

**Method of decoying of a system for interception and jamming by
inserting dummy synchronization patterns into the signal emitted and
emitter implementing the method**

5 The technical field of the invention is that of fighting against jamming of telecommunication systems and intrusion in said systems. The aim of the invention is to provide a solution allowing the prevention of interception and/or jamming of a signal transmitted by a piece of transmitting equipment, such as a telephone or a portable terminal.

10

 The invention relates to a method for decoying an interception and jamming system by inserting fake synchronization patterns into the signal transmitted by the transmitting equipment. The fake patterns inserted are intended to disturb the analysis of the signal from the transmitter by the
15 interception and jamming system and to break up both the computation power of its interception function and the power of the jamming signal. The fake synchronization pattern(s) are introduced into the transmitted signal by generating an adapted binary sequence directly in the binary data sequence to be transmitted.

20 The invention does not require any modification of the transmitting equipment because it operates at the input of the transmission chain.

 When generating the signal to be transmitted, communication systems use special sequences, inserted into the signal, that are used to synchronize
25 the transmitting and receiving equipment with one another. These synchronization sequences are fixed once and for all by the communication protocol or the implementation standard of the system and are inserted into the signal to be transmitted with the modulated information sequences.

30 In the field of jamming, one objective is to neutralize a communication system by transmitting, at the same radio frequency, a high-power signal and

thus to bring about a loss of synchronization in the communication system. An interception and jamming system has interception and analysis functions that can be used to detect a synchronization sequence in a signal transmitted by a transmitter, and then to jam precisely this sequence in a coherent
5 manner vis-à-vis a receiver, so that the synchronization between the transmitter and a receiver that is compatible with the transmitter is no longer possible.

One problem to be solved concerns the development of a decoy
10 solution implemented in a transmitter allowing the jamming of the synchronization sequences in the transmitted signal to be prevented or making it more complex and/or less effective. An associated secondary problem is to design a solution that is not very complex, from the point of view of its implementation, and that requires little or no modification of the
15 transmitting and/or receiving equipment of the communication system. In particular, the solution must allow the preservation of compatibility with the telecommunication standard implemented and protection of the synchronization between transmitters and receivers despite the presence of jammers.

20

Various methods are known to protect communication from external disturbances due to electromagnetic interference or to intentional jamming. Notably, the techniques of Direct Sequence Spread Spectrum, Frequency Hopping or Time Hopping can be cited.

25

Moreover, spatial anti-jamming methods using antenna processing are also known that can be applied to receivers and, if need be, to transmitters equipped with antenna networks.

30

All of these methods are implemented on the low-level layers, notably the physical layer, of a modem and thus require modification of the waveform

or the integration of additional processing on the transmitted or received signal. Thus, these methods require a specific design that affects not only the onboard software in the transmitting and receiving stations but also the hardware components such as the radio system or the antennas.

5

The aim of the invention is to solve the aforementioned problems and to remove the limitations of the prior art solutions by proposing a method for decoying interception and jamming systems that involves producing specific encoding of the useful data to be transmitted so as to indirectly generate, in
10 the ultimately transmitted signal, one or more fake synchronization patterns in the form of steady-state patterns. The implementation of the method according to the invention at the useful data to be transmitted rather than at the output of the transmission chain makes it possible not to alter the properties of the transmitter vis-à-vis compliance with the telecommunication
15 standard used.

The subject matter of the invention is thus a method for decoying against an interception and/or jamming system, involving generating, in a signal to be transmitted, at least one fake sequence so as to decoy said
20 system while preserving the synchronization with a receiver of said transmitted signal, said method being characterized in that it comprises the following steps:

- defining said fake sequence and its time and/or frequency position within the signal to be transmitted, the values of the symbols of
25 said fake sequence and their time and/or frequency positions being different from those of the symbols of a synchronization sequence that said signal has,
- estimating the value and the position of the fake bits to be inserted within the data sequence to be transmitted at the input of the
30 transmission chain or of a subportion of the transmission chain so as to obtain, in the sequence produced at the output of the

transmission chain, the predefined value and time position of the symbols of said fake sequence,

- inserting, within said data sequence, the fake bits at the positions obtained.

5

At the expense of a slight loss of useful bit rate, the implementation of the method for decoying according to the invention when applied to a communication system provides it with resistance toward a selective interception and jamming system that will then selectively catch hold of and jam the fake synchronization sequences instead of selectively catching hold of and jamming the synchronization patterns that are actually used by a receiver that is compatible with said communication system.

According to one special aspect of the invention, when the transfer function F of the transmission chain is reversible, the value and the position of the fake bits are estimated by determining the inverse transfer function F^{-1} of said transfer function F and by applying said inverse transfer function F^{-1} to said fake sequence.

According to one special aspect of the invention, the inverse transfer function F^{-1} of said transmission chain is determined by composing, in reverse order, the inverse transfer functions of the various blocks making up said chain.

According to one special aspect of the invention, the transmission chain has at least one correction code with a yield of k/n for which the inversion of its transfer function is performed by resolving the system of equations according to

$$[i_0 \quad \dots \quad i_{k-1}] = [c_{p_1} \quad \dots \quad c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \quad \text{where } [c_{p_1} \dots c_{p_d}] \text{ are}$$

the symbols whose value is fixed in the encoded sequence, $[i_0, \dots, i_{k-1}]$ is the sequence to be produced at the input of said code, and m_{i,p_j} are the coefficients of the generator matrix for said code, $p_1, p_2, \dots, p_d$ being the set of positions for the symbols whose value is fixed in the encoded sequence, with
 5 d being an integer at most equal to the number n of symbols in the encoded sequence.

According to one special aspect of the invention, said correction code is a linear block code or convolutional code or a turbocode or an LDPC low-
 10 density code.

According to one special aspect of the invention, the transmission chain moreover has a jumbler and/or an interleaver and/or a framing module and/or a signal binary encoder and/or a modulator.

15 According to one special aspect of the invention, when the transfer function F of the transmission chain is nonsurjective, the value and the position of the fake bits are estimated by looking for the input sequence T' of the transmission chain that minimizes a distance criterion between the sequence $F(T')(t)$ obtained at the transmission chain output when said
 20 sequence T' is actually produced at its input and said fake sequence.

According to one special aspect of the invention, said distance criterion is taken to be equal to the integral, over a given duration, of the squared norm for the difference between the sequence $F(T')(t)$ obtained at
 25 the transmission chain output and said fake sequence.

According to one special aspect of the invention, the input sequence T' of the transmission chain that minimizes said distance criterion is looked
 30 for over a subset of the set of binary sequences that are possible at the input of the transmission chain.

According to one special aspect of the invention, said data sequence is produced by an application, for example an audio, image or video encoder.

5 According to one special aspect of the invention, when the transfer function F of the transmission chain is nonsurjective, the fake bits are inserted into the sequence produced at the input of a subportion of the transmission chain whose transfer function is surjective and whose output is common to the output of the transmission chain.

10

According to one special aspect of the invention, said transmission chain has an encryption device and said subportion of the transmission chain excludes this device.

15 According to one special aspect of the invention, said fake sequence is positioned in an invulnerable time and/or frequency area of the signal to be transmitted.

20 According to one special aspect of the invention, said invulnerable area is a portion of the signal that is protected by a correction code or is a stuffing area containing no useful information.

25 According to one special aspect of the invention, when the signal comprising said fake sequence is received by said receiver, the fake bits are removed from the data sequence.

30 The subject matter of the invention is likewise a device for transmitting a signal having a transmission chain for transforming a data sequence to be transmitted into a signal to be transmitted and a means that is suited to implementing the method according to the invention.

The subject matter of the invention is likewise a computer program having instructions for executing the method according to the invention when the program is executed by a processor.

5 The subject matter of the invention is likewise a processor-readable recording medium on which is recorded a program having instructions for executing the method according to the invention when the program is executed by a processor.

10 Other features and advantages of the present invention will become more apparent upon reading the description that follows with reference to the appended drawings, in which:

- figure 1 shows a diagram illustrating the problem of the jamming of synchronization sequences in a signal transmitted by a transmitter
15 to a receiver,
- figure 2 shows a diagram illustrating the solution proposed by the invention in order to solve the aforementioned problem,
- figures 3a and 3b show two diagrams illustrating the generation, according to the invention, of fake synchronization patterns,
- 20 - figure 4 shows a block diagram of the various functions successively implemented by a transmitter of a communication system in order to transmit a signal containing data to be transmitted,
- figure 5 shows a diagram of the shift registers for a convolutional
25 code with a yield of $\frac{1}{2}$,
- figure 6 shows a representation, for the convolution code example associated with figure 5, of the generator matrix for such a code,
- figure 7 shows an illustration of the necessary and sufficient condition for imposing, at the output of the convolutional code
30 defined in figures 5 and 6, the values of a sequence of consecutive bits,

- figure 8 shows an illustration of the parity relations for a punctured code.

In the remainder of the description, the expression “useful data”,
5 “useful bits”, “useful information” or “useful symbols” is used to denote the
binary data to be transmitted between the application executed by a
transmitter and the corresponding application executed by a receiver as
opposed to the binary data that are present in the transmitted frames but that
are not intended for the application executed by the receiver but are used for
10 the purposes of signaling, synchronization or any other function that is
necessary to the correct operation of the communication system.

Figure 1 illustrates, using a diagram, the problem of the jamming of
synchronization sequences in a signal transmitted by a transmitter and to a
15 receiver.

The top of the figure shows a wireless communication system in the
form of a transmitter EM that communicates with a receiver REC by means of
radio waves. The transformation of the binary data to be transmitted into a
20 radio signal S can be specified by a telecommunication standard or norm.
This specification notably defines the insertion, in a signal to be transmitted,
of synchronization sequences SYNC. Such sequences are made up of
symbols that are known to the pieces of equipment in the system and are
positioned periodically or in a time pattern that is likewise known both to the
25 transmitter EM and to the receiver REC, which implement the same
telecommunication standard. By correlating the known synchronization
sequence with the received signal, the receiver REC can synchronize its
timing to that of the transmitter, for example by detecting the start or end of a
frame, as indicated by the presence of said sequence, in the transmitted
30 signal.

In the field of passive listening, there are pieces of equipment called interceptors INT that are suited to intercepting the signal transmitted by a transmitter EM, to synchronizing themselves using the synchronization sequences SYNC and then to demodulating and/or analyzing, as the case
5 may be, the intercepted symbols in order to decode the information or reproduce the corresponding signal.

In the field of jamming, there are moreover jammers BR, which are coupled to the interceptors INT, that aim to neutralize the communication
10 system EM, REC by transmitting, at the same radio frequency, a compatible high-power signal that has a high level of correlation or the like with the signals that are expected by the targeted receiver, thus bringing about the loss of synchronization in the system. The interception and jamming systems BR, INT thus have both interception functions, which they can use to detect,
15 identify and regenerate synchronization sequences in a transmitted signal, and jamming functions that are based on transmissions of signals that have a high level of correlation or are identical with/to the signal expected by the target receiver. By concentrating the jamming power solely on a synchronization sequence SYNC, and, as the case may be, by reproducing
20 the sequence SYNC or a close sequence, as illustrated at the bottom of figure 1, the effectiveness of the jamming is reinforced because it concentrates its jamming power on the time portions of the transmitted signal that are most vulnerable, namely the synchronization sequences, and thus penetrates deeply into the reception chain of the receiver REC.

25

The example of the synchronization sequences is nonlimiting and can be extended to any sequence of known symbols that are fixed *a priori* by the telecommunication standard or the radio access technology and that, when jammed, bring about a malfunction of the system. In particular, such
30 sequences likewise include equalization sequences and frame synchronization patterns in the signal. These sequences, depending on the

case, precede or are multiplexed with the information that is to be transmitted. They are most often a combination that is limited in terms of position and value, which increases their vulnerability to interception and jamming.

5

Figure 2 illustrates, using a diagram similar to that in figure 1, the principle on which the invention is based that allows decoying of an interception system INT and/or a jammer BR in order to protect the synchronization within the communication system EM, REC.

10

According to the invention, one or more fake synchronization patterns SYNC_F are introduced into the signal transmitted by the transmitter EM. These fake sequences have steady-state characteristics that are easily detectable by an interceptor. They are inserted periodically or in a
15 predetermined time pattern into the signal that is to be transmitted. By way of example, they can be inserted with a period that is identical to that of the real synchronization patterns SYNC that are present in the signal but with a time shift and/or a frequency shift so that the fake synchronization sequences do not replace, in full or in part, the real synchronization sequences that need to
20 be protected in order to ensure correct operation of the communication system. The fake sequence(s) SYNC_F can advantageously be positioned on the least vulnerable or best protected portions of the signal, for example on portions that are protected by a robust correction code or on stuffing frequencies or time intervals that are empty of useful data and not used for
25 synchronization or access to the network, or else on frequencies or time intervals that are empty of useful data and dedicated to that purpose.

The values and positions of the symbols in the fake synchronization sequences SYNC_F are defined by values and/or positions that are different
30 than those used by the legitimate receiver for the needs specific to setting up radio communication with the transmitter, in order to avoid any confusion with

the sequences used by the legitimate receiver. The expression legitimate receiver in this case denotes a receiver REC that is compatible with the communication system and capable of communicating with a transmitter EM of said system.

5

The periodicity values and characteristics of the fake synchronization sequences SYNC_F are preferably chosen from among patterns close to those used in the field of telecommunications and easily identifiable by an interceptor. By way of example, they may be patterns close to those
 10 commonly intended for frequency synchronization that are made up of a series of identical symbols "0000..." or "1111...". They may likewise be patterns close to those commonly intended for time synchronization that are made up of series of alternating symbols "010101...". They may likewise be symbol sequences close to those used in some known civil standards using
 15 the same frequencies ranges as the transmitter. In all cases, the values and positions of the symbols in the fake synchronization sequences must nevertheless be different than those used by the communication system in order to ensure synchronization between a transmitter and a receiver.

20 The insertion of such fake synchronization sequences does not disturb the synchronization between the transmitting EM and receiving REC equipment because the fake sequences chosen are sufficiently decorrelated from the real synchronization sequences that are used in the communication system. The values and the positions, in the time/frequency domain, of the
 25 symbols in the fake sequences are chosen so that they are different than the values and the positions of the real synchronization sequences. Thus, the fake sequences can be processed by a receiver of the communication system either as random data that are comparable to transmitted data or as decoys whose values and positions are known *a priori* and that are therefore
 30 easily identifiable in order to be eliminated in the course of the synchronization process.

The aim of inserting fake synchronization sequences into the signal is to decoy an interception and jamming system INT, BR. The capability of the jammer to identify and analyze the signal is adversely affected by the
5 existence of these fake sequences, which the jammer tends to detect and to identify as being real synchronization sequences. The computation capability of the interception function of the jammer is requested over insignificant sequences of the signal, and the jamming signal will tend to be transmitted over time zones of the signal that correspond to the fake sequences while the
10 real synchronization sequences are protected. The techniques of oriented identification that the interception function would be able to implement are disturbed by the appearance of the fake signals, and further still if the latter are produced in great quantity and are variable from one communication to another or managed using long-term fluctuation mechanisms.

15

The levels of performance of an interception and jamming system are decreased all the more when a large number of fake synchronization patterns are present in the transmitted signal. This is because the analysis system of the jammer has to break up its computation efforts, to the point of possible
20 saturation of its computation capabilities, over artificial patterns other than the real synchronization pattern, since there is no way of recognizing the latter *a priori*. Even if the analysis system of the jammer uses a database of signals that are to be identified (on the basis of an approach of oriented analysis and identification), the implementation of the invention will tend to load said
25 database artificially, to break up the efforts of oriented analysis and search and to disadvantage identification, especially if the fake patterns artificially transmitted are intentionally made to fluctuate over time.

The invention thus allows decoying of an intelligent jammer but also
30 prevention of synchronization of the signal transmitted by any receiving or intercepting equipment that does not know the real synchronization pattern.

Inserting a fake synchronization pattern into a frame of symbols that are modulated at the output of the transmission chain leads to modification of the transmission chain itself and thus the transmitter itself, which is not
5 always technically possible, on the one hand, or desirable from the point of view of the complexity of implementation.

The aim of the invention is to allow the introduction of fake synchronization patterns by operating at the input of the transmission chain,
10 that is to say directly on the binary data that are to be transmitted, without modifying the transmission chain, and using software means.

Figures 3a and 3b illustrate the method for generating fake sequences according to the invention.

15

Figure 3a schematically illustrates the transformation undergone by a sequence of useful binary data D_U that are to be transmitted in order to obtain a sequence of modulated symbols S_T , which are ready to be transmitted in the form of a radio signal. The transformation performed
20 corresponds to the transfer function F of the transmission chain of the transmitter. The sequence of modulated symbols S_T is made up firstly of useful symbols S_U from the transformation of the useful binary data D_U and secondly of at least one synchronization sequence SYNC or of an equivalent sequence that is composed of symbols that are known to all equipment in the
25 communication system.

Figure 3b illustrates the implementation of the method according to the invention.

30 Firstly, the location and makeup of the fake synchronization sequence $SYNC_F$ are chosen in an empty frame T_F , of the same size as a real frame of

modulated symbols S_T and on a channel of a given frequency f . The temporal position and/or the frequency channel that are chosen are different than the temporal position and/or the frequency channel for a real synchronization sequence. The exact size and nature of the fake sequence may be variable.

5

Secondly, the value and the position of the fake bits B_F to be inserted into the data sequence that is to be transmitted, at the input of the transmission chain, are estimated so as to obtain, at the output of the transmission chain, the predefined value and temporal position of the symbols in said fake sequence $SYNC_F$. This operation can be carried out by computing the inverse transfer function F^{-1} for the transfer function F implemented by the transmission chain and then applying the inverse transfer function F^{-1} to the fake frame T_F in order to obtain a modulated frame D_F comprising the fake bits B_F .

15

The fake bits B_F are then inserted into the real data sequence to be transmitted D_U at the input of the transmission chain by puncturing, by shifting or by multiplexing the useful bits corresponding to the real data sequence. The sequence of modulated symbols that are obtained at the output of the transmission chain comprises the useful symbols S_U , the real synchronization pattern $SYNC$ and the fake synchronization pattern $SYNC_F$ all at once.

The fake bits can thus be introduced directly into the sequence of data to be transmitted, without either modifying or intruding in the transmission chain of the transmitting equipment. If the data to be transmitted are from an application, for example an audio or video source encoder, it is possible to intercept the application binary data before they enter the transmission chain that is implemented on the physical layer of a modem. This interception can take place on an intermediate layer, for example on the network layer.

30

On reception of the signal by the receiver REC, the fake bits are removed from the demodulated and decoded data sequence.

Figure 4 shows a block diagram of the various functions successively implemented by a transmitter of a communication system in order to transmit a signal containing data for transmission. The main functions traditionally implemented are shown, it being understood that the diagram in figure 4 is given by way of illustration and without limitation. In particular, some functions can be omitted and the order of some functions can be modified.

The transfer function F of the transmission chain is equal to the composition of the transfer functions of each independent functional block of the chain, it being understood that the blocks are connected in series. The inverse transfer function F^{-1} is, when it exists, equal to the composition, in the inverse order, of the inverse transfer functions of each block. In other words, if $f_1, f_2, \dots, f_N$ are the transfer functions of each functional block in the chain, then the overall transfer function is equal to $F = f_1 \circ f_2 \circ \dots \circ f_N$ and the inverse transfer function F^{-1} is equal to $F^{-1} = f_N^{-1} \circ f_{N-1}^{-1} \circ \dots \circ f_1^{-1}$.

To estimate the overall inverse transfer function F^{-1} , it is therefore advisable to determine the inverse transfer function of each unitary block.

The direct transfer function F of the transmission chain may be known when the invention is implemented by the designer of the communication system or when said system complies with a known standard. It can likewise be estimated by testing the transmitting equipment, for example by injecting test signals at its input and by analyzing the signals obtained at the output.

The transformations applied in the transmission chain on the bit stream are generally reversible, that is to say that it is possible to find the binary input stream from the bit stream at the output. This is the operation carried out by the receiver.

However, some functions implemented by the transmission chain of a communication system may not always be surjective. In other words, it may be that an encoded bit stream TBC, the values of which would want to be forced, at the output of a module of the transmission chain does not correspond to any series of useful bits TBU at the input of said module. By way of example, the channel encoding or framing operations transform a useful bit stream of length L_u into an encoded bit stream of length L_c . Owing to the framing operations, which are necessary in order to ensure synchronization of the receiver, and error correction coding operations, which are necessary in order to compensate for the effect of the propagation channel, in practice $L_c > L_u$ is still the case. This means that among the 2^{L_c} sequences of L_c encoded bits, only 2^{L_u} sequences can be obtained through encoding. Encoding is therefore never surjective.

15

In such a case, it is not possible to determine the inverse transfer function F^{-1} of the overall transmission chain, but only the inverse F^{-1} of the transmission chain on the restricted image $F(\{TBU\})$ from the set $\{TBU\}$ of the streams of useful bits at the input of the transmission chain.

20

Efforts are therefore made to determine the extent to which it is possible to force the value of some of the bits of the encoded signal. In particular, efforts are made to determine the number of bits whose value can be imposed and the extent to which it is possible to choose not only the value but also the position of these bits. In the case of a channel encoder, efforts are made to impose the value of a series of consecutive encoded bits so as to obtain encoded patterns that are similar to synchronization patterns.

25

The practical implementation involves successively analyzing the various transformations of the bit stream beginning with the transformation that operates last in the transmission chain. For each transformation, the

30

inputs that need to be applied in order to obtain the desired encoded bit stream at the output are determined.

With this in mind, the elementary transformations of the bit stream and
5 the reversibility thereof are analyzed on a case by case basis in the description that follows.

The transmission chain 400 shown in figure 4 has an application 401 that is capable of generating or transforming a binary data sequence for
10 transmission. The data for transmission may be textual, audio or video data or any other information. The application 401 may likewise have a source encoding function, for example, an audio, image or video encoder that is capable of removing or reducing the information redundancy or of reducing the noise affecting the sequence. At the output, the application 401
15 generates a useful binary sequence T for transmission. The invention is advantageously implemented at the output of the application 401 by modifying the useful binary sequence T in order to insert fake bits into it so as to obtain, at the transmission chain output, a sequence of modulated symbols $F(T)(t)$ for transmission comprising at least one fake synchronization pattern.

20

The transmission chain 400 may likewise have a correction coding module 402.

The goal of a correction coding function is to transform the binary
25 sequence of useful data that is received at the output of the application 401 into a protected binary sequence so that the impact of errors due to the transmission channel is as small as possible. In order to make the binary sequence of useful data more robust to the imperfections of the transmission channel, the correction coding function adds redundancy to this binary
30 sequence.

The determination of the inverse transfer function of a correction coding module is equivalent to searching for the binary sequence to be produced at the input of the correction encoder in order, at the output, to obtain an encoded sequence in which the value and the position of a
 5 predetermined number of bits are imposed.

There are various types of correction codes, including linear block codes, convolutional codes or else turbocodes and LDPC low-density codes.

10 In the text that follows, the determination of the inverse transfer function of a correction encoder is described for various types of correction codes, linear block codes, convolutional codes and also turbocodes and LDPC codes.

15 Linear block codes

A linear block correction code with a yield of k/n transforms a binary sequence comprising k symbols into a protected binary sequence comprising n symbols, where n is absolutely greater than k . Such a code therefore
 20 introduces $n-k$ redundancy symbols. The symbols may be bits or be made up of a plurality of concatenated bits. The block encoding operation is a one-to-one transformation of a word of the message $i = (i_0, \dots, i_{k-1})$ into a code word $c = (c_0, \dots, c_{n-1})$ defined by the following system of linear equations (where “+” denotes modulo 2 addition, “.” denotes modulo 2 multiplication) and g_{en} are
 25 coefficients with a value in the Galois field $GF(2)$, arranged in a matrix of size $n \times k$:

$$c_n = \sum_{e=0}^k g_{en} \cdot i_e, \text{ for } 0 \leq i \leq n-1$$

Among the 2^n existing binary sequences comprising n bits, only 2^k can be generated. The correction coding operation therefore limits the possibility of generating any desired binary sequence.

5 The block encoding involves obtaining the product of a k -bit input information vector with a binary matrix, of full rank, with a size $k \times n$, called a generator matrix, in order to obtain an encoded vector of n bits. Often, the code is referred to as left, or right, systematic, when the first k , or the last k , bits of the encoded vector of n bits correspond to the k bits of the input information vector. The encoding operation can be illustrated by the following relation, where $i_0, \dots, i_{k-1}$ are the bits of the useful sequence of the input, $c_0, \dots, c_{n-1}$ are the bits of the encoded sequence and $m_{i,j}$ are the coefficients of the generator matrix for the code.

$$[c_0 \quad \dots \quad c_{n-1}] = [i_0 \quad \dots \quad i_{k-1}] \cdot \begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & \dots & m_{0,n-1} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,0} & m_{k-1,1} & m_{k-1,2} & \dots & m_{k-1,n-1} \end{bmatrix}$$

15

Case of systematic block codes

When the code is left systematic, the encoded sequence is written $[i_0 \quad \dots \quad i_{k-1} \quad c_k \quad \dots \quad c_{n-1}]$. The inverse transform of the encoding operation involves analyzing the received word in order to determine whether it is a possible code word. If it is not a possible code word, it is necessary to replace it with the code word that is at a minimum distance from the received code word. Next, since the code is systematic, the information is obtained by removing the last $n-k$ bits of the word. In other words, it is possible to impose, through the choice of the input sequence, the output value of the first k bits of the encoded word. The values of the $n-k$ remaining bits are then deduced from the values chosen for the k bits that have been forced. The situation is exactly the same for a right systematic block code.

30

Case of block encoding from cyclic codes

The commonly used block codes are cyclic block codes or else are derived from cyclic block codes through puncturing or shortening.

- 5 In the case of a cyclic block code, if $[c_0 \dots c_{n-1}]$ is a code word, any circular permutation of the word $[c_i \ c_{i+1} \dots c_{n-1} \ c_0 \dots c_{i-1}]$ is also a code word.

By writing the code words $c(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1}$ in polynomial form, all the code words appear as multiples of one and the same polynomial $g(x) = g_0 + g_1 \cdot x + g_2 \cdot x^2 + \dots + g_{n-k} \cdot x^{n-k}$, of $n-k$ degree, called the generator

- 10 polynomial for the code.

An information word $[i_0 \dots i_{k-1}]$ can also be written in polynomial form $i(x) = i_0 + i_1 \cdot x + i_2 \cdot x^2 + \dots + i_{k-1} \cdot x^{k-1}$. It is always possible to write the encoding operation in right systematic form. This encoding operation involves computing the division of $i(x) \cdot x^{n-k}$ by $g(x)$. The remainder of the division is

- 15 $v(x)$ (of degree less than or equal to $n-k-1$) and the quotient of the division is $k(x)$. Therefore, $i(x) \cdot x^{n-k} = k(x) \cdot g(x) + v(x)$, and $c(x) = i(x) \cdot x^{n-k} + v(x) = k(x) \cdot g(x)$.

Since $v(x)$ is of a degree less than or equal to $n-k-1$, the values of the coefficients of $c(x)$ for degrees greater than or equal to $n-k$ are the coefficients of $i(x)$ shifted by $n-k$.

20

It is therefore always possible to write cyclic block codes in right systematic form, and therefore to impose the value of the last k bits that are equal to the information bits. Since the code is cyclic, any circular permutation of a code word is also a code word, which means that it is also

25 possible, still for the cyclic codes, to impose the value of any group of k consecutive bits of a code word.

Moreover, the dependence between the values of the bits at the input of the encoder and the bits of the output of the encoder is linear. The values

30 of the bits that need to be forced at the input of the encoder are linearly

dependent on the values of the other bits at the input of the encoder and the values of the forced bits at the output of the encoder.

General case of block codes

5

In the more general case where the code is neither systematic nor comes from cyclic codes, a sufficient condition in order to be able to impose the value of a group of d bits at the output of the encoder, where d is less than or equal to k , is that the set of positions $p_1, p_2, \dots, p_d$ of the bits in the encoded sequence must be such that the submatrix of the generator matrix for the code:

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}$$

is of full rank, that is to say of rank equal to d .

15

The reason is that when this condition is fulfilled, it is possible to determine the sequence $[i_0 \dots i_{k-1}]$ at the input of the encoder that allows the values of d bits or symbols to be fixed in the encoded sequence by resolving the following system of equations:

$$[i_0 \dots i_{k-1}] = [c_{p_1} \dots c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \text{ where } [c_{p_1} \dots c_{p_d}] \text{ are}$$

20

the bits or symbols whose value is fixed in the encoded sequence, the indices $p_1, p_2, \dots, p_d$ denoting the positions of the bits or symbols in the sequence of n bits or symbols.

Thus, for all the usual block codes, which are systematic, or cyclic, or constructed from cyclic codes, that is to say for most of the usual codes, it is possible to impose, through the choice of inputs of the encoder, any group of k consecutive bits among the n bits of the encoded vector. Moreover, in the

25

case of two successive code words, by imposing the last k bits of the 1st code word and the first k bits of the 2nd code word, it is possible to impose the value of a group of $2k$ successive bits on a binary sequence comprising at least two code words.

5

In the more general case, the submatrix of the generator matrix for the block code corresponding to the positions of the bits or symbols to be fixed is of full rank. However, in some cases, some submatrices of the generator matrix may be not of full rank. Such a case is illustrated in a nonlimiting
10 example of a Hamming code (7,4) whose generator matrix $M(7,4)$ is given by

$$M(7,4) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The generator polynomial for this code is $g(x) = 1 + x + x^3$. For this code, it is possible to impose the last four bits (m_0 to m_3) of the code word because the
15 encoding is right systematic. In order to obtain the following code word:

$[c_0 \ c_1 \ c_2 \ m_0 \ m_1 \ m_2 \ m_3]$, it suffices to encode the information vector $i(x) = [m_0 \ m_1 \ m_2 \ m_3]$.

The encoding operation is represented by the following relation:

$c_{sys, right}(x) = i(x) \cdot x^{n-k} + v(x) = k_{sys, right}(x) \cdot g(x)$, where $k_{sys, right}(x)$ is the vector to be
20 encoded and $c_{sys, right}(x)$ is the code word obtained.

It is likewise possible to impose, by way of example, the first four bits of the code word on the values of the information vector $[m_0 \ m_1 \ m_2 \ m_3]$.

This necessitates finding the information vector that, once encoded, gives the
25 following code word:

$$k_{sys, left}(x) \cdot g(x) = m_0 + m_1 \cdot x + m_2 \cdot x^2 + m_3 \cdot x^3 + c'_4 \cdot x^4 + c'_5 \cdot x^5 + c'_6 \cdot x^6.$$

In order to compute this information vector $k_{sys, left}(x)$, the property that the code is invariant through circular permutation is used. There is a change from

right systematic encoding to left systematic encoding through four circular permutations to the right. Therefore, the code word $[c_0 \ c_1 \ c_2 \ m_0 \ m_1 \ m_2 \ m_3]$ becomes the code word $[m_0 \ m_1 \ m_2 \ m_3 \ c_0 \ c_1 \ c_2]$ by performing these four circular permutations.

- 5 The code word $[m_0 \ m_1 \ m_2 \ m_3 \ c_0 \ c_1 \ c_2]$ is obtained by encoding the information vector $i'(x) = [m_3 \ c_0 \ c_1 \ c_2]$ (because the code is right systematic).

By contrast, it can be noted that if the second, fourth, fifth and sixth
10 columns of the generator matrix for the code $M(7,4)$ are considered, the following submatrix is obtained, which is not of full rank, the coefficients in its last line being equal to 0:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Indeed, the rank of a matrix corresponds to the number of independent
15 columns of the matrix or equivalently to the number of independent rows of the matrix.

It is therefore not possible to force the values of these four bits (the second, fourth, fifth and sixth) of the encoded word: if the value of three of these bits is forced, the value of the fourth bit is deduced from the values imposed on
20 the three forced bits.

It is therefore not possible, for any code (n, k) with a yield of k/n , to force the value of any group of k bits. By contrast, for the most frequently used codes, it is possible to force the value of any group of k consecutive bits
25 when the submatrix of the generator matrix for the code, which matrix is associated with the positions of the bits to be fixed, is of full rank.

The dependencies between the values of the bits at the input of the encoder and the values of the bits at the output of the encoder are specified more precisely below.

- If it is desirable to impose the value of a limited number l , less than k , of bits at the output of the encoder, it suffices to impose the value of l bits at the input of the encoder.
- By contrast, the value of these bits is dependent not only on the value of the pattern generated at the output of the encoder but also on the values of the other bits at the input of the encoder (which are *a priori* random because they are linked to the useful information).

Returning to the example of the encoding matrix $M(7,4) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$,

if it is desirable to force the value of the first two encoded bits, c_0 and c_1 , it is possible to choose an information vector $i(x) = [f_0 \ f_1 \ i_0 \ i_1]$, where i_0 and i_1 are information bits that are left free and f_0 and f_1 are bits forced in order to obtain the desired pattern. The values that it is necessary to choose for f_0 and f_1 in order to obtain the desired values of c_0 and c_1 are given by the relations:

$$f_0 = c_0 + i_1 + i_2$$

$$f_1 = c_0 + c_1 + i_2$$

This illustrates that the values of the forced bits at the input of the encoder (f_0 and f_1) are linearly dependent on the values of the bits of the pattern at the output of the encoder (c_0 and c_1) and the values of the other bits at the input of the encoder.

25 Convolutional codes

Convolutional codes make up the second large family of error correction codes. Whereas linear block codes allow the message to be chopped into

blocks of k symbols, convolutional codes apply a sliding window of $k^*(m+1)$ symbols to the message and produce a continuous sequence of encoded symbols.

In general, the symbols are binary (i.e. with a value of 0 or 1 in the Galois field GF(2), “+” denotes modulo 2 addition and “.” denotes modulo 2 multiplication). Most often, convolutional codes have the parameter $k=1$ and the yield of the code is therefore in the form $1/n$. If a_j is an information symbol, the parity symbols b_j associated with the symbol a_j are defined by the following convolutional relation, where $g_{e,j,i}$ are the coefficients of n^*k polynomials of degree m with coefficients and values in the Galois field GF(2) (the code is wholly defined by the set of coefficients $g_{e,j,i}$, $e = 0, \dots, m$, $j=0 \dots k-1$, $i=0 \dots n-1$) :

$$b_{p \cdot n + i} = \sum_{e=0}^m \sum_{j=0}^{k-1} g_{e,j,i} \cdot a_{p \cdot k + j - e \cdot k}, \text{ for } i = 0 \dots n-1, \forall p.$$

The n symbols at the output of the encoder are linearly dependent on the last $k^*(m+1)$ symbols at the input of the encoder.

From a code with a yield of $1/n$, codes referred to as “derived”, corresponding to $k > 1$, can be constructed by means of puncturing (most often $k=n-1$ after puncturing).

More generally, when the yield of the code is equal to k/n , the convolutional encoding is periodic encoding with a period of k bits on the binary signal at the input. For each new group of k bits, n encoded bits are computed. The n encoded bits are bit combinations relating to the last $(m+1)$ groups of k bits. m is the constraint length of the code.

A nonlimiting example will now be used to illustrate the step of the method according to the invention that involves reversing the transfer function of a convolutional code, in other words determining the sequence of

bits to be produced at the input in order to obtain at the output an encoded sequence in which the value and the position of a predetermined number of bits are fixed.

5 Nonlimiting example of a convolutional code with a yield of $\frac{1}{2}$

A common binary convolutional code is considered, the registers of which are shown in figure 5, with a yield of $\frac{1}{2}$ and a length of $m+1=7$ defined by two polynomials of degree 6 that are defined in octal notation by (171,
10 133). These two polynomials are written $G_1(X)=1 + X + X^2 + X^3 + X^6$ and $G_2(X)= 1 + X^2 + X^3 + X^5 + X^6$ and correspond to the recurrence relations $b_{2n}=a_n + a_{n-1} + a_{n-2} + a_{n-3} + a_{n-6}$ and $b_{2n+1} = a_n + a_{n-2} + a_{n-3} + a_{n-5} + a_{n-6}$ in the Galois field GF2 ("+" denotes modulo 2 addition). The polynomials G_1 and G_2 are applied to the input bits in order to respectively form the output bits with
15 an even subscript and the output bits with an odd subscript interleaved and then two by two in the form $b_{2n}b_{2n+1}$ in order to form a binary stream of size equal to a multiple of 2. The text below illustrates the possibility of choosing the bit stream at the input so as to generate desired patterns after encoding.

20 For this code, in each period, for a bit produced at the input of the encoder, two bits are generated at the output. Depending on the state of the registers of the encoder, the two bits (b_{2n} , b_{2n+1}) at the output need to be chosen from either (0,0) or (1,1), or (0,1) or (1,0). The reason is that the last bit that enters the encoder is used to compute each of the two outputs of the
25 encoder: by changing this bit, the values of the two outputs are changed. This means that, for a given state of the register, the two possible bits at the output need to be chosen from two complementary groups. It is therefore always possible to choose the bit at the input so as to force the value of one of the two output bits. It is therefore easily possible to use this code to force
30 one bit out of two at the output of the encoder.

In practice, for the codes with the highest level of performance that are used in a modem, the groups of output bits, for one state of the encoder, are chosen so as to be at maximum distance from one another. For these codes, the last bit that enters the encoder is used to compute each of the two
 5 outputs of the encoder. It is therefore possible, for all common codes with a yield of $1/2$, to force the value of one bit out of two at the output of the encoder.

It will now be shown that it is likewise possible to choose a series of
 10 bits at the input of the encoder so as to obtain at the output of the encoder an encoded sequence comprising a series of consecutive bits of fixed value. We will consider once again the previous example of the encoder $r=k/n=1/2$, (171, 133) with a length of $m+1=7$. For this encoder, the impulse response, that is to say the response of the encoder to a binary input sequence
 15 comprising a bit of value 1 preceded and followed by bits all having the value 0, is given by the sequence 11101111000111, with a length of $14=2m+2$.

The encoding operation can be written in the matrix form, shown in figure 6, where the rows of the generator matrix for the code correspond to
 20 the impulse response of the encoder, shifted by two bits from one row to the next (because $n=2$), or more generally shifted by n bits from one row to the next when the code has a yield of $1/n$.

Thus, an identical formalism to that used for the linear block codes is
 25 obtained, that is to say that the encoded sequence is obtained by obtaining the matrix product of the information sequence with the generator matrix defined above.

The same rule established above concerning linear block codes can
 30 thus be applied to convolutional codes, that is to say that it is possible to impose the value and the position of a set of bits at the output of the encoder

if and only if the submatrix M corresponding to the columns of the output bits is of full rank, which is illustrated in figure 7.

It can therefore be seen that for this convolutional code it is possible to force the value of 14 consecutive bits at the output of the encoder, on the basis of the inputs. It is also possible to interpret this result from the point of view of the parity relations between encoded bits. These – deterministic – relations characterize the dependencies verified by the groups of encoded bits on a one-to-one basis. In other words, the values of the encoded bits corresponding to a parity relation are interdependent. The values of the encoded bits not corresponding to a parity relation can be fixed independently of one another. The code considered above has parity relations of a length 14 that are spaced apart by two bits. It is therefore possible to choose 14 consecutive encoded bits that do not correspond to any whole parity relation, which means that their values can be fixed independently.

More generally, it is possible to impose the value and the position of a set of bits at the output of the encoder if and only if the submatrix

$$20 \quad \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix} \text{ is of full rank, where } m_{i,j} \text{ is the coefficients}$$

of the generator matrix and $p_1, p_2, \dots, p_d$ is the set of positions of the bits whose value is fixed in the encoded sequence.

Finally, in the case of a code with a yield of $\frac{1}{2}$, the maximum number of successive bits for which it is possible to fix the value is equal to the length of the impulse response of the code, that is to say $2m+2$, where m is the constraint length of the code.

Nonlimiting example of a punctured convolutional code constructed from a convolutional code with a yield of $\frac{1}{2}$

For punctured codes, which have lower redundancy, it is possible to
 5 force a larger number of consecutive bits at the output of the encoder. By
 way of example, the code defined by the preceding polynomials (171, 133)
 can be punctured in order to obtain a code with a yield of $\frac{3}{4}$. The parity
 relations of this code have a length of 26 (11111101011011001010011111)
 and are spaced apart by four bits. This example is illustrated in figure 8,
 10 which shows the subscripts 800 of the bits at the output of the encoder, a
 portion P_{28} of 28 consecutive bits having no whole parity relation and three
 parity relations R1, R2, R2 linked to said code, with a length equal to 26 bits.
 The parity relations are checked for all sequences of 26 bits starting at a
 subscript shifted by four bits for each new sequence.

15

By direct analogy with the above, it is therefore possible to choose the
 values taken by a group of $4(6+1)=28$ bits.

General case of common convolutional codes

20

For the more general case of a convolutional code with a yield of $(n-1)/n$,
 and with a constraint length of m , the parity relations are generally of
 length $n \cdot m + 2$ (in the case of the previous example, the situation was that
 $m=6$ and $n=4$ after puncturing, and the parity relations had a length of 26). In
 25 this case, and by analogy with the above, it is possible to choose the bits at
 the input of the encoder so as to force the value of $n \cdot (m+1)$ consecutive bits
 at the output of the encoder.

It is therefore possible to see that, with the convolutional codes, it is
 30 possible, as for the block codes, to force the value of some encoded bits,
 which are possibly consecutive and sometimes in large number. The

condition for this operation to be possible is that the matrix formed by the columns of the encoding matrix corresponding to the forced bits be of full rank.

It is particularly possible to force the value of consecutive bit
 5 sequences of significant length, particularly when the yield of the code is close to one and when the memory m of the code is large.

Just as for block codes, the dependence between the values of the bits at the input of the encoder and the bits at the output of the encoder is
 10 linear. The values of the bits that need to be forced at the input of the encoder are linearly dependent on the values of the other bits at the input of the encoder and the values of the forced bits at the output of the encoder.

Product turbocodes

15 The invention can likewise be applied to correction codes of product turbocodes type.

Turbocodes are correction codes that combine at least two single
 20 codes by interleaving the inputs so that each of the single codes sees a series of different information, on the one hand, and the information specific to each bit, block or message is distributed over these neighbors, on the other hand. As a result, even if some of the bits, blocks or messages are altered during the transmission, the corresponding information still exists
 25 more or less on neighboring bits, blocks or messages. The decoding procedure is iterative and collaborative between each single code. It introduces a notion of trust to each decoded bit, block or message and postpones the final decision about the values thereof ("soft decision"). Each of the decoders sends the others the information from its own decoding
 30 (called extrinsic information), which is multiplexed with the information at the input of the other encoders. The bit, block or message transmitted in this way is decoded a second time by the other single encoders, and the

corresponding information retransmitted to the other encoders (hence the name “turbo”, which is linked to the decoding procedure rather than to the actual code).

5 There are many single codes that can be employed.

It is possible to use convolutional codes. Recursive and systematic convolutional codes are particularly suited in practice. The codes can be positioned in series or in parallel. Clever management of the interleaving and
10 the iterative detection/correction of the data by each single code allows an increase in the detection and correction power of the overall method while limiting the number of iterations and the complexity.

Another turbo coding structure corresponds to product codes. In the
15 simpler version corresponding to two-dimensional product codes, the product code for the yield of $\frac{k_1 \cdot k_2}{n_1 \cdot n_2}$ is constructed from two elementary codes C_1 and C_2 with a yield of $\frac{k_1}{n_1}$ and $\frac{k_2}{n_2}$. The elementary codes used are very simple block codes (typically parity codes, Hamming codes or else extended Hamming codes). The $n_1 \cdot n_2$ successive bits appear as a succession of n_2
20 code words C_1 . When considering the bit stream decimated by a factor of n_1 , code words C_2 are obtained.

Product turbocodes, constructed from a plurality of block codes, are similar to block codes when it is necessary to determine if it is possible to
25 generate the desired pattern. The reason is that the encoding operation can be broken down into encoding and interleaving operations.

The breakdown of a product code into a plurality of encoding and interleaving operations will now be illustrated on the basis of a simple and

nonlimiting example. We will consider a product code constructed from two codes C_1 and C_2 with a yield of $\frac{k_1}{n_1}$ and $\frac{k_2}{n_2}$.

The encoding operation for a block of $k_1 \cdot k_2$ bits can be broken down as follows:

- Encoding with the code C_1 of the bit stream: the encoding transforms k_2 groups of k_1 bits into k_2 groups of n_1 bits.
- Single row/column interleaving: the bits are written on a row-by-row basis to a matrix with a size of k_2 rows and n_1 columns.
The bits are then read on a column by column basis.
- Encoding with the code C_2 of the bit stream: the encoding transforms n_1 groups of k_2 bits into n_1 groups of n_2 bits.
- Single row/column interleaving: the bits are written on a row by row basis to a matrix with a size of n_1 rows and n_2 columns.
The bits are then read on a column by column basis.

In practice, if a limited portion of consecutive bits are a block of $n_1 \cdot n_2$ encoded bits (a portion of length $n \cdot k'$ substantially smaller than $n_1 \cdot k_2$) is considered, the constraints of the code C_2 do not affect this portion because there is no parity relation linked to the code C_2 that is contained wholly in this portion of bits. Consequently, if a portion of consecutive bits of smaller size than $n_1 \cdot k_2$ is considered, everything occurs as if there were just the code C_1 when it is necessary to determine the patterns that it is possible to force. Like block codes, it is therefore possible, notably, to generate consecutive bit patterns of size $2 \cdot k_1$.

LDPC (Low Density Parity Check) codes

LDPC codes are special block codes that are constructed so that the parity bits are computed by introducing a parity relation of low significance. In practice, they are systematic (but not cyclic) block codes and the analysis performed on the block codes can be applied by analogy to the description
 5 above.

LDPC codes are generally very long. As established for block codes, LDPC codes therefore allow, notably, generation of patterns chosen with a series of consecutive bits of great length.

10

In summary, for any correction code for which the encoding operation can be performed by multiplying the information sequence by a generator matrix in order to obtain the encoded sequence, it is possible to fix the value and the position of a set of bits in the encoded sequence by imposing a
 15 special binary input sequence. However, this possibility exists only if the submatrix, of the generator matrix, defined by

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}$$

is of full rank, where $m_{i,j}$ is coefficients of

the generator matrix, and $p_1, p_2, \dots, p_d$ is a set of positions of the bits whose value is fixed in the encoded sequence. A full-rank matrix is a matrix in which
 20 all the columns are independent. If this is the case, it suffices to resolve the

$$\text{system of equations } [i_0 \dots i_{k-1}] = [c_{p_1} \dots c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1}$$

where $[c_{p_1} \dots c_{p_d}]$ are the bits or symbols whose value is fixed in the encoded sequence and $i_1, \dots, i_k$ are the unknowns in the system to be provided at the input of the encoder.

25

The transmission chain 400 may likewise have a scrambling module 403, likewise called jumbling.

Jumbling, or scrambling, is used to render the binary sequence for transmission as random as possible, with the aim of improving symbol synchronization but also of contributing to protection of the content of the jumbled messages. Its goal is to remove long successions of bits equal to zero or to one that prevent correct recovery of the symbol rate. There are several types of jumlbers, including synchronous jumlbers requiring a prior time reference or self-synchronizing jumlbers.

This transformation is reversible, the scrambling operation involves transforming one group of L bits into another group of L bits and the operation of inversion of the transformation of the bit stream is an operation performed by a receiver. Thus, the inverse transfer function of a jumbler or scrambler can easily be deduced from its direct transfer function.

In the case of a synchronous scrambler, the dependence between the value i of a bit at the input of the scrambler and the value c of the bit at the output of the scrambler is affine. Depending on the position of the bit under consideration, the dependence is either $c=i$ or $c=i+1$.

Case of a synchronous scrambler

In the case of a synchronous scrambler, a pseudorandom sequence is added modulo 2 to the binary signal that is to be scrambled. The binary series $\{\dots, e_k, e_{k+1}, \dots\}$ representing the scrambling sequence is periodic, with a long period L . The sequences most commonly used for scrambling are sequences of maximum length, constructed on the basis of a shift register. These sequences are generated by a shift register looped back according to a primitive P th-degree polynomial $E(X) = 1 + c_1X + \dots + c_PX^P$. A sequence with a period of $L = 2^P - 1$ is obtained. The sequences used for scrambling may be truncated, however. The k -th output e_k of the shift register checks the

recurrence relation $e_k = c_1 \cdot e_{k-1} + \dots + c_P \cdot e_{k-P}$, where $c_1, \dots, c_P$ are constant coefficients in the Galois field GF2 that are independent of the input data and the scrambled data, "+" still denoting modulo 2 addition in GF2. For an input binary series denoted $\{\dots, b_k, b_{k+1}, \dots\}$, the transformation F performed by a synchronous scrambler can be written $F(\{\dots, b_k, b_{k+1}, \dots\}) = \{\dots, b'_k, b'_{k+1}, \dots\}$, where $b'_k = b_k + e_k$. The transformation F is perfectly reversible. This is because the transformation F is equal to its inverse $F^{-1}=F$, since $b_k = b'_k + e_k$ and in the knowledge that $e_k + e_k = 0$ in the Galois field GF2. The operation F involves a new modulo 2 addition of the value of the scrambled bits by the outputs of the same shift register as on transmission in order to produce a descrambled sequence of identical length. In order to be able to force the bit stream at the output of a synchronous scrambler, it suffices to be correctly synchronized with the scrambling operation. In other words, the positions of the subscripts k or the start of a period must coincide before applying the transformation $F^{-1}=F$ to the scrambled bits.

Case of a self-synchronizing scrambler

In the case of a self-synchronizing scrambler, the states of the register are filled with a finite number of scrambled data. The output of the register is modulo 2 added to the data bit at the input in order to form the new scrambled bit.

The transformation F, which is performed by a self-synchronizing scrambler, is defined by a shift register with a Pth-degree primitive polynomial $E(X) = 1 + c_1 X + \dots + c_P X^P$ with a period of $L = 2^P - 1$ and is written, for a binary input series denoted $\{\dots, b_k, b_{k+1}, \dots\}$ in the form: $F(\{\dots, b_k, b_{k+1}, \dots\}) = \{\dots, b'_k, b'_{k+1}, \dots\}$, where $b'_k = b_k + c_1 \cdot b'_{k-1} + \dots + c_P \cdot b'_{k-P}$ ("+" still denoting modulo 2 addition in GF2). This allows the reception descrambling to be performed simply and without the need to synchronize itself beforehand. On reception, the scrambled bits $b'_{k-1}, \dots, b'_{k-P}$ are injected into the same register

as on transmission in order to reconstruct the output $b_k = b'_k + c_1 \cdot b'_{k-1} + \dots + c_P \cdot b'_{k-P}$.

The difficulty for forcing the bits at the output arises in this case from
 5 the fact that the expression of the states of the register as a function of the
 input data b_k (nonscrambled data) introduces an unlimited number of said
 input data. In other words, all the inputs b_k from startup of the scrambler
 intervene in the value of the states of the register.

10 In order to be able to force the value of the scrambled data, it is
 essential to have synchronous access to the scrambled data in order to be
 able to adapt the input data dynamically. This condition is necessary to the
 application of the invention if the transmission chain has a self-synchronizing
 scrambler.

15 The transmission chain 400 can likewise have an interleaving module
 404.

Interleaving is widely used on transmission channels for which the
 20 occurrences of errors are grouped by packets. Its function is to distribute said
 errors as uniformly as possible. On reception, the errors are, following
 deinterleaving, positioned such that they affect different code words. These
 errors can then be considered as decorrelated, and the corrective power of
 the decoders allows the impact thereof to be minimized. Interleaving also
 25 appears to be a means of introducing temporal diversity into the transmission
 chain and therefore contributes to protecting the latter from fading,
 interference and possible jamming. Many interleavers are made up of a table,
 and the input bits are then arranged by rows in the table, and the output bits
 are produced by reading columns of the table.

30

The transformation performed by an interleaver is likewise a reversible operation. Its inverse transfer function can be deduced from its direct transfer function. This is because it is a transformation that transforms one block of L bits into another block of L bits, by simply permutating the order of the bits. In order to obtain a desired block of bits following interleaving $B' = b'_k, \dots, b'_{k+L-1}$, it suffices to be synchronized to the block and to apply the inverse permutation to this bit stream B' which corresponds to the deinterleaving operation conventionally performed by a receiver. A block of bits $B = b_k, \dots, b_{k+L-1}$ is then obtained, which, once interleaved, is absolutely equal to the bit stream B .

10

The transmission chain 400 can likewise have a framing module 405. The framing 405 allows the receiver to synchronize itself to transformations of the bit stream such as interleaving or decoding operations, and then to recover data structured in the form of multiplexing of a plurality of streams or data in the form of words or bytes. For that, once the data structured in the form of frames corresponding to periodic scheduling patterns, the synchronization of the receiver to the frames is performed using synchronous periodic patterns of the frames. Each frame is thus preceded and/or followed, and/or contains a specific synchronization word that is used to synchronize the receiver to the received frames. A plurality of frames can likewise be grouped in order to form a multiframe or hyperframe. The synchronization patterns used for framing are repeated in the transmitted sequence and can be detected and jammed as explained above.

15

20

Framing limits the possibility of forcing the desired modulated signal, since some bits take values imposed at regular intervals, and it is necessary to keep these bits for correct operation of the receiver in contact with the transmitter. This transformation is therefore not surjective for blocks of data whose output size exceeds that of a block of useful data per frame.

25

30

However, framing takes up only a very limited share (and one that is quite delimited in terms of timing) of the total bit rate (example: frame start pattern using only a few symbols etc.) and does not prevent generation of the desired encoded signal within a frame.

5

This point is therefore not a particular pitfall for implementation of the invention.

The transmission chain 400 can likewise have a signal binary
10 encoding module 406.

Signal binary encoding is used to adapt the signal to suit the transmission channel. It transforms the digital message into an electrical signal in baseband or into a low-frequency signal. Two large classes of signal
15 binary codes may be cited, NRZ (non return to zero) transcoding codes and alphabetical codes.

The transformation performed by a signal binary encoder is a reversible operation. Its inverse transfer function can therefore be deduced
20 from its direct transfer function.

Finally, the transmission chain 400 has a modulator 407 that transforms notably the binary sequence into a sequence of modulated symbols. The symbols are taken from a complex set called a constellation. A
25 symbol can group a plurality of bits. By way of example, digital phase modulations or PSK (Phase Shift Keying) or digital amplitude modulations or QAM (Quadrature Amplitude Modulation) may be cited.

The transformation performed by a modulator is a reversible
30 operation. Its inverse transfer function can therefore be deduced from its direct transfer function.

According to the above, the transformations applied in the transmission chain to the bit stream are reversible, that is to say that it is possible to find the binary input stream from the binary output stream. In practice, redundancy is even added so as to be able to find the original binary signal in the presence of errors on the encoded bit stream.

However, still according to the above, some functions implemented by the transmission chain of a communication system may be not always surjective or reversible. It may be that an encoded bit stream TBC that it is desirable to force at the module output does not correspond to any "useful" series of bits TBU at the module input. This is the case notably for a block or convolutional correction coding function that does not meet the criteria set out above or of any other non-bijective encoding operation. In such a case, it is not possible to determine the inverse transfer function F^{-1} of the overall transmission chain.

In order to get around this difficulty, two variant embodiments of the invention are envisaged.

20

A first variant involves looking, among the set $F(\{TBU\})$ of modulated sequences that it is possible to obtain at the transmission chain output on the basis of the set $\{TBU\}$ of possible binary sequences at the input of the transmission chain, for the binary sequence T' that minimizes the distance between the modulated output transformation $F(T')$ of the sequence T' and the sought modulated sequence D that contains at least one fake synchronization pattern $SYNC_F$ positioned at the desired location. The distance under consideration can, by way of example, be a distance within the meaning of the least squares that is computed by integrating the difference between a possible sequence $F(T)$, T being part of the set $\{TBU\}$,

30

and the sought sequence D over a fixed time interval. A possible criterion can be computed using the following relation:

$$\forall T \in \{TBU\}; C(T, D) = \|F(T) - D\|_{L^2} = \int |F(T)(t) - D(t)|^2 dt$$

The input sequence $T' \in \{TBU\}$ of the transmission chain that minimizes the
 5 criterion C : $T' = \underset{T \in \{TBU\}}{\operatorname{Argmin}} [C(T, D)]$ is then sought.

However, the practical implementation of this variant must take account of two main constraints.

10 Firstly, for reasons of complexity, it is not possible in practice to obtain the image $F\{TBU\}$ of the modulated signals at the transmission chain output corresponding to the set of all possible useful bit streams (in other words, corresponding to the whole of the set $\{TBU\}$), but only a restricted subset of this image.

15

Secondly, the implementation of the invention can be simplified by taking the temporal supports principally linked to the operations of interleaving, encoding and framing as a basis. The reason is that it is not useful in practice, for implementation of the invention, to consider useful bit
 20 streams T in the set $\{TBU\}$ whose modulated output signals $F(T)$ (following encoding and interleaving, notably) would be spread over excessively long time intervals, or over an excessively large number of output frames, or whose distribution of the positions of the symbols would be too incomplete.

25 In practice, therefore, this variant of the invention can be implemented by restricting the set $\{TBU\}$ to a subset $\{TBU'\}$ that comprises only useful bit sequences T whose lengths and positions after transiting through the transmission chain (and notably after transiting through the encoding and interleaving modules) correspond to fake patterns SYNC'_F , which may be

less than optimum vis-à-vis the decoying of an interception and jamming system, but which are compatible with the implementation standard for the telecommunication system in terms of positions, recurrence and frame periodicity, and therefore easy to generate and to insert into the frames of
 5 useful data.

The time/frequency structure and/or the choice of positions, lengths and recurrences of the fake signals SYNC'_F similar, within the meaning of the aforementioned criterion, to the fake signals SYNC_F that are desired at the
 10 output are fixed by taking the frame periodicities of the transmission as a basis and taking away the fake patterns of the data that are really useful for synchronization and demodulation of the receiver as far as possible (in the time/frequency domain). By way of example, in a transmission of TDMA (Time Division Multiple Access) type using signals transmitted in the form of
 15 packets or bursts with or without frequency hopping, the fake patterns will be placed at positions that are different than those of the real patterns, at the end of a packet rather than at the beginning of a packet, on fictitious beacon frequencies rather than on those that are really used. In a transmission multiplexing a plurality of user streams using orthogonal codes, the fake
 20 patterns used will be codes that are orthogonal to the set of codes actually used by the users.

The restricted set $\{\text{TBU}'\}$ is then predetermined by reversing, analytically or through simulation, the modules of the transmission chain, and
 25 notably the interleaving modules and the encoding modules, for the patterns SYNC'_F .

This notably allows simplification of the implementation of the invention by concentrating the search for the binary sequences T in the
 30 inverse image (notably using the encoding and interleaving transformations) of a restricted set of fake output patterns SYNC'_F , correctly similar to the

desired fake sequence SYNC_F within the meaning of the aforementioned criterion, and with the utmost correspondence to the duration of a frame at the transmission chain output (and/or to the duration of a frame of useful information at the input of said transmission chain). The binary sequences T
 5 obtained in this manner are periodized per frame of useful information in order to generate fake patterns at the output, which are themselves periodic and have a period that is indexed to that of the frame of the output signal. It is then possible for the positions of the bits of the binary sequence T' for a fake output pattern to be computed once and for all, and it is also possible for the
 10 dependence of the bits of the binary sequence T' on the neighboring bits of information to be computed once and for all.

In practice, in order to generate said binary sequences T, it is therefore necessary to force the value of some useful bits (situated at the
 15 input of the channel encoding) at precise positions. The transformation applied involves first of all determining a partitioning for the frame of useful data by reversing the encoding interleaving modules: this determines quite precise positions at which, instead of transmitting useful data, the useful information bits are replaced with forced useful bits that are used to generate
 20 the fake sequences after interleaving and encoding. By contrast, these positions having been determined, the values of these fake useful bits can change depending on the value of the neighboring useful information bits (which are variable and random): the value of each of these forced useful bits is dependent not only the values of the desired encoded fake bits but also on
 25 the values of some neighboring useful bits (according to the scrambling and encoding scheme). These dependencies are linear or affine, which means that the value of each forced useful bit is expressed as the sum of fake bits at the output of encoding and neighboring useful information bits (i.e. linear dependence on the useful information bits plus possibly inversion of the
 30 encoded forced bits).

Finally, in this simplified variant of implementation of the invention,

- the positions taken by these forced useful bits are determined by starting from the position and the values of the fake sequences SYNC'_F and reassembling the transformations of the bit stream one by one,
- the linear dependence relations between the values taken by these forced useful bits and the neighboring useful information bits (variable and random) are also determined by reassembling the transformations of the bit stream one by one.

If all stages of the transmission chain are affected, the key steps relate to the inversion of the interleaving and correction coding operations.

Remaining on the scale of a signal frame at the transmission chain output, the positions and dependence relations of the fake useful bits in the stream of useful information are fixed. These positions and dependence relations can be computed once and for all and then applied for computation and positioning of the forced bits on each successive frame.

On reception, the demodulated and decoded data are recovered frame by frame. The fake useful bits that have been used to generate the fake sequences after encoding have a position that is known *a priori* to the receiver, which removes them directly from the demodulated/decoded stream of information bits.

A second variant embodiment of the invention involves applying the invention to a subset that is situated downstream of the transmission chain and made up of blocks in series whose transfer functions are all reversible. In practice, the portion of maximum length from the transmission chain for which the reversal of the bit stream can be performed satisfactorily is identified. This involves returning from the signal at the modulator output to

the sequence of useful data for transmission. The bit stream intended to produce modulated signals $D(t)$, containing a fake synchronization sequence, is injected only at the input of this subassembly, in other words at the output of the first irreversible function of the transmission chain in the reverse sequential order of the transmission functions (from the modulator – end of the chain – to the correction encoder – start of the chain). This second variant of the invention is applied, by way of example, if the transmission chain has an encryption or cryptography device whose function has an indeterminable inverse by nature.

10

The method according to the invention can be used with fake sequences ($SYNC_F$) having significant combinations, and periodicities or slow variations, so as to maintain the analysis combination over time and to disadvantage the capability of identification and storage of the “interception” function of the decoyed interception and jamming system.

15

The method according to the invention can be implemented on the basis of software elements. Such software can be executed by the transmitting equipment so as to modify the useful binary sequence for transmission that is produced by an application, for example. It can likewise be executed by a computer connected to said transmitter for the purpose of configuration thereof.

20

The method according to the invention can be available as a computer program product on a computer-readable medium. The medium may be electronic, magnetic, optical, electromagnetic or may be a broadcasting medium of infrared type. By way of example, such media are semiconductor memories (Random Access Memory RAM, Read-Only Memory ROM), tapes, floppy disks or magnetic or optical disks (Compact Disk – Read Only Memory (CD-ROM), Compact Disk – Read/Write (CD-R/W) and DVD).

30

The invention can be used for any communication system that needs to be made more robust toward interception and jamming; in particular, the invention can be used for retrofitting older-generation communication systems that have therefore become vulnerable to modern interceptors and
5 jammers.

CLAIMS

1. A method for decoying against an interception and/or jamming system, involving generating, in a signal to be transmitted, at least one fake sequence (SYNC_F) so as to decoy said system while preserving the synchronization with a receiver of said transmitted signal, said method being characterized in that it comprises the following steps:
 - defining said fake sequence (SYNC_F) and its time and/or frequency position within the signal to be transmitted, the values of the symbols of said fake sequence and their time and/or frequency positions being different from those of the symbols of a synchronization sequence (SYNC) that said signal has,
 - estimating the value and the position of the fake bits (B_F) to be inserted within the data sequence ($D_{U,T}$) to be transmitted at the input of the transmission chain or of a subportion of the transmission chain so as to obtain, in the sequence produced at the output of the transmission chain, the predefined value and time position of the symbols of said fake sequence (SYNC_F),
 - inserting, within said data sequence ($D_{U,T}$), the fake bits (B_F) at the positions obtained.
2. The method for decoying as claimed in claim 1, in which, when the transfer function F of the transmission chain is reversible, the value and the position of the fake bits (B_F) are estimated by determining the inverse transfer function F^{-1} of said transfer function F and by applying said inverse transfer function F^{-1} to said fake sequence (SYNC_F).
3. The method for decoying as claimed in claim 2, in which the inverse transfer function F^{-1} of said transmission chain is determined by

composing, in reverse order, the inverse transfer functions of the various blocks making up said chain.

4. The method for decoying as claimed in claim 3, in which the transmission chain has at least one correction code (402) with a yield of k/n for which the inversion of its transfer function is performed by resolving the system of equations according to

$$\begin{bmatrix} i_0 & \dots & i_{k-1} \end{bmatrix} = \begin{bmatrix} c_{p_1} & \dots & c_{p_d} \end{bmatrix} \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \text{ where } [c_{p_1} \dots c_{p_d}]$$

- are the symbols whose value is fixed in the encoded sequence, $[i_0, \dots, i_{k-1}]$ is the sequence to be produced at the input of said code, and m_{i,p_j} are the coefficients of the generator matrix for said code, $p_1, p_2, \dots, p_d$ being the set of positions for the symbols whose value is fixed in the encoded sequence, with d being an integer at most equal to the number n of symbols in the encoded sequence.

5. The method for decoying as claimed in claim 4, in which said correction code is a linear block code or convolutional code or a turbocode or an LDPC low-density code.
6. The method for decoying as claimed in one of claims 3 to 5, in which the transmission chain moreover has a jumbler (403) and/or an interleaver (404) and/or a framing module (405) and/or a signal binary encoder (406) and/or a modulator (407).
7. The method for decoying as claimed in claim 1, in which, when the transfer function F of the transmission chain is nonsurjective, the value and the position of the fake bits are estimated by looking for the input sequence T' of the transmission chain that minimizes a distance criterion between the sequence $F(T')(t)$ obtained at the transmission chain output

when said sequence T' is actually produced at its input and said fake sequence ($SYNC_F$).

8. The method for decoying as claimed in claim 7, in which said distance
5 criterion is taken to be equal to the integral, over a given duration, of the squared norm for the difference between the sequence $F(T')(t)$ obtained at the transmission chain output and said fake sequence ($SYNC_F$).
9. The method for decoying as claimed in either of claims 7 and 8, in which
10 the input sequence T' of the transmission chain that minimizes said distance criterion is looked for over a subset of the set of binary sequences that are possible at the input of the transmission chain.
10. The method for decoying as claimed in one of the preceding claims, in
15 which said data sequence (D_U, T) is produced by an application (401), for example an audio, image or video encoder.
11. The method for decoying as claimed in claim 1, in which, when the
20 transfer function F of the transmission chain is nonsurjective, the fake bits (B_F) are inserted into the sequence produced at the input of a subportion of the transmission chain whose transfer function is surjective and whose output is common to the output of the transmission chain.
12. The method for decoying as claimed in claim 11, in which said
25 transmission chain has an encryption device and said subportion of the transmission chain excludes this device.
13. The method for decoying as claimed in one of the preceding claims, in
30 which said fake sequence ($SYNC_F$) is positioned in an invulnerable time and/or frequency area of the signal to be transmitted.
14. The method for decoying as claimed in claim 13, in which said
invulnerable area is a portion of the signal that is protected by a correction code or is a stuffing area containing no useful information.

15. The method for decoying as claimed in one of the preceding claims, in which, when the signal comprising said fake sequence (SYNC_F) is received by said receiver, the fake bits (B_F) are removed from the data sequence ($D_{U,T}$).
- 5
16. A device for transmitting a signal having a transmission chain for transforming a data sequence ($D_{U,T}$) to be transmitted into a signal to be transmitted and a means that is suited to implementing the method as claimed in one of claims 1 to 15.
- 10
17. A computer program having instructions for executing the method as claimed in any one of claims 1 to 15 when the program is executed by a processor.
- 15
18. A processor-readable recording medium on which is recorded a program having instructions for executing the method as claimed in any one of claims 1 to 15 when the program is executed by a processor.