Authenticating GNSS Proofs against Spoofs Part 1

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

Concerns about the authenticity and security of GNSS signals are usually associated with military applications. On the one hand, military GNSS users need to detect and avoid spoofing – the generation of false ranging signals by an adversary to mislead a foe – while on the other preventing the exploitation of one's own GNSS broadcasts by an enemy. But a growing number of civil safety- and security-related applications also require these capabilities: GNSS-based electronic toll collection, aviation, financial transactions and so forth. However, the ability to authenticate signals or deny them to unauthorized users is currently lacking in open civil GNSS services. This two-part column explores the realm of cryptographic tools that can be used to ensure that only authorized users have access to GNSS services and that the positioning information they use – and report – is the real thing.

Authentication is an essential problem in the field of communication: confirming that a pretended identity of a user or transmitted information does, in fact, really correspond to the true identity or source.

In the context of a face-to-face conversation, we can authenticate the communication by recognizing the voice and the look of one's conversational partner. A letter can be authenticated by identifying the handwriting of its sender, and in small, wired local area networks the originator is apparent by following the cable connections. This situation changes fundamentally in the scope of modern communication over wide area networks or radio links.

Circumstances also vary the importance of authentication. We may not care that the author of a cooking recipe is indeed Paul Bocuse, as long as the ingredients and preparation instructions produce an excellent meal. On the other hand, it is important to know that the originator of an application for payment is indeed the plumber who repaired the toilet or the electrician who installed the new light fixtures.

Even more important is authenticity in matters of safety- and security-related information. For example, in the handling of a railway level crossing, system operators must be assured that only the signalman is able to raise the gate.

Authentication in GNSS

In the context of satellite navigation systems, the transmission of information takes place over radio links — by their very nature, an insecure data channel.

GNSS communications contain several areas of concern regarding the confirmation of the origin of a message and/or the identity of its sender. In matters involving user segment data links, two different kinds of authentication issues arise: first, confirmation that a navigation signal actually originates from the indicated satellite and, second, proof that the user receiver is authorized to make use of the signal.

Both proofs of identity are relevant for the security of military applications of GPS. On the one hand, it must be assured that a potential foe is not able to gain any benefit from the system, for example, by aiding the navigation of the foe's cruise missiles with GPS signals. On the other hand, authorized GPS military users need to guarantee that misrouting of one's weapons by means of spoofed signals does not occur.

Spoofing denotes the misguiding of users by means of forged signals and manifests itself as an error source in both military and civil GNSS applications. In safety-critical civil realms such as aviation, detection of forged signals must be reliable.

Both mechanisms are implemented in the GPS for military users since the Full Operational Capability (FOC) by means of encrypting the precision or P-codes to generate a Y-code that can only be processed by authorized users with keyed receivers. (We'll have much more to say about cryptographic "keys" later in this column.) In contrast, civil GPS users cannot currently authenticate received signals of the open C/A-codes on L1 and the new civil signal on L2 — at least not by using intrinsic capabilities of the GPS system itself.

The implementation of new global satellite navigation systems in the coming years will result —by the increased number of available satellites alone — in an improvement of the performance parameters for "accuracy," "integrity," "availability," and "continuity." For this reason, civil GNSS applications will be used more and more often in safety- and security-related fields. Perhaps the most familiar example is GNSS-only guidance of aircraft in approach and landing at the so-called Category 1 (CAT-1) level or higher.

The emergence of a multi-GNSS world will, therefore, inevitably require the civil GNSS user community to address the issue of signal authentication, too. In this two-part column we will discuss authentication in GNSS. The column begins by introducing some of the cryptographic concepts and terminology used to develop and implement authentication methods in the field of navigation systems. In Part 2, the presentation describes the authentication methods and the design, benefits, and drawbacks of the particular techniques. An overview of the current state and planning of GNSS authentication methods will be followed by a few short conclusions.

Two Sides of Authentication

In the context of GNSS, two basically different kinds of authentication arise:

- authentication of a user and the accompanying potential need for restriction of user access, i.e., selective availability
- signal authentication.

The term *user authentication* addresses the proof of the identity of a user against a monitoring entity. As satellite navigation systems are based upon one-way directional communication, many classical authentication methods cannot be used. Indeed, the purpose of user authentication is to prevent unauthorized entities from using the service. For this mechanism, cryptographic methods are available to cipher the communication channel, for example, the underlying GNSS spreading codes.

The encryption of the modulated message (navigation message encryption or NME) only restricts an unauthorized user from receiving the modulated data; the signal itself can still be used for pseudorange measurements without any constraints. If the spreading codes are encrypted, however — depending on the chip rate of the encryption stream — some techniques can still be implemented to use the signal for ranging, but all of them result in a decreased ranging capability, for example, with respect to dynamics.

Unlike user authentication, signal authentication refers to the proof that a received signal indeed originates from the claimed source (e.g., a navigation satellite, satellite-based augmentation system or SBAS satellite, or a ground-based augmentation system — GBAS — transmitter). Authenticable signals thus allow a user to recognize forged signals.

Cryptographic Terms and Concepts

In the following section, we will present a variety of cryptographic terms and procedures that will be used later on in our discussion of authentication techniques. The objective of this section is not to establish a mathematically stringent definition of the topics, but rather to give an overview of the concepts and tools used in cryptography. The presentation follows in shortened form the schema outlined in the publication by R. Oppliger cited in the Additional Resources section near the end of this column.

One-way Functions. This concept plays an essential role in modern cryptography. It refers to a function that can always be readily evaluated but for which it is not feasible to find preimages efficiently, which would leave encrypted navigation messages or digital signatures vulnerable to being broken. Thus, a function $f : X \rightarrow Y$ is said to be one-way, if

- the function f is easy to compute in the sense that $f(x) \in Y$ can be calculated efficiently for all $x \in X$. A polyomial time algorithm A can deliver $A(x) = f(x)$ for all $x \in X$, and
- the function f is hard to invert in the sense that it is not known how to compute $f^{-1}(f(x))\ \forall x \in X$ efficiently. In other words, no polynomial time algorithm A exists that can deliver $A(y) = f^{-1}(y)$ for all $y \in f(X)$.

The inverting algorithm is not required to find a specific preimage but only to find an arbitrary preimage. For injective functions, however, the preimage is unique.

As will be discussed later, one-way functions are, among others, fundamental to cryptographic hash functions.

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

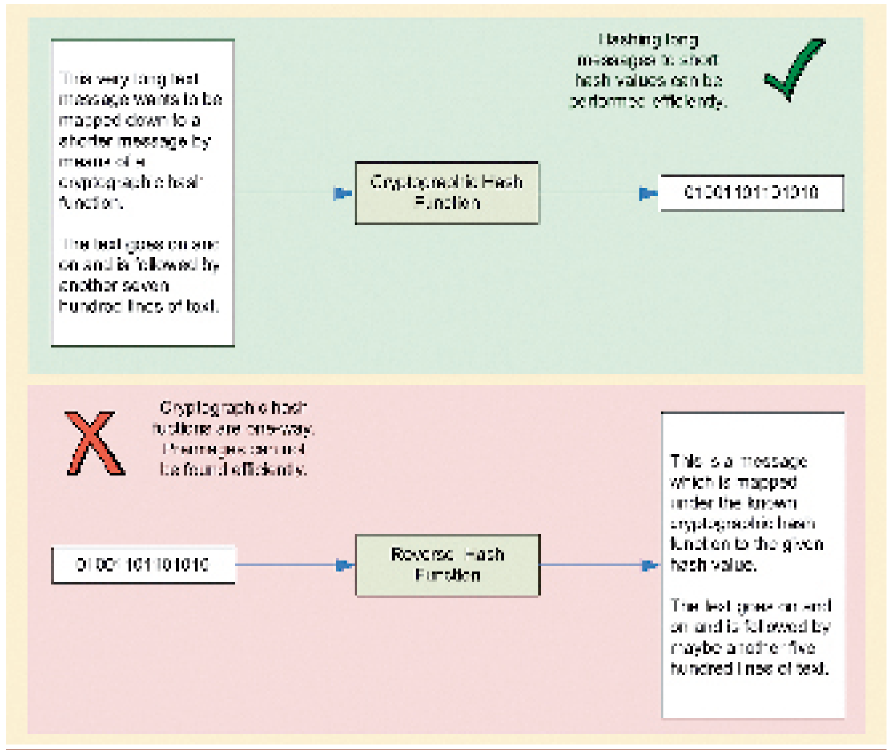

This very long text message wants to be mapped down to a shorter message by means of a cryptographic hash function.
The text goes on and on and is followed by another seven hundred lines of text.
Cryptographic Hash Function
0100110110101010
Hashing long messages to short hash values can be performed efficiently.
Cryptographic hash functions are one-way. Preimages cannot be found efficiently.
0100110110101010
Reverse Hash Function
This is a message which is mapped under the known cryptographic hash function to the given hash value.
The text goes on and on and is followed by maybe another five hundred lines of text.

FIGURE 1 Cryptographic hash functions: preimage-resistant reductions

Trapdoor Function. This term refers to a one-way function for which some extra information exists that, when employed, makes it possible to invert the function efficiently. Thus, the function $f : X \rightarrow Y$ is called a trapdoor function, if

- the function f is a one-way function and
- the function f is, together with some extra information t — also called *trapdoor information* — efficiently to invert. A polynomial time algorithm I exists that delivers $\tilde{x} = I(f(x), t)$ with $f(\tilde{x}) = f(x)$.

Again, the inverting algorithm is only required to find an arbitrary, not a specific preimage. An example of a trapdoor function is the RSA algorithm used in asymmetric encryption systems.

Hash Functions. The reductionist quality of hash functions enables the efficient mapping of messages of arbitrary length to messages of fixed length. For example, let Σ be an alphabet comprised of the numbers 0 to 9. The function $h : \Sigma^* \rightarrow \Sigma^1$, $h(x) = x_1$ then defines a hash function mapping every input value x to the value of its first decimal place. This simple kind of hash function is often applied in everyday life. For example, the hash function defined by "map some name down to its first letter" is used to sort telephone numbers in a notebook.

Cryptographic hash functions h have, in additional to the basic hashing property, several other important properties. It has to be computationally infeasible to find for any given hash value a corresponding message mapping to this hash value. Moreover, it has to be computationally infeasible to find for any given message, a further (distinct) message that results in the same hash value.

If a cryptographic hash function proves computationally infeasible to find two arbitrary messages that map to the same hash value, that hash is said to have strong collision resistance. Thus, a hash function $h : \Sigma^* \rightarrow \Sigma^n$ is a cryptographic hash function, if

- the hash function is preimage resistant, that is, it is computationally infeasible to find an input message $x \in \Sigma^*$ that maps to $h(x) = y$ for a given hash value $y \in h(\Sigma^*)$ and
- the hash function is second-preimage–resistant, which makes it computationally infeasible to find a second input word $\tilde{x} \in \Sigma^*$ with $x \neq \tilde{x}$ that maps to the same hash value $h(x) = h(\tilde{x})$ for any given input word $x \in \Sigma^*$.

Furthermore, a cryptographic hash function is called collision resistant if it is computationally infeasible to find two arbitrary but distinct messages $x, \tilde{x} \in \Sigma^*$ with $x \neq \tilde{x}$ that map to the same hash value $h(x) = h(\tilde{x})$.

As one can see, the condition of the preimage resistance implies that cryptographic hash functions are one-way functions. The principle of cryptographic hash functions is visualized in Figure 1.

Furthermore, (keyed) families of cryptographic hash functions have an application in message authentication codes (MACs). If sender and receiver share a common but secret key, messages can be authenticated by sending — in addition to the message — the hash value of the message under the keyed cryptographic hash function. The receiver computes the hash value applying the secret key and gains authenticity thus by comparing it with the received hash value.

Common representatives of cryptographic hash functions include the MD4 (Message-Digest algorithm 4), MD5, and SHA (Secure Hash Algorithm).

Pseudorandom Bit Generator. A PRBG is an efficient deterministic algorithm that maps an arbitrary bit sequence of a certain length to a bit sequence of considerably greater length that appears to be random. To construct pseudorandom bit sequences, the initialization seed of the PRBG has to be generated non-deterministically. In effect, a PRBG acts rather as a "randomness expander": The output of a cryptographic PRBG has to be, at least under the assumption of bounded computational power, indistinguishable from a truly random bit sequence.

Thus, a function $p : \{0,1\}^* \rightarrow \{0,1\}^*$ is a cryptographic pseudorandom bit generator, if $|p(s)|>|s|$ holds for all $s \in \{0,1\}^N$ and $p(s)$ is pseudorandom, that is, $p(s)$ is under the assumption of bounded computational power indistinguishable from a truly random bit sequence v.

$|\bullet|$ denotes the length of a bit sequence and s, v are random bit sequences with $|v| = |p(s)|$.

The first condition addresses the expanding character of pseudorandom bit generators. The second condition names the pseudorandomness of the output sequence.

In order to appear random, the output sequence has to pass basic statistic tests. Any arbitrary n-tupel has to occur at nearly the same probability, e.g. one can count nearly the same number of occurrence of the 2-tupels [1,1], [1,0], [0,1] and [0,0] from a pseudorandom sequence. A further possibility for assessing the randomness of a bit sequence is to examine it for compressibility: if the sequence is compressible (e.g., by gzip), it contains redundancy and therefore appears not to be random.

Pseudorandom bit generators are implemented as finite state machines. Figure 2 illustrates the principle of a PRBG.

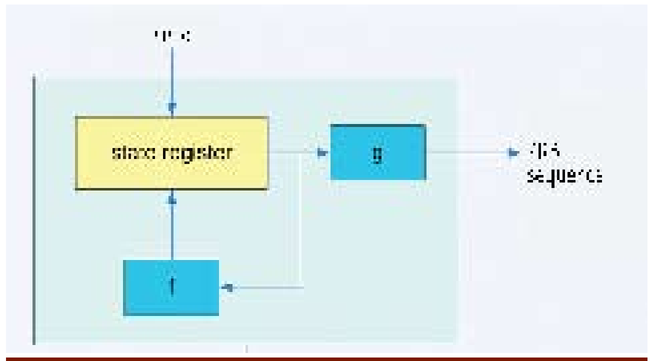


FIGURE 2. Model of a finite state machine PRBG

Pseudorandom bit generators find application in symmetric encryption systems, namely in additive stream ciphers.

Symmetric Encryption Systems. These systems enable the encryption and decryption of messages under the assumption that the sender and the receiver possess a common key, which is secret to the outside world. (See Figure 3.) Symmetric encryption systems consist of the following components:

- the space of the plaintext messages M,
- the space of the ciphertext messages C,
- the space of usable keys K,
- the (keyed) family of encryption functions $E = \{E_k : k \in K, E_k : M \rightarrow C\}$ and
- the (keyed) family of decryption functions $D = \{D_k : k \in K, D_k : C \rightarrow M\}$.

To benefit from a symmetric encryption system, it has to hold that $D_k(E_k(m)) = m \; \forall m \in M$ and $\forall k \in K$ — the decryption function is inverse with respect to the encryption function. Moreover, without knowledge of the secret key, both the encryption function and the decryption function have to be one-way. Thus, the objective of symmetric encryption systems is to make it impossible to move from the ciphertext to the plaintext without knowledge of the secret key.

We can assess the relative security of symmetric encryption systems in terms of the following types of attacks.

- Ciphertext-only attacks: the adversary only knows one or several ciphertexts and tries to determine the corresponding plaintexts or keys. An encryption system that is vulnerable to this kind of attack is totally insecure.
- Known-plaintext attacks: the adversary knows one or several pairings ciphertext/plaintext and tries to determine the key used for encryption or to decrypt unknown plaintext from some ciphertext.
- Chosen-plaintext attacks: the adversary is able to encrypt arbitrary plaintext messages under the key used for a cryptographic system. By means of this, the adversary can create plaintext/ciphertext pairings and uses this information to determine the encryption key or to decrypt unknown plaintext from some ciphertext.
- Chosen-ciphertext attacks: the adversary can decrypt arbitrary ciphertext messages under the used key. By doing this, the adversary is able to create plaintext/ciphertext pairings and uses this information to determine the key used for encryption or to find valid ciphertext messages for some plaintext messages.

Besides the security issue of a cryptographic system's resistance against the types of attacks described here, other criteria can help decide on the choice of symmetric encryption systems, such as the size of the key space and the computational effort of encrypting and decrypting.

Symmetric encryption systems are usually categorized as either block ciphers or stream ciphers.

Block Ciphers. These methods for symmetric encryption do not encrypt or decrypt messages sign by sign but rather packages of signs in one step. Both the plaintext space M and the ciphertext space C consists of all n-tupels over the alphabet Σ.

Because the plaintext space and the ciphertext space are the same sets, one way to encrypt and decrypt messages is by defining a permutation table for all possible messages. The number of possible messages increases with the block

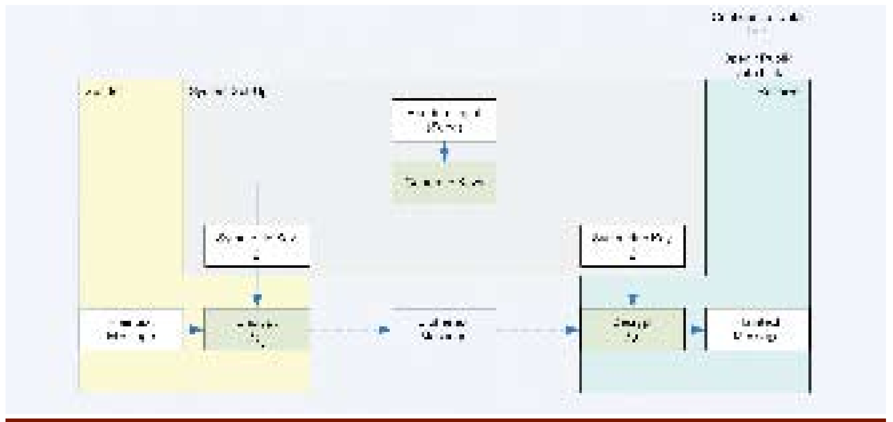


FIGURE 3 Principle of symmetric encryption systems

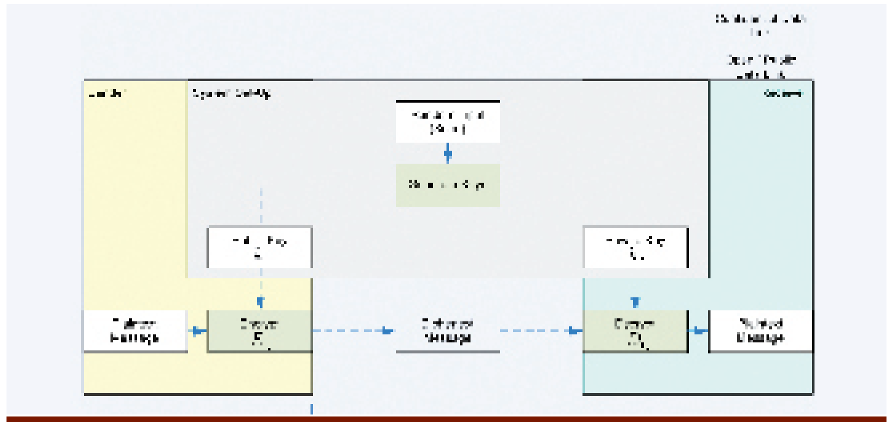


FIGURE 4 Principle of asymmetric encryption systems

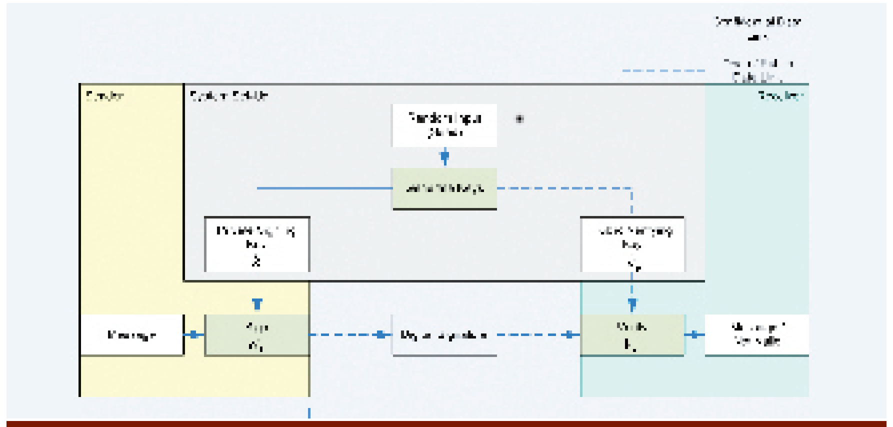


FIGURE 5 Principle of digital signature systems with appendices

length and the size of the alphabet by $|\Sigma|^n$. Every representative of the family of encryption functions $E_k : \Sigma^n \to \Sigma^n$ is given by one specific permutation of these values. Altogether, the family of encryption functions by means of permutation consists of $(|\Sigma|^n)!$ members. For a block length of 64 signs in the binary alphabet, one would need more than 2^{69} bits to encode the key.

As the size of the resulting key space is not manageable, only subsets of the set of all permutations are used. In common symmetric encryption systems, combinations of alphabet substitutions and permutations are used to reach the goals of “confusion” (the complex relation between plaintext and ciphertext) and “diffusion” (the broad influence of one single plaintext sign on the ciphertext).

Well-known representatives of symmetric encryption systems are DES (Data Encryption Standard) and AES (Advanced Encryption Standard).

Stream Ciphers. In contrast to block ciphers, stream ciphers encrypt and decrypt messages sign by sign. For this reason, a cipher stream k of the length of the plaintext message is generated from a starting key $k_{start} \in K = \Sigma^n$ of reasonable a length n. The encryption of a message $m \in \Sigma^*$ in dependency of the starting key $k_{start} \in K = \Sigma^n$ is carried out by adding the generated cipher stream k to the plaintext message, as follows:

$$Ek_{start}(c) = m_1 + k_1, . . . ,m_l + k_l$$

In the same manner, the ciphertext is decrypted:

$$Dk_{start}(c) = c_1 + k_1^{-1}, . . . ,c_l + k_l^{-1}$$

The generation of the cipher stream k from the key k_{start} can, e.g., be performed with pseudorandom bit generators. If the messages are sequences in the binary alphabet, the encryption function and decryption function are identical, as every element in $\Sigma = \{0,1\}$ is its own inverse.

The most famous stream cipher is ARCFOUR, which was used to secure wLAN connections until it was recognized to be insecure.

Asymmetric Encryption Systems. Unlike symmetric encryption systems that use the same key for encryption and decryption, *asymmetric encryption systems* use two distinct keys for encryption and decryption (Figure 4). Symmetric encryption systems require the distribution of confidential keys to the sender and receiver in a secure manner. In many applications, secure key distribution is impossible or, at least, very complicated.

Asymmetric encryption systems avoid this problem by the use of a public key, allowing everybody to encrypt messages, and a private key, which is used to decrypt the encrypted messages.

Asymmetric encryption systems consist of the following components:

- the space of plaintext messages M
- the space of ciphertext messages C
- space of usable encryption keys K_e
- space of usable decryption keys K_d
- the (keyed) family of encryption functions $E = \{E_{k_e} : k_e \in K_e,\ E_{k_e} : M \to C\}$ and
- (keyed) family of decryption functions $D = \{D_{k_d} : k_d \in K_d,\ D_{k_d} : C \to M\}$.

To benefit from an asymmetric encryption system, the encryption function and the decryption function have to be inverse one-way functions, thus:

$D_{k_d}(E_{k_e}(m)) = m$ holds for all $m \in M$ and valid key pairs $(k_e, k_d) \in K_d \times K_e$.

The split in an encryption key and a decryption key, together with the disclosure of the encryption key, enables chosen-plaintext attacks. Security assessments of asymmetric encryption systems use the methods of complexity theory and assume computationally bounded attackers.

The most common asymmetric encryption systems are RSA, Rabin, ElGamal, and ECC (Elliptic curve cryptography).

Digital Signature Systems. These methods allow the authentication of messages through use of asymmetric encryption systems in which the sender and recipient do not need to possess a common, but secret, key.

Digital signature systems consist of the following components:

- space of the plaintext messages M,
- the space of signatures S,
- the space of usable signing keys K_s
- space of usable validation keys K_v
- the (keyed) family of signing functions $S = \{S_{k_s} : k_s \in K_s,\ S_{k_s} : M \to S\}$ and
- the (keyed) family of validation functions $V = \{V_{k_v} : k_v \in K_v,\ V_{k_v} : M \times S \to \{true, false\}\}$.

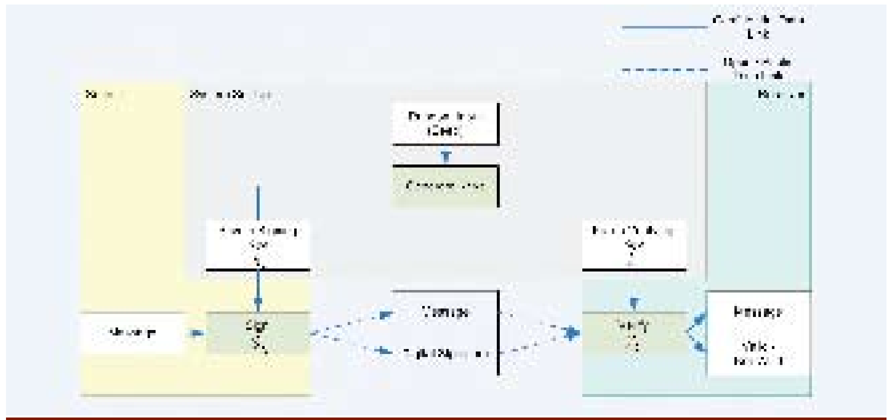


FIGURE 6 Principle of digital signatures with message recovery

In the scope of digital signature systems, the private decryption key plays the role of the signing key, and the public encryption key plays the role of the validation key (Figure 5). To digitally sign a message, the sender transmits the plaintext message and the signature. For example, this can be accomplished by hashing the plaintext message under a certain cryptographic hash function and subsequently encrypting this hash value under the private signing key (until now, decryption key) of an asymmetric encryption system.

Both steps are merged into the signing transformation. The receiver itself computes the hash value of the plaintext message, too. This value is then compared to the outcome of the decryption of the signature under the public validation key (until now, encryption key) of an asymmetric encryption system. Equality of both values leads to authenticity and integrity of the message. Again, both steps are merged in the validation transformation.

The most common representatives are categorized by the used asymmetric encryption system: RSA, Rabin, ElGamal, and ECDSA (Elliptic Curve Digital Signature Algorithm).

This categorization does not determine a digital signature system, as the cryptographic hash function to use is not fixed. One example of a digital signature system is the combination of the SHA-1 hash function and the RSA cryptosystem.

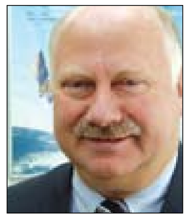
For some procedures and sufficiently small messages, one can digitally sign messages in recovery mode (see Figure 6). In this procedure, the message itself is encrypted under the signing key, transmitted to the recipient, and decrypted under the validation key. If the outcome of the decryption is a valid message, authenticity can be assumed. The problem of recognizing a received message as valid can be solved by means of plausibility tests or by means of redundancy introduced *ex ante*.

In the September/October issue of *Inside GNSS*, Part 2 of this Working Papers column will discuss the possibilities of navigation message authentication, and examine public and private spreading code authentication as well as encryption. We will also draw some conclusions about these concepts’ application in GNSS.

Additional Resources

R. Oppliger, R., *Contemporary Cryptography*, Artech House, Boston (MA) USA, 2005

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 Commis-sion, 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