code authentication as well as navigation message encryption and spreading code encryption. We will also draw some conclusions about these concepts' application in GNSS.

Navigation Message Authentication (NMA)

NMA denotes the authentication of satellite signals by means of digitally signing the modulated navigation data. For each satellite, valid signing/validation-key pairs (k_s, k_v) are generated. The signing key k_s is kept secret and is only known by the ground segment and the respective satellite. The validation key k_v is made public in an authenticable manner, for example, by means of certificates in a public key infrastructure.

The navigation message consists of one or several data blocks D_N, containing the orbit and clock parameters, and of one or several data blocks containing the digitally signature D_S. The digital signature D_S is computed, as described in Part 1 of this article, e.g., by hashing the data blocks D_N and subsequently encrypting the hash value under the signing key k_s.

The recipient receives the data blocks D_N and the signature blocks D_S via the modulated data bits on the ranging signal. After receiving a complete message, the recipient can authenticate it by means of the validation function, for example by comparing the hash value of the data blocks D_N with the outcome of the decrypted signature under the publicly known validation key k_v.

This method demands that the validation key k_v is known to the receiver in an authentic manner. This can be achieved by means of an interface to the receiver with which the user can input the validation key. The validation key is published on the Internet merged in

a certificate signed by a trustworthy entity.

As shown in **Figure 1**, however, the key distribution scheme for the upcoming Galileo system is planned in a different way. Rather than publishing on the Internet, the validation key is packed in a certificate and transferred via the modulated data on the ranging signal itself. The receiver proofs the validity of the validation key over a validation chain up to the root certificate. Again, this root certificate has to be known to the receiver in an authentic manner. Thus, the same arrangements have to be implemented as previously described.

Compared to the first approach — using the Internet and a direct interface to the receiver — the data overhead is notably increased. Furthermore, in this proposed Galileo architecture, the authentication method is not available until all certificates are received and validated. In case of direct key input to the receiver, the question of acquisition time does not arise.

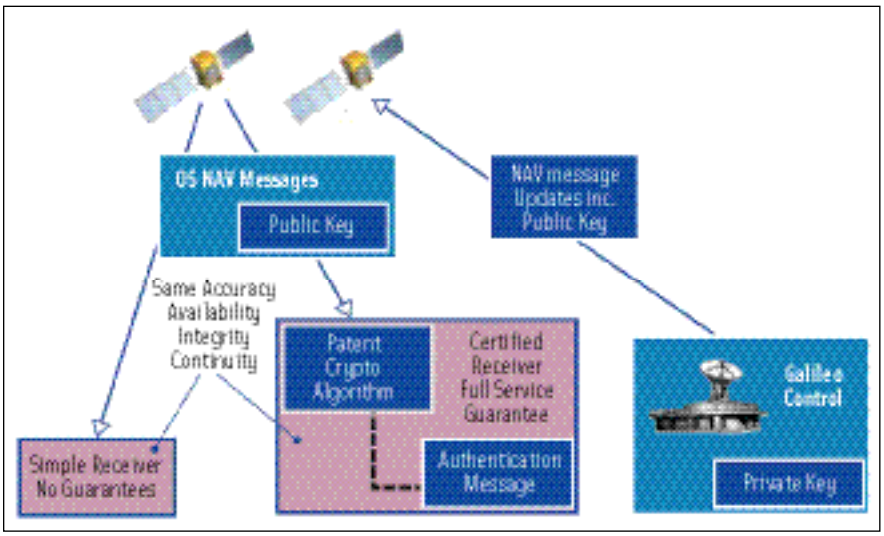

FIGURE 1 NMA key infrastructure proposed by European Commission

Delayed Authentication with NMA

One obvious disadvantage of navigation message authentication is the existence of a delay in authentication. The receiver is only able to authenticate a signal after the reception of the whole navigation message including the digital signature. For example, with a navigation message of 1,500 bits and a digital signature of 500 bits, an authentication delay of 40 seconds occurs at a transfer rate of 50 bits per second (bps). The principle of authentication delay is visualized in **Figure 2**.

The estimation of the authentication delay assumes that it is possible for a spoofer to adopt an authentic signal without being noticed, that is, the spoofing attack does not cause the intended user's receiver to lose tracking. Upcoming studies will investigate the prerequisites — accurate knowledge of the user's position, synchronization issues of signal generators and respective signal transmitters — for a spoofer to substitute an authentic signal by the spoofed replica without attracting attention.

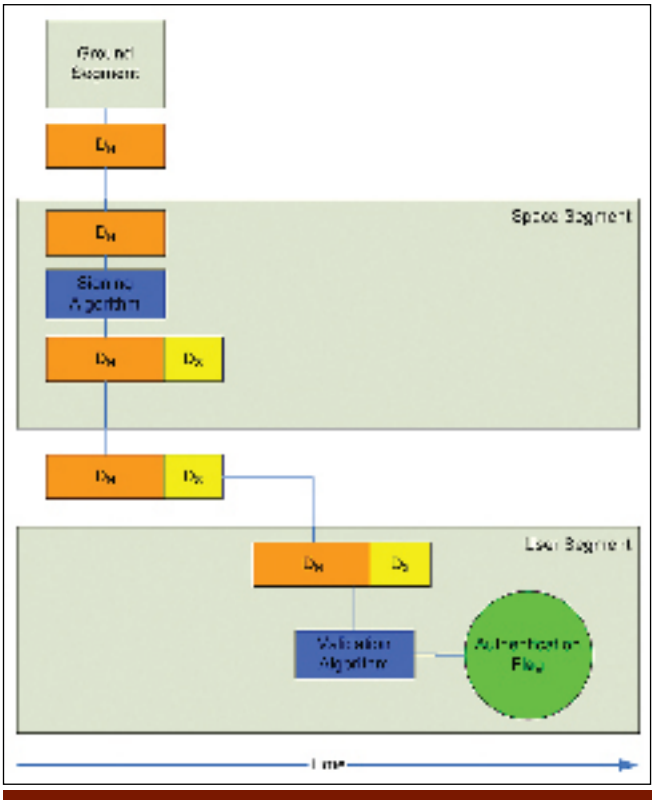

FIGURE 2 . Authentication delay caused by Navigation Message Authentication using digital signature as appendix

In addition to the potential for spoofing, this implementation of NMA cannot meet any of the "time to alert" requirements in civil aviation for ensuring the integrity — or "safe to use" status — of GNSS satellite signals.

Of course, one could individually generate signatures for separate parts of the navigation message. With such an approach, however, the data overhead created by the digital signatures is formidable. Furthermore, one has to take care that every single data block does not reappear within the period of validity of the public key. This, in turn, requires the addition of information — for example, time stamps — to each data block, resulting in an even higher data overhead.

One possible solution for this problem is the use of digital signatures in message recovery mode. In this mode of operation, in contrast to their use as appendices, no signatures are transmitted in addition to the plaintext message. Just the message, encrypted under the signing key, is transmitted to the user. This mode of operation is only practical, however, when the message is short enough to be represented as an element of the set of plaintext messages.

In message recovery mode, the receiver decrypts the transmitted sig-

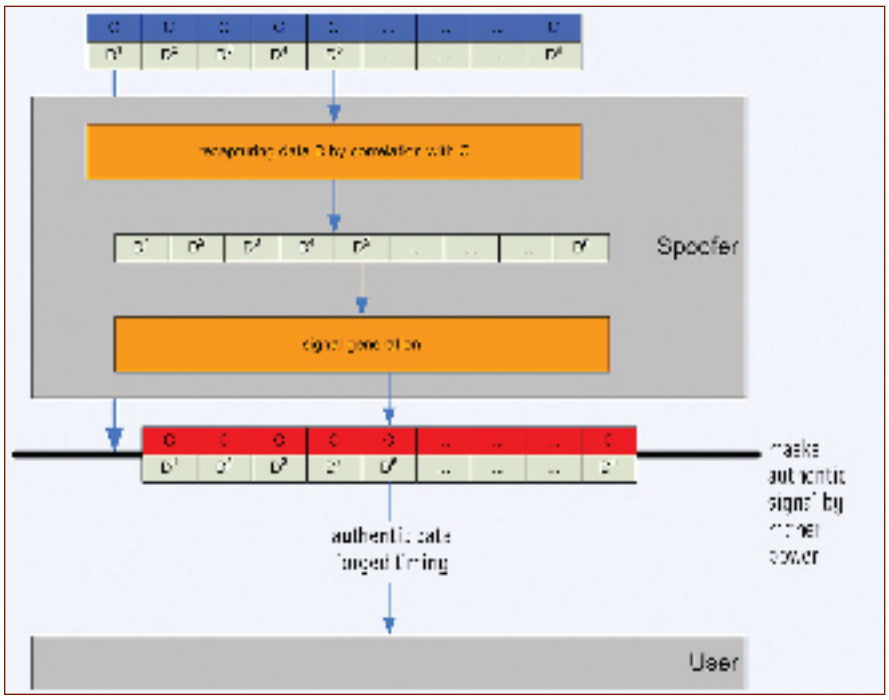

FIGURE 3 Spoofing of NMA protected signals

nature using the validation key. If the outcome of the decryption is a valid navigation message, the navigation message is recaptured and authenticity can be assumed, as it is computationally infeasible for an adversary to find a signature that maps to a consistent navigation message. In order for users to be able to determine whether a navigation message is valid, the message-recovery approach must also apply plausibility tests and/or introduce redundancy in the original message.

To assess the real-word suitability of this approach, let's look once again at the civil aviation requirements for integrity time-to-alarm (TTA). Splitting up the navigation message into blocks of 500 bits and adding another 100 bits redundancy to each block, a transfer rate of 100 bits per second would be adequate to meet the Category 1 (CAT I) approach and landing TTA requirement of six seconds. For this sample, the data overhead due to authentication would only be 300 bits — much smaller than the data overhead in the classical appendix method. We should point out, however, that for certain encryption techniques, such as the use of the RSA algorithm, the size of the single message blocks equals the size of the keys used. Thus, small data blocks result in a less secure authentication scheme and, correspondingly, in a shorter lifetime of the keys.

Spoofing NMA

In addition to the non-achievability of the TTA limits, the main drawback to navigation message authentication is the possibility of by-passing the authentication method by comparatively simple means, as illustrated in Figure 3.

The spoofer receives the authentic navigation signal of the satellite and bitwise reads out the navigation message. The bit stream of the valid navigation message is transferred to the signal generator, which modulates the cryptographically correct data on the forged navigation signal and transmits it with a comparatively high signal power in order to mask the authentic signal. As the spoofer merely interchanges the signal transmission time and sends an identical copy of the navigation message, the receiver is not able to detect the forgery using cryptographic methods.

This forgery is detectable by monitoring the receiver clock bias for sudden jumps. At the very least, the time delay of the forged signal is driven by the amount of time the spoofer needs to process the satellite signal and read out a single bit of the navigation message. This time delay is roughly approximated by the reciprocal transmission rate, thus, on the order of 10 milliseconds.

For a receiver capable of synchronizing its receiver clock with GNSS system time by means of tracking authentic signals, detecting clock bias jumps of this magnitude should present no major problem even if the receiver had not been tracking GNSS signals for up to 10 minutes. However, targeted receivers performing a cold start could easily be fooled by this kind of attack. Moreover, use of spoofing architectures described in the following discussion could fully break navigation message authentication.

Public Spreading Code Authentication

Another approach to preventing spoofs employs the proposed *public spreading code authentication* (PubSCA) described in the paper by L. Scott, cited in the Additional Resources section at the end of this article. This method expands navigation message authentication by adding another security feature. Besides the digital signature of the navigation data, additional codes are inserted into the ranging code in fixed time windows.

These so-called *Spread Spectrum Security Codes* (SSSC or SC) are generated as an enlargement of the digital signature of the present navigation message in the form of pseudorandom bit sequences. In contrast to the intended user and the spoofer, the transmitting satellite knows *ex ante* the complete navigation message and, thereby, also the digital signature. Thus, the satellite is able to compute and transmit the spread spectrum security codes.

As illustrated in Figure 4, the receiver stores the down-converted samples of the spread spectrum security code time windows in a data storage device. After the reception of the complete navigation message and the complete digital signature (at which point the receiver can first compute the spread spectrum security codes), the SCs are generated using the received digital signature as initialization "seed" of the pseudorandom bit generator (PRBG). Then the usual cor-

relation process begins. The correlation power of the replicated and the received SSSC provides a measure for the authenticity of the received signal.

Spoofing PubSCA

A spoofer using standard receiver hardware does not have the capability to read out the spread spectrum security codes, which are buried under the noise floor like all GNSS signals. Furthermore, common signal generators do not have the capability to add or induce user-defined spreading codes in real time.

For these reasons, a spoofer cannot feasibly send a cryptographically correct signal until the reception of the digital signature. The induced time delay of the forged, but cryptographically correct signal is about as large as the transmission time for a complete navigation message including the digital signature. Consequently, the receiver clock jump arising from this delay should be recognized even by receivers that have not been tracking GNSS signals for as long as two days.

With the right equipment, however, a spoofer can thwart navigation signal authentication incorporating PubSCA without creating a substantial — and detectable — time delay. One approach would be to raise the satellite signals above the RF noise floor using directional antennas or beam-forming phased array antennas, down-converting and sampling the signals, buffering the samples, and retransmitting the signals without changing the modulated data but with an appropriate time delay and Doppler shift. This principle of retransmitting signals is a common practice in radar techniques, an implementation known as *digital radio frequency memory* (DRFM).

A further possibility is to determine the public SCs using a real-time software receiver, after having once again raised the signal above the noise floor by adequate means. The resulting codes are then transferred to a signal generator, creating a cloned but time-delayed signal. The clock bias jumps induced by this method primarily arise from the spoofer's computation time and the triangular inequality: the satellite-spoofer-user distance, and thus the signal travel time, is always greater than the satellite-user distance. Nonetheless, such receiver clock jumps are hardly detectable.

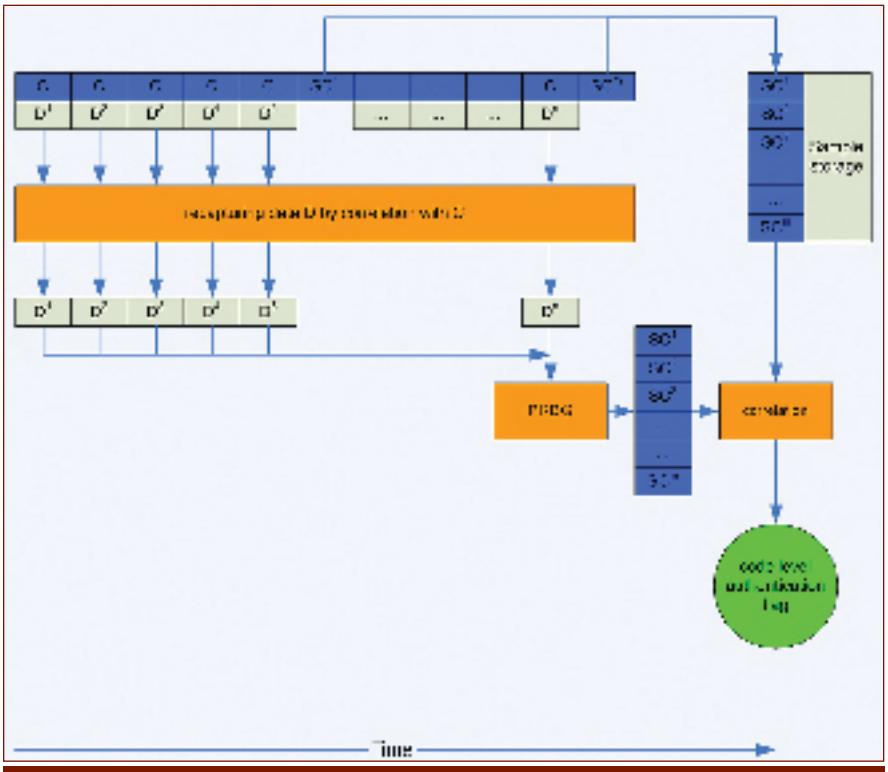

FIGURE 4 Public spreading code authentication processing within the user terminal

Although theoretically possible, the complexity of the equipment needed makes this kind of attack on public SC encryption quite unlikely. In the previously cited paper, L. Scott estimates that even an 80-centimeter diameter antenna would not suffice to determine the spreading codes at an appropriate error level. Due to the high directivity of such antennas, a separate one would be needed for each satellite whose signal was going to be spoofed.

As with simple navigation message authentication, however, NMA with PubSCA suffers an authentication delay. So, a signal can only be authenticated after the reception of a complete navigation message including the digital signature. Consequently, civil aviation time-to-alert limits are not achievable.

Moreover, the changes in the signal characteristics created by adding the PubSCA have other effects, for instance, an influence on the stability of the lock loops, because while SCs are being transmitted, ranging information is unavailable. In turn, this creates a greater sensitivity of the receiver to the effects of high dynamic operations.

Private Spreading Code Authentication

Private spreading code authentication (PrivSCA) is similar to PubSCA in that SSSCs are embedded in the conventional ranging code in fixed time windows. Unlike in PubSCA, where the digital signature of the current navigation message was used to generate the SSSCs, in PrivSCA the digital signature of the last navigation message, encrypted with a symmetrical encryption system using the secret key k_{psca}, is used as the seed for the spreading code sequence generation. (See **Figure 5.**)

The advantage of this architecture stems from the fact that the SCs are known to the receiver after the reception of a complete navigation message, including its signature, at the beginning of the following transmission period. Therefore, authenticity can be achieved in every time window by measuring the correlation power of the replicated and the received SSSC.

Under the assumption that the secret key k_{psca} is indeed confidential

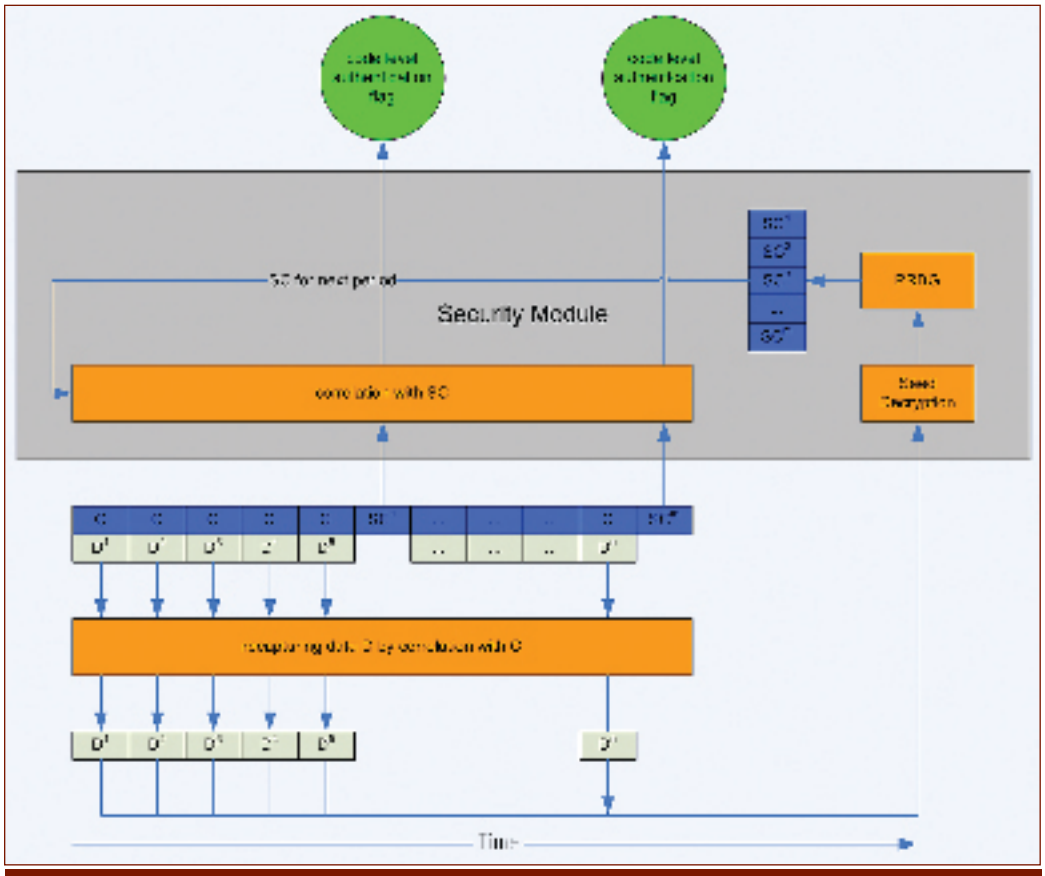

FIGURE 5 Private spreading code authentication processing within the user terminal

and secure, the previously described measures for breaking PubSCA would also have to be implemented to break private spreading code authentication. On the one hand, the proposed PrivSCA architecture demands that the key k_{psca}, which encrypts the digital signature to the initialization seed of the spreading code sequence generator, is available to the receiver. On the other hand, it requires that k_{psca} is secret to the outside world and to the spoofer in particular.

In order to fulfill both requirements, the key must be encapsulated in tamper-resistant hardware. The receiving unit inputs the signature of the last received navigation message into this security module where the seed of the SC is recaptured using the encryption key and the correlation of the replicated and the received SSSC takes place. The output of this correlation process goes to the receiver and provides the indicator of signal authenticity.

As the key k_{psca} has a limited validity period, measures must be implemented to update the key in a secure and authentic manner. One possibility is to assign to each receiver unit an additional symmetric key k_{id_R}, according to a unit number id_R. The key updates are distributed by a trusted entity, which sends to each receiver $E_{k_{id_R}}(k_{psca})$ the new key encrypted by the unit's update key k_{id_R}. The receiver decrypts this information within the security module and gains the new key $k_{psca} = D_{k_{id_R}}(E_{k_{id_R}}(k_{psca}))$. This architecture is similar to the black key/red key architecture of the military GPS.

Navigation Message Encryption

The term *navigation message encryption* (NME) refers to encrypting the data modulated on satellite ranging signals. NME uses symmetric systems for encryption and can provide user authentication, if either the user community is trustworthy (that is, the secret key used for encryption/decryption of the navigation data is not relayed by the entities) or the use of the transmitted data demands the publishing of the data. In the latter case, an unauthorized person could not use the information, even if he is able to decrypt it, because the unauthorized use could then be detected. In this context, NME does not restrict users from the service itself, but from the benefit of the service.

A further possibility for using NME as a method of user authentication is to encapsulate the symmetric encryption/decryption key in tamper-resistant hardware. The receiver inputs the encrypted data to the additional module, where the ciphertext is decrypted. The plaintext message is returned to the receiver. Key distribution issues were discussed in the preceding section.

Spreading Code Encryption

User and signal authentication can be achieved by means of *spreading code encryption* (SCE). In this process, the spreading code used by the satellite is encrypted by modulo 2 addition of a pseudorandom bit sequence. (See **Figure 6**.)

One parameter of SCE is the bit rate of the encryption stream. If the chip rate of the encryption stream is identical to that of the unencrypted spreading code, the modulo 2 addition results in true (pseudo-) random sequences. If the chip rate of the encryption stream is considerably slower than the chip rate of the spreading code, more or less long code sequences result that are known except for the sign. This fact limits the possibility of user authentication because these known code sequences can be used, for example, to perform pseudorange code measurements on the encrypted P(Y)-code on GPS L2.

The pseudorandom bit sequence is an application of stream ciphers. The generation of pseudorandom bit sequences was addressed in the discussion of cryptographic concepts in Part 1 of this column.

In SCE systems such as the GPS P(Y) code, rather than attempting direct acquisition of the encrypted Y-code, the C/A-code is frequently used to acquire a satellite signal and then "hand over" the signal observables for P(Y)-code acquisition and tracking. To limit the computational effort in this technique as well as to limit the computational effort for direct acquisition, the length of the cipher stream is bounded and its generation reset at fixed times.

As a feedback function of the pseudorandom bit generator, a keyed symmetric block cipher such as DES or AES can be used. The time stamp of the first bit of the pseudorandom bit sequence can be used as the initialization seed of the PRBG.

To gain full user and signal authentication from spreading code encryption, the PRBG key must not be knowable by the outside world. Therefore, the confidential information must again be encapsulated in tamper-resistant hardware.

The implementation of this security module is much more complicated compared to navigation message encryption modules and PrivSCA modules. With NME only a few bits need to be decrypted and with PrivSCA only short code sequences have to be decrypted and correlated; for spreading code encryption the whole digital signal processing unit has to be embedded and secured.

The key distribution can be carried out by similar means as PrivSCA and NME. The requirements for a spoofer to break SCE were mentioned in the discussion of public spreading code authentication.

Non-Cryptographic Detection of Spoofers

All of the cryptographic methods for signal authentication that we've presented in this column can be overcome, although the complexity of the necessary spoofing facilities varies. For this reason, we should also consider some non-cryptographic methods with which to achieve signal authentication. These methods can particularly be used to bridge the authentication delay inherent of some of the methods we have described.

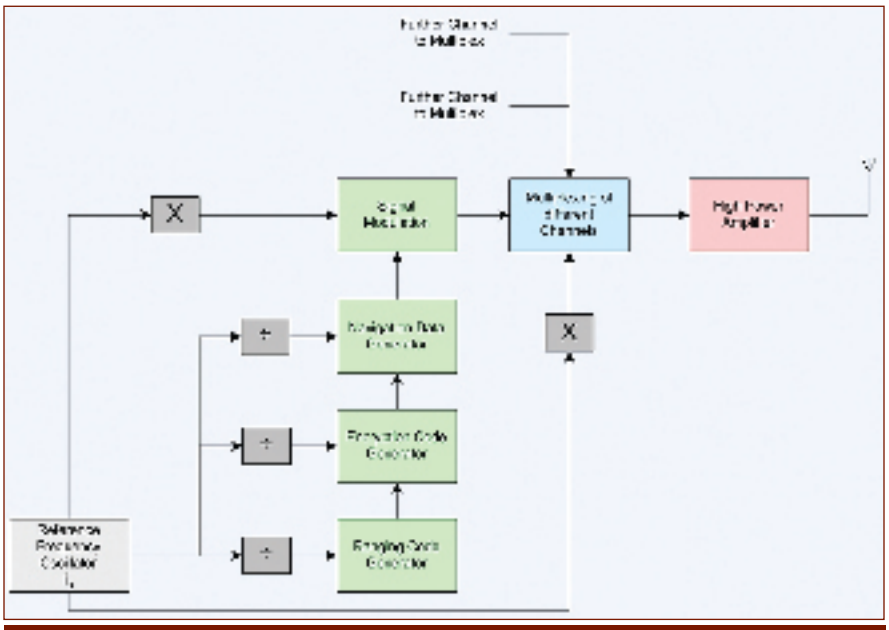

FIGURE 6 Code Generation Block for SCE

We've already mentioned one such approach: the monitoring of the receiver clock bias for unsuspected jumps. Now let's look at some others.

In order to spoof a satellite signal, the forged transmission must mask the authentic signal. For this reason, spoofing signals are broadcast with considerably higher power. So, one possible means for detecting the presence of forged signals is to monitor the absolute power of received signals as well as their signal-to-noise ratio against unexpected behavior.

Usually, a spoofer will send forged signals only from a single location. For this reason, a user can detect spoofed signals by monitoring the direction of signal origin using phased array antennas/receivers. Deviations from the expected satellite geometry are a strong indicator of the presence of forged signals. A further method, simpler to implement but based on the same principle, is the monitoring of a fixed baseline of a two-antenna/receiver system.

The aim of spoofing attacks is to convince a user to be at a different location and/or velocity than the user's intended one. This raises an additional possibility for detecting forged signals — the use of a separate, non-GNSS–based positioning technology. With the aid of sensors such as inertial measurement units, barometric height sensors, odometers, or compasses, a combined navigation system can recognize the drifting of the GNSS position caused by spoofing within the system's integrating filter.

Applying Signal Authentication to GNSS

The cryptographic methods discussed here for user and signal authentication are all implementable measures for future GNSS systems. They differ both in their complexity and security-related aspects. The state of the implementation respective the planning of future implementation is lined out in the following paragraphs.

Modernization of the GPS space segment will add new signals for civil use on L2 (beginning with the Block IIR-M spacecraft currently being launched) and two additional signals for civil use on L5 (beginning with Block IIF). A new civil signal has also been proposed and designed for L1, which would be implemented beginning with GPS Block III satellites. However, no cryptographic methods for authentication are implemented or planned for these civil GPS signals. All military GPS services apply user and signal authentication by means of spreading code encryption.

In the same manner and according to the currently available information,

we can recognize that Galileo will also provide some authentication methods:

- Open Service (OS) — none of the data channels of the OS signals will provide any encryption nor any signal authentication.
- Safety of Life Service (SoL) — all data channels of the SoL signals will provide authenticated integrity data.
- Commercial Service (CS) — the CS signal on the third frequency band, E6, will broadcast encrypted navigation data.
- Public Regulated Service (PRS) — The PRS service will be provided by the E1-A and E6-A signals. These will use encrypted PRS ranging codes and navigation data messages.

Finally, it is important to mention that none of the augmentation systems plans to implement cryptographic methods. So, what this overview reveals is that authentication techniques still have a long way to go until they become usual in all navigation systems. Nevertheless, as this column has made evident, given the great world of opportunities that cryptographic methods offer, we might certainly expect that these techniques will drive future work in this field.

Conclusion

User and signal authentication provides important tools for the growing "GNSS infrastructure." The methods providing user authentication, particularly spreading code encryption, are already in use to protect the military services of GPS. The European Galileo system now under development will use encryption not only for the restriction of the PRS user community but also against unauthorized use of the CS signal on E6 by non–fee-paying users.

The use of NME without an embedded encryption key is only useful for applications in which the benefit of the service demands the publication of the received data. As this kind of application is seldom in use, NME will be implemented with encapsulated keys, albeit with the requisite tamper-resistant hardware and key distribution infrastructures. Whilst the evaluation of the use of user authentication schemes is quite canonical, the choice of appropriate signal authentication schemes is more difficult.

An adequately equipped spoofer — one with beam-forming antennas or highly directive antennas and digital

Acronyms Used in This Column

DRFM	Digital Radio Frequency Memory
FOC	Full Operational Capability
GBAS	Ground Based Augmentation System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
MAC	Message Authentication Code
NME	Navigation Message Encryption
NMA	Navigation Message Authentication
OS	Open Service
PKI	Public Key Infrastructure
PRBG	PseudoRandom Bit Generator
PrivSCA	Private Spreading Code Authentication
PRS	Public Regulated Service
PubSCA	Public Spreading Code Authentication
SBAS	Space Based Augmentation System
SCE	Spreading Code Encryption
SoL	Safety of Life
SSSC, SC	Spread Spectrum Security Codes
RSA	Rivest-Shamir-Adleman

Use It or Lose It!

You are at risk of missing out on future issues of *Inside GNSS* unless you register for a **FREE** subscription at **www.insidegnss.com**

InsideGNSS
GPS | GALILEO | GLONASS | COMPASS

radio frequency memory (DRFM) or real-time software receivers and signal generators — can probably break all suggested cryptographic signal authentication techniques. Such a capability not only demands a tremendous amount of know-how, but the physical dimensions of such a facility would be hard to handle, too. For this reason, in the near future this type of attack will most likely not occur or only rarely be implemented. Thus, we might reasonably assume that only spoofing attacks making use of standard hardware will be the main area of concern.

Due to a capability for nearly instantaneous signal authentication, spreading code encryption and private spreading code authentication offer excellent prospects. However, both procedures are based on the confidentiality of the associated cryptographic keys. Therefore, for both methods, tamper-resistant hardware and a reliable key distribution infrastructure have to be set up. The security of the systems is particularly constrained by the security of the tamper-resistant hardware.

The essential difference between navigation message authentication and public spreading code authentication is that the minimum satellite clock offset for re-emitted (spoofed) cryptographically correct NMA signals is much smaller compared to the minimum satellite clock offset for re-emitted cryptographically correct PubSCA signals. Users having the opportunity to synchronize their receiver clock in a spoofing free environment, e.g. an airport monitored by ground-based augmentation system sensor stations, can detect re-emitting attacks against both authentication procedures. On the other hand, both methods can be broken by a "self-spoofer" — for instance, a participant of GNSS-based tolling system — intentionally disrupting the receiver clock, simply by cutting off the power supply.

Therefore, the additional expenses for the hardware components needed to implement a PubSCA solution, as well as the technique's decreased navigation capabilities, provide a strong argument for only implementing navigation message authentication in a GNSS. The drawback of NMA's authentication delay can be resolved by means of sensor integration or eventually by implementing NMA in message recovery mode (acknowledging the short key length problem).

NMA manifests three advantages compared to SCE and PrivSCA. NMA does not need tamper-resistant hardware in the user terminal, key distribution can be solved very efficiently by means of a public key infrastructure, and NMA does not require dissemination of confidential information — by any means — to the user community.

Therefore, the use of NMA seems to be the technique of choice for GNSS signal authentication purposes. To assure security against DRFM-like attacks, a combination with one or more – non-cryptographic anti-spoofing methods should be performed.

Additional Resources

[1] Dent, A., and C. Mitchell, "Cryptography and Standards," Artech House, Boston, Massachusetts USA, 2005

[2] European Commission, "The Galilei Project: GALILEO Design Consolidation," <http://ec.europa.eu/dgs/energy_transport/galileo/doc/galilei_brochure.pdf>, 2003

[3] European Space Agency/Galileo Joint Undertaking, *Galileo Open Service Signal In Space Interface Control Document* (OS SIS ICD) Draft 0 23/05/2006, <http://www.galileoju.com>, 2006

[4] Federal Aviation Administration, "Category I Local Area Augmentation System Ground Facility," FAA-E-2937, Washington, D.C. USA, 2002

[5] Hein, G., and J. Godet, J.-L. Issler, J.-C. Martin, P. Erhard, R. Lucas-Rodriguez, and T. Pratt, "Status of Galileo Frequency and Signal Design," ION GPS 2002, Portland, Oregon USA, 2002

[6] Oppliger, R., *Contemporary Cryptography*, Artech House, Boston, Massachusetts USA, 2005

[7] Scott, L., "Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems," ION GPS 2003, Portland, Oregon USA, 2003

[8] Wullems, C., and A. Pozzobon and O. Pozzobon, "Signal Authentication and Integrity Schemes for Next Generation Global Navigation Satellite Systems," GNSS 2005, Munich, Germany, 2005

[9] Wullems, C., and O. Pozzobon and K. Kubik, "Secure Tracking using Trusted GNSS Receivers and Galileo Authentication Services," *Journal of Global Positioning Systems*, Vol. 3 No. 1-2, 2004

Authors

"Working Papers" explore the technical and scientific themes that underpin GNSS programs and applications. This regular column is coordinated by **PROF. DR.-ING. GÜNTER HEIN.** Prof. Hein is a member of the European Commission's Galileo Signal Task Force and organizer of the annual Munich Satellite Navigation Summit. He has been a full professor and director of the Institute of Geodesy and Navigation at the University of the Federal Armed Forces Munich (University FAF Munich) since 1983. In 2002, he received the United States Institute of Navigation Johannes Kepler Award for sustained and significant contributions to the development of satellite navigation. Hein received his Dipl.-Ing and Dr.-Ing. degrees in geodesy from the University of Darmstadt, Germany. Contact Prof. Hein at <Guenter.Hein@unibw-muenchen.de>.

Felix Kneissl studied at the Technical University of Munich and graduated with a diploma in mathematics. He is now a research associate at the Institute of Geodesy and Navigation at the University of the Federal Armed Forces in Munich. His main subjects of interest are in the context of integrity.

José-Ángel Ávila-Rodríguez is a research associate at the Institute of Geodesy and Navigation at the University FAF Munich. He is responsible for research activities on GNSS signals, including BOC, BCS, and MBCS modulations. Ávila-Rodríguez is involved in the Galileo program, in which he supports the European Space Agency, the European Commission, and the Galileo Joint Undertaking, through the Galileo Signal Task Force. He studied at the Technical Universities of Madrid, Spain, and Vienna, Austria, and has an M.S. in electrical engineering. His major areas of interest include the Galileo signal structure, GNSS receiver design and performance, and Galileo codes.

Stefan Wallner studied at the Technical University of Munich and graduated with a diploma in techno-mathematics. He is now research associate at the Institute of Geodesy and Navigation at the University FAF Munich. Wallner's main topics of interests are the spreading codes and the signal structure of Galileo and also interference and interoperability issues involving GNSS systems. IG