

METHOD AND APPARATUS FOR SCRAMBLING SEQUENCE GENERATION IN A COMMUNICATION SYSTEM

CLAIM OF PRIORITY UNDER 35 U.S.C. §119

[0001] This application claims the benefit of United States Provisional Patent Application No. 61/039,713, entitled METHOD AND APPARATUS FOR SCRAMBLING SEQUENCE GENERATION IN A COMMUNICATION SYSTEM, and filed on March 26th, 2008, the entirety of which is incorporated herein by reference.

BACKGROUND

I. Field

[0002] The following description relates generally to wireless communications systems, and more particularly to scrambling sequence generation in a wireless communications system.

II. Background

[0003] Wireless communication systems are widely deployed to provide various types of communication content such as voice, data, and so forth. These systems may be multiple-access systems capable of supporting communication with multiple users by sharing the available system resources (e.g., bandwidth and transmit power). Examples of such multiple-access systems include code division multiple access (CDMA) systems, time division multiple access (TDMA) systems, frequency division multiple access (FDMA) systems, 3GPP Long Term Evolution (LTE) systems including E-UTRA, and orthogonal frequency division multiple access (OFDMA) systems.

[0004] An orthogonal frequency division multiplex (OFDM) communication system effectively partitions the overall system bandwidth into multiple (N_F) subcarriers, which may also be referred to as frequency sub-channels, tones, or frequency bins. For an OFDM system, the data to be transmitted (i.e., the information bits) is first encoded with a particular coding scheme to generate coded bits, and the coded bits are further grouped into multi-bit symbols that are then mapped to modulation symbols. Each modulation symbol corresponds to a point in a signal constellation defined by a particular modulation scheme (e.g., M-PSK or M-QAM) used

for data transmission. At each time interval that may be dependent on the bandwidth of each frequency subcarrier, a modulation symbol may be transmitted on each of the N_F frequency subcarrier. Thus, OFDM may be used to combat inter-symbol interference (ISI) caused by frequency selective fading, which is characterized by different amounts of attenuation across the system bandwidth.

[0005] Generally, a wireless multiple-access communication system can concurrently support communication for multiple wireless terminals that communicate with one or more base stations via transmissions on forward and reverse links. The forward link (or downlink) refers to the communication link from the base stations to the terminals, and the reverse link (or uplink) refers to the communication link from the terminals to the base stations. This communication link may be established via a single-in-single-out, multiple-in-single-out or a multiple-in-multiple-out (MIMO) system.

[0006] A MIMO system employs multiple (N_T) transmit antennas and multiple (N_R) receive antennas for data transmission. A MIMO channel formed by the N_T transmit and N_R receive antennas may be decomposed into N_S independent channels, which are also referred to as spatial channels, where $N_S \leq \min\{N_T, N_R\}$. Generally, each of the N_S independent channels corresponds to a dimension. The MIMO system can provide improved performance (e.g., higher throughput and/or greater reliability) if the additional dimensionalities created by the multiple transmit and receive antennas are utilized. A MIMO system also supports time division duplex (TDD) and frequency division duplex (FDD) systems. In a TDD system, the forward and reverse link transmissions are on the same frequency region so that the reciprocity principle allows estimation of the forward link channel from the reverse link channel. This enables an access point to extract transmit beam-forming gain on the forward link when multiple antennas are available at the access point.

[0007] In Long Term Evolution (LTE) systems, Gold sequences are used for various randomization purposes such as uplink (UL) VRB-to-PRB mapping, sequence scrambling (virtual resource block and physical resource block), random sequence generation, UL demodulation (DM) reference signal (RS) index hopping, and so forth. The sequences can be individualized by setting the initial states of the component shift registers to different values. In certain cases, such as the decoding of the physical broadcast channel (PBCH), multiple sequence time shift hypotheses need to be tested, which necessitates that multiple cyclic shifts of the same sequence need to be generated

at the same time. Another common problem with the accepted Gold sequence design is that the first few dozen sequence bits are not sufficiently random. Thus, sequences that are initialized with similar values can yield similar sequence bits up to the length of the component sequence generator shift registers. This is aggravated by the fact that some of the sequences generated are relatively short, thus the insufficiently random initial segment is not an insignificant portion of the sequence length.

SUMMARY

[0008] The following presents a simplified summary in order to provide a basic understanding of some aspects of the claimed subject matter. This summary is not an extensive overview, and is not intended to identify key/critical elements or to delineate the scope of the claimed subject matter. Its sole purpose is to present some concepts in a simplified form as a prelude to the more detailed description that is presented later.

[0009] Systems and methods provide concurrent random sequence generation for multiple cyclic shift hypotheses. In order to improve randomization, sequence generators can be fast forwarded to a specified future state, where respective sequence bits from the generators can be output from then on. To perform this fast forward function efficiently, it is beneficial to provide a method that can jump to a future state in a substantially rapid manner. Different cyclic shifts of Gold sequences can be generated by masking shift register output values with desired vectors and with modulo-2 adding the result, for example. This method can also be employed to generate multiple cyclic shift copies of the Gold sequence at about the same time, if desired. The masking vector can be derived from the sequence generator polynomial and the desired cyclic shift. In general, the mask for two component m-sequences of the Gold sequence generator can be different. Sequence and randomization components can also be paralleled to improve system performance.

[0010] To the accomplishment of the foregoing and related ends, certain illustrative aspects are described herein in connection with the following description and the annexed drawings. These aspects are indicative, however, of but a few of the various ways in which the principles of the claimed subject matter may be employed and the claimed subject matter is intended to include all such aspects and their equivalents. Other advantages and novel features may become apparent from the following detailed description when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0011] FIG. 1 is a high level block diagram of a system that employs scrambling sequence components in a wireless communications environment.
- [0012] FIG. 2 is a system that illustrates an exemplary scrambling sequence generator for a wireless system.
- [0013] FIG. 3 illustrates example sequence processing aspects for a wireless communications system.
- [0014] FIG. 4 illustrates an alternative scrambling sequence generator.
- [0015] FIG. 5 illustrates a wireless communications method for scrambling sequence generation.
- [0016] FIG. 6 illustrates an example logical module for a wireless protocol.
- [0017] FIG. 7 illustrates an example logical module for an alternative wireless protocol.
- [0018] FIG. 8 illustrates an example communications apparatus that employs a wireless protocol.
- [0019] FIG. 9 illustrates a multiple access wireless communication system.
- [0020] FIGS. 10 and 11 illustrate example communications systems.

DETAILED DESCRIPTION

- [0021] Systems and methods are provided to generate random sequences for wireless communications in an efficient manner. In one aspect, a wireless communications method is provided. The method includes employing a processor executing computer executable instructions stored on a computer readable storage medium to implement various acts. The method also includes generating cyclic shifts for a sequence generator by masking shift register output values with one or more vectors. The method includes forwarding the sequence generator to a future state based in part on the output values and the vectors.
- [0022] Referring now to **Fig. 1**, scrambling sequence components are employed for a wireless communications system. The system 100 includes one or more base stations 120 (also referred to as a node, evolved node B – eNB, femto station, pico station, and so forth) which can be an entity capable of communication over a wireless network 110 to a second device 130 (or devices). For instance, each device 130 can be

an access terminal (also referred to as terminal, user equipment, mobility management entity (MME) or mobile device). The base station 120 communicates to the device 130 via downlink 140 and receives data via uplink 150. Such designation as uplink and downlink is arbitrary as the device 130 can also transmit data via downlink and receive data via uplink channels. It is noted that although two components 120 and 130 are shown, that more than two components can be employed on the network 110, where such additional components can also be adapted for the wireless protocols described herein. As shown, a scrambling sequence component 160 and 170 respectively (or components) is provided to generate random Gold (or other type) sequences in an efficient manner. It is noted that as used herein, the term scrambling sequence component 160 or 170 can include generator and/or decoder aspects. For example, the component 160 can be a generator of random sequences, whereas the component 170 could be a decoder of random sequences, for example.

[0023] In general, the scrambling sequence components 160 and 170 provide concurrent random sequence generation for multiple cyclic shift hypotheses. In order to improve randomization, sequence generators (or other components) can be fast forwarded to a specified future state (as shown and described with respect to Fig. 2 below), where respective sequence bits from the generators can be output from then on. To perform this fast forward function efficiently, it is beneficial to provide a method that can jump to a future state in a substantially rapid manner. A plurality of different cyclic shifts of Gold sequences can be generated by masking shift register output values with desired vectors and with modulo-2 adding the result, for example. This method can also be employed to generate multiple cyclic shift copies of the Gold sequence at about the same time, if desired. The masking vector can be derived from the sequence generator polynomial and the desired cyclic shift. In general, the mask for two component m-sequences of the Gold sequence generator can be different. Sequence and randomization components can also be paralleled to improve system performance as illustrated in an example system depicted in Fig. 4. As will be described in more detail below, various m-codes can be selected to determine a desired fast forward starting point for the sequence generators. Various g-codes can be selected to generate different random sequences for different base stations 120, for example.

[0024] It is noted that the system 100 can be employed with an access terminal or mobile device, and can be, for instance, a module such as an SD card, a network card,

a wireless network card, a computer (including laptops, desktops, personal digital assistants (PDAs)), mobile phones, smart phones, or any other suitable terminal that can be utilized to access a network. The terminal accesses the network by way of an access component (not shown). In one example, a connection between the terminal and the access components may be wireless in nature, in which access components may be the base station and the mobile device is a wireless terminal. For instance, the terminal and base stations may communicate by way of any suitable wireless protocol, including but not limited to Time Divisional Multiple Access (TDMA), Code Division Multiple Access (CDMA), Frequency Division Multiple Access (FDMA), Orthogonal Frequency Division Multiplexing (OFDM), FLASH OFDM, Orthogonal Frequency Division Multiple Access (OFDMA), or any other suitable protocol.

[0025] Access components can be an access node associated with a wired network or a wireless network. To that end, access components can be, for instance, a router, a switch, or the like. The access component can include one or more interfaces, *e.g.*, communication modules, for communicating with other network nodes. Additionally, the access component can be a base station (or wireless access point) in a cellular type network, wherein base stations (or wireless access points) are utilized to provide wireless coverage areas to a plurality of subscribers. Such base stations (or wireless access points) can be arranged to provide contiguous areas of coverage to one or more cellular phones and/or other wireless terminals.

[0026] Referring now to **Fig. 2**, a system 200 illustrates an example sequence generator for a wireless system. A PRN scrambling sequence generator structure is illustrated by the system 200. The system 200 is flexible to enable generating scrambling sequences for current applications and also for new applications that may be added in the future. At various instances of the transmit signal generation process, scrambling is to be applied. This is to avoid persistent interference between different signals and to avoid undesired signal spectrum properties. For certain signals, it is beneficial to have the scrambling associated with the resource element the signal occupies, for other type of scrambling sequence applications, it is desirable to have the scrambling sequence to be disassociated from the resource element occupied. In this aspect, scrambling sequence generation is provided that can be used for substantially all applications.

[0027] A binary m-sequence can be employed as the basic scrambling code. For different purposes, different cyclic shifts of the same sequence are used. The underlying assumption is that the different shifts of the same sequence are sufficiently de-correlated. The shift register sequence length should be sufficiently large. In the example shown at 200, assume a 50-bit shift register, which is capable of generating a sequence with a period of 2^{50} . The generating polynomial

$G(x) = x^{50} + g_{49}x^{49} + g_{48}x^{48} + \dots + g_2x^2 + g_1x + 1$ can be the same for all applications, which avoids having to reconfigure the shift registers. The shift register can be set to the same initial stage for each application, if desired. The different cyclic shifts can be achieved by modulo-2 summing certain register outputs, where the choice of which register outputs to be included in the sum controls the cyclic shift selection. An example architecture is provided by the system 200. Note that each of the polynomial coefficients (at reference numeral 210) $g_{49} \dots g_2, g_1$ represents a connection if the coefficient is '1' and no connection if the coefficient is '0'.

[0028] As noted above, the scrambling sequences can be individualized by the choice of the register outputs. Since 50 shift registers (or other number) are available, 50 selector bits can be provided, yielding 2^{50} different shifts. The allocation of the controlling 50 bits is defined as follows. The 50 bits are divided into the first 2 bits that are reserved, the next 4 bits that are the channel/signal type, and the remaining 44 bits that are allocated in a channel/signal-specific manner. This is shown in Table 1 below.

Field	Sequence selector assignment	Number of bits
Reserved	$m_{48} \dots m_{49}$	2
Channel/signal type	$m_{44} \dots m_{47}$	4
Channel/signal-specific fields	$m_0 \dots m_{43}$	44

Table 1 Scrambling Selector Bit Designation

The channel/signal types are enumerated as listed in Table 2 below:

Channel/Signal	Channel type value
PRS (normal CP)	'0000'
PRS (extended)	'0001'

CP)	
PDCCH	'0010'
PCFICH	'0011'
PHICH	'00100'
PBCH	'0101'
PMCH	'0110'
PDSCH	'0111'
PUSCH	'1000'
Other	Reserved

Table 2 Channel Type Value Assignment

Note that 10ms periodicity can be assumed for: PRS (normal and extended CP), PDCCH, PDSCH, PUSCH. Also, 40ms periodicity can be assumed for PBCH. For PCFICH, PHICH and PMCH a decision on the scrambling periodicity is to be made. The channel-specific fields can be defined separately for each channel type as shown below.

PRS (normal CP) signal specific fields:

Parameter	Number of bits
SSC_ID	8
Antenna_ID	2
Subframe_ID	4
Symbol_ID	4
Frequency +/-	1
Reserved	25

Table 3 PRS (normal CP) signal-specific Fields

PRS (extended CP) signal specific fields:

Parameter	Number of bits
Cell_ID	9
Antenna_ID	2
Subframe_ID	4
Symbol_ID	4
Frequency +/-	1
Reserved	24

Table 4 PRS (extended CP) signal-specific Fields

PDCCH Channel-specific fields:

Parameter	Number of bits
Cell_ID	9
Subframe_ID	4
Symbol_ID	4
Reserved	27

Table 5 PDCCH Channel-specific Fields

PDSCH Channel-specific fields:

Parameter	Number of bits
Cell_ID	9
UE MAC_ID	16
Stream_ID	1
Code block ID	6
Reserved	12

Table 6 PDSCH Channel-specific Fields

Note that the table above assumes the possibility to have a PDSCH scrambling that is a function of the Cell_ID as well as of the UE MAC_ID.

PBCH Channel-specific fields:

Parameter	Number of bits
Cell_ID	9
Frame_ID	2
Subframe_ID	4
Symbol_ID	4
Reserved	25

Table 7 PBCH Channel-specific Fields

PCFICH Channel-specific fields:

Parameter	Number of bits
Cell_ID	9
Subframe_ID	4
Reserved	31

Table 8 PBCH Channel-specific Fields

PHICH Channel-specific fields:

Parameter	Number of bits
Cell_ID	9
xx	xx
Reserved	xx

Table 9 PBCH Channel-specific Fields

PMCH Channel-specific fields:

Parameter	Number of bits
Cell_ID	9
xx	xx
Reserved	xx

Table 10 PMCH Channel-specific Fields

PUSCH Channel-specific fields:

Parameter	Number of bits
UE MAC_ID	16
Code block ID	6
Reserved	22

Table 11 PUSCH Channel-specific Fields

[0029] The sequence generator can be reset at the beginning of each application of scrambling. This can be performed once at each symbol for the downlink (DL) RS, and once for the code block in the case of the PDSCH, for example. For the scrambling of binary encoded bits, one scrambling bit can be taken for each encoded bit. For the generation of the PRS sequences, two scrambling sequences can be generated, distinguished by the Frequency +/- bit. The first sequence can be used to scramble the 'positive frequencies' starting from the smallest positive frequency and mapped to DL RS tone indices in the order of increasing frequency. The second sequence can be used to scramble the 'negative frequencies' starting from highest negative frequency (e.g., closest to DC) and mapped to the DL RS tone indices in the opposite order. This allows that the PRS in the center of the system band is the same irrespective of the system bandwidth. Also, it does not require generating the entire scrambling sequence at each OFDM symbol carrying PRS for the possible PRS length. The proposed structure enables using a single shift register to generate the scrambling sequences. This assumes that the shift register is clocked as many times as required for the longest scrambling sequence. For shorter sequences, the initial part corresponding to the appropriate length is taken. It is also possible to have multiple instances of the shift register, one for each application, which may fit certain hardware architectures more suitably.

[0030] Turning to **Fig. 3**, example sequence processing aspects 300 are illustrated. In an LTE system, in particular the LTE uplink (UL), the resources used for demodulation reference signal (DM RS) and the various control channel spreading signals are to be randomized. This following describes mechanisms and techniques to

achieve this goal by utilizing various pseudorandom sequences and deterministic sequences. Optionally, techniques to support an easy-to-implement cell planning are described. Alternatively, uncoordinated allocation can also be supported by the mechanisms described herein.

[0031] Various uplink (UL) sequence hopping aspects are provided. The following design criteria may be applied:

- Simple arithmetic to compute Physical Uplink Control Channel (PUCCH) and Physical Uplink Shared Channel (PUSCH) allocation parameters in every symbol
- Flexible PUCCH cyclic shift – orthogonal cover allocation. The hopping pattern is independent of the overall allocation strategy (CS-OC map). The UE determines its own initial parameter set; it does not need to determine what strategy was used to optimize the cyclic shift and orthogonal cover allocation.
- A single set of rules for either the cell specific hopping or the resource specific hopping case

[0032] At 310 of Fig. 3, sequence hopping considerations are provided. For the PUCCH and PUSCH, a set of possible reference signal (RS) sequences denoted by their sequence index can be defined for each possible resource block (RB) allocation case. In the case of the PUCCH, the same set of sequences can also be used to convey control information.

[0033] Assume the following:

- For $N_{RB} \leq 5$, there are 30 sequence indices available. There are 30 sequence groups, with one sequence in each group (numbers other than 30 can also be employed)
- For $N_{RB} > 5$, there are 60 sequence indices available. There are 30 sequence groups with two sequences in each group

Assume that there is a single downlink (DL) signaling bit informing the user equipment (UE) about whether sequence hopping should be used or not. In the following, sequence hopping and no hopping cases are described separately.

[0034] With sequence hopping disabled, the UE uses the PUSCH RS sequence index(es) corresponding to the signaled sequence group.

- For $N_{RB} \leq 5$, the UE uses a single sequence index (one of 30).

- For $N_{RB} > 5$, the UE uses the first sequence index in the signaled sequence group in the first slot of the subframe and it uses the second sequence index in the signaled sequence group in the second slot of the subframe. Therefore, the UE alternates between the two sequences defined for the sequence group.

If it is desired to have more sequences (e.g., more than two) per sequence group for some $N_{RB} > 5$ then the UE cycles through the sequence indices in a similar manner. If there are m indices per sequence group, e.g., the set of indexes is $\{k_0, k_1, \dots, k_{m-1}\}$ in a given sequence group, then in the i^{th} slot of a frame, the UE would use the sequence with index $k_{i \bmod m}$. In the first slot of a frame, k_0 would be used.

[0035] When sequence hopping is disabled, the UE uses a single sequence based on the signaled sequence group for the RS and the control data modulation. The generic scrambling sequence generator (Gold sequence) can be used in order to generate the index hopping sequence. With sequence hopping enabled, the UE employs the PUSCH RS sequence index as determined by the scrambling sequence generator output. The sequence generator can be initialized at each subframe boundary and clocked once in each slot, for example. At initialization, the 33-bit seed sequence is constructed according to the following:

Initializer Bit	$b_{32} \dots b_{30}$	$b_{29} \dots b_{27}$	$b_{26} \dots b_{13}$	$b_{12} \dots b_9$	$b_8 \dots b_0$
Value	0,0,0	0,0,1	0,0,...,0	Subframe_ID	Cell_ID

Note that since the subframe ID is part of the initialization bits, the resulting sequence period is one frame (10ms). Suppose the scrambling generator output is $s_0, s_1, \dots, s_{8-u}$ where u is the number of slots per frame, then the PUSCH sequence index k_i in slot i is determined as $k_i = \left(\sum_{l=0}^7 s_{8-i+l} \cdot 2^l \right) \bmod (m \cdot 30)$ (e.g., taking consecutive bytes of the scrambling sequence, one for each slot and take the corresponding integer value modulo the total number of sequence indices) where m is the number of sequences indices per sequence group. Note that:

$$m = \begin{cases} 1 & N_{RB} \leq 5 \\ 2 & N_{RB} > 5 \end{cases}$$

[0036] With sequence hopping enabled, the UE employs the PUCCH RS and control sequence index as determined by the scrambling sequence generator output.

The sequence generator is initialized at each subframe boundary and clocked once for each symbol, for example. At initialization, the 33-bit seed sequence can be constructed according to the following:

Initializer Bit	$b_{32}...b_{30}$	$b_{29}...b_{27}$	$b_{26}...b_{13}$	$b_{12}...b_9$	$b_8...b_0$
Value	0,0,0	0,0,1	0,0,...,0	Subframe_ID	Cell_ID

Note that the subframe ID is part of the initialization bits, the resulting sequence period is one frame (10ms). Suppose the scrambling generator output is $s_0, s_1, \dots, s_{8v}$ where v is the number of symbols per frame, then the PUCCH CGS sequence index k_i in symbol i

is determined as $k_i = \left(\sum_{l=0}^7 s_{8 \cdot i + l} \cdot 2^l \right) \bmod 30$. Note that from the sequence index

generation purposes, the RS and control symbols within the PUCCH are not distinguished.

[0037] At 320 of Fig. 3, cell-specific shift hopping considerations are described.

In general, cyclic shift hopping is not provided for the PUSCH RS. The cyclic shift is either explicitly signaled in the assignment or otherwise it is set to a static value conveyed by higher layer signaling. For the purpose of inter-cell interference randomization, a cell specific cyclic shift offset sequence can be provided. In order to simplify implementation, assume that for the cell specific cyclic shift application purposes, the RS and control symbols within the PUCCH are not distinguished.

Let l_i be the cyclic shift offset in symbol i . Assume $l_i \in \{0, 1, 2, \dots, 11\}$. And if the cyclic shift in a symbol is u_i before applying the cell specific cyclic shift offset then it will be $(l_i + u_i) \bmod 12$ after applying the cell specific cyclic shift offset. Two options for generating the l_i are described in the following section.

[0038] In this case, the cyclic shift offset pattern depends on the Cell_ID, the cell specific cyclic shift offset can be determined by the scrambling sequence generator output. The sequence generator can be initialized at each subframe boundary and clocked once in each symbol. At initialization, the 33-bit seed sequence can be constructed according to the following:

Initializer Bit	$b_{32}...b_{30}$	$b_{29}...b_{27}$	$b_{26}...b_{13}$	$b_{12}...b_9$	$b_8...b_0$
Value	0,0,0	0,1,0	0,0,...,0	Subframe_ID	Cell_ID

Note that since the subframe ID is part of the initialization bits, the resulting sequence period is one frame (10ms). Suppose the scrambling generator output is $s_0, s_1, \dots, s_{8-v}$ where v is the number of symbols per frame, then the cell specific cyclic shift offset l_i in symbol i is determined as $l_i = \left(\sum_{b=0}^7 s_{8-i+b} \cdot 2^b \right) \bmod 12$ e.g., taking consecutive bytes of the scrambling sequence, one for each symbol and take the corresponding integer value modulo 12.

[0039] The cyclic shift offset is generally the sum of two components; the first is a pseudo-random sequence dependent on the (secondary sequence) SSC_ID, while the second is a deterministic sequence dependent on the (primary sequence) PSC_ID. The purpose of this construction is to minimize the cyclic shift alignments in cells with the same SSC_ID. The pseudorandom cyclic shift offset component, t_i is determined by the scrambling sequence generator output. The sequence generator is initialized at each subframe boundary and clocked once in every symbol, for example. At initialization, the 33-bit seed sequence can be constructed according to the following:

Initializer Bit	$b_{32} \dots b_{30}$	$b_{29} \dots b_{27}$	$b_{26} \dots b_{13}$	$b_{12} \dots b_9$	$b_8 \dots b_0$
Value	0,0,0	0,1,1	0,0,...,0	Subframe_ID	SSC_ID

Note that since the subframe ID is part of the initialization bits, the resulting sequence period is one frame (10ms). Suppose the scrambling generator output is $s_0, s_1, \dots, s_{8-v}$ where v is the number of symbols per frame, then the cell specific cyclic shift offset t_i in symbol i is determined as $t_i = \left(\sum_{b=0}^7 s_{8-i+b} \cdot 2^b \right) \bmod 12$ e.g., taking consecutive bytes of the scrambling sequence, one for each symbol and take the corresponding integer value modulo 12. The PSC_ID dependent deterministic cyclic shift offset value r_j $0 \leq j < 12$ is defined as:

$$(r_0, r_1, \dots, r_{11}) = \begin{cases} (0,0,0,0,0,0,0,0,0,0,0,0) & \text{if } PSC_ID = 0 \\ (0,1,3,7,2,5,11,10,8,4,9,6) & \text{if } PSC_ID = 1 \\ (0,12,10,6,11,8,2,3,5,9,4,7) & \text{if } PSC_ID = 2 \end{cases}$$

Note that a single formula is provided for generating the sequences above. Also note that the sum of r_j for PSC_ID=1 and PSC_ID=2 is zero modulo 13. For any pair of

PSC_IDs, the element-wise shift differences are distinct. The cell specific cyclic shift offset l_i in symbol i is determined as $l_i = (t_i + r_{i \bmod 6}) \bmod 12$

[0040] At 330 of Fig. 3, resource-specific cyclic shift hopping is provided. The resource specific cyclic shift hopping can be performed on a per symbol basis. The hopping pattern is based on a factor 3 decimation. The resource specific cyclic shift c_j in control data symbol j is determined as $c_j = 2 \cdot ((\lfloor c_0 / 2 \rfloor + 1) \cdot 3^{j+c_0 \bmod 2}) \bmod 7 + c_0 \bmod 2$. In the first symbol of every frame, $j=0$. After that, j is incremented by one for every control symbol but it is not incremented for RS symbols. The resource specific cyclic shift c_k in RS symbol k is determined as $c_k = 2 \cdot ((\lfloor c_0 / 2 \rfloor + 1) \cdot 3^{k+c_0 \bmod 2}) \bmod 7 + c_0 \bmod 2$. In the first RS symbol of the frame, $k=0$. After that, k is incremented by one for each RS symbol but it is not incremented for control data symbols.

[0041] At each slot boundary, the cyclic shift allocation is offset according to a deterministic pattern. The purpose of this is to maximize the distance in a new slot between resources that were sharing the same cyclic shift resource in the previous slot. Resource hopping is achieved by adding a slot and resource dependent cyclic shift offset d_i^j for slot i and orthogonal cover index j . The cyclic shift offset d_i^j for slot i and orthogonal cover index j is determined as

$$d_i^j = \begin{cases} 0 & \text{if } j = 0 \\ (2 \cdot i \cdot (j+1)) \bmod 12 & \text{if } j > 0 \end{cases}$$

For which the orthogonal cover index j is mapped to spreading sequences as shown below

$$\begin{cases} j = 0 & +1+1+1+1 \\ j = 1 & +1-1+1-1 \\ j = 2 & +1+1-1-1 \\ j = 3 & +1-1-1+1 \end{cases}$$

The cyclic shift offset d_i^j for slot i and orthogonal cover index j is determined as

$$d_i^j = (4 \cdot i \cdot j) \bmod 12$$

For which the orthogonal cover index j is mapped to spreading sequences as shown below

$$\begin{cases} j = 0 & 1 & 1 & 1 \\ j = 1 & 1 & e^{i2\pi/3} & e^{-i2\pi/3} \\ j = 2 & 1 & e^{-i2\pi/3} & e^{i2\pi/3} \end{cases}$$

[0042] At 340 of Fig. 3, orthogonal cover hopping aspects are provided.

Generally, the orthogonal cover is changed at each slot boundary. The relationship between the orthogonal cover functions associated with a pair of PUCCH resources is

generally the same across slot boundaries, however, a cell dependent linear offset can be applied to each orthogonal cover function. The offset implies that a cell specific cover function is added (element-wise multiply) to each orthogonal cover function used in the cell. This approach preserves the optimum distribution of orthogonal covers that may have been used. The cell specific orthogonal cover offset index is determined by the scrambling sequence generator output. The sequence generator is initialized at every subframe boundary and clocked once in every slot. At initialization, the [33]-bit seed sequence can be constructed according to the following:

Initializer Bit	$b_{32}...b_{30}$	$b_{29}...b_{27}$	$b_{26}...b_{13}$	$b_{12}...b_9$	$b_8...b_0$
Value	0,0,0	1,0,0	0,0,...,0	Subframe_ID	Cell_ID

Note that since the subframe ID is part of the initialization bits, the resulting sequence period is one frame (10ms). Suppose the scrambling generator output is $s_0, s_1, \dots, s_{8-u}$ where u is the number of slots per frame, then the cell specific orthogonal cover offset

index d_i for the ACK data in slot i is determined as $d_i = \left(\sum_{b=0}^7 s_{8-i+b} \cdot 2^b \right) \bmod 4$ while the

cell specific orthogonal cover offset index e_i for the RS in slot i is determined as

$e_i = \left(\sum_{b=0}^7 s_{8-i+b} \cdot 2^b \right) \bmod 3$ Then the actual applied orthogonal cover is the sum (element-wise product) of the initial assigned orthogonal cover and the orthogonal cover function indicated by d_i and e_i for the ACK data and ACK RS, respectively.

[0043] Referring to **Fig. 4**, an example alternative sequence generator 400 is illustrated. For various purposes, pseudorandom sequences can be used in the sequence hopping pattern generation. For this purpose, the structure shown in Fig. 4 can be used, where an h vector can be employed at 410. A shorter sequence generator could be used as well. The advantage of the structure shown in Fig. 4 is that a single generator can be used to generate all pseudorandom sequences. Various UL sequence hopping applications have been described. In general, a solution for both cell-specific and resource specific cyclic shift hopping is provided. This includes sequence index hopping pattern details and cell-coordinated cyclic shift hopping patterns. A cell-specific orthogonal cover offset index hopping pattern can also be provided. Simple arithmetic can be employed to compute PUCCH and PUSCH allocation parameters in the respective symbols. Use of a scrambling generator for various pseudo-random

sequences can be employed. Flexible PUCCH cyclic shift – orthogonal cover allocation can also be provided. The hopping pattern can be independent of the overall allocation strategy. The UE needs to determine its own initial parameter set; it does not need to determine what strategy was used to optimize the cyclic shift and orthogonal cover allocation. A single set of rules for either the cell specific hopping or the resource specific hopping case can be provided. These aspects can be applied to UL DM RS and PUCCH randomization specification in LTE, for example.

[0044] Referring now to **Fig. 5**, a wireless communications methodology 500 is illustrated. While, for purposes of simplicity of explanation, the methodology (and other methodologies described herein) are shown and described as a series of acts, it is to be understood and appreciated that the methodologies are not limited by the order of acts, as some acts may, in accordance with one or more embodiments, occur in different orders and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be utilized to implement a methodology in accordance with the claimed subject matter.

[0045] Proceeding to 510, an m-parameter is determined to select fast-forwarding operations. As noted previously, binary m-sequences can be employed as base-level scrambling codes. In one example, a 50 digit binary value can be employed by other values are also possible. At 520, different starting random sequences are selected. As noted previously, these values can be assigned for differing baser stations. Also, g-parameters or polynomials can be selected as noted above for the random sequences. At 530, the fast forward parameter at 510 is applied to a sequence selector. This can be applied via a collection of gates for example that apply the fast forward parameter to a modulo-2 adder for example. At 540, multiple m-sequences are combined to form a desired Gold sequence. For example, two (or more) m-sequences can be combined via an exclusive OR (XOR) operation to form the Gold sequence. At 550, a scrambling sequence is performed by shifting the combination of the m-parameters and the g-parameters described above.

[0046] In some cases, the use of multiple XOR operations for obtaining each new register output value can represent an undesirable degree of complexity. In these cases, it is advantageous to perform fast forwarding of the pseudorandom sequences by

setting the shift register stage to the desired future state. That state depends on the generating polynomial (g-parameters), the number of steps of fast forward required and the initial state. Since the future register state depends on the initial state, the future state for each possible initial state should be stored or obtained by some other means. One possible method for this is to use the m-parameters to generate a consecutive sequence of bits occurring just before the desired future state and then employ those bits and use them to initialize the shift registers. In this manner, the number of times the m-parameter controlled XOR operation is to be performed can be reduced from the length of the desired sequence to the length of the shift registers. This method can be efficient since the m-parameter to be used depends on the g-parameter and the time shift value but it does not depend on the shift register initial state. Therefore, storing a single m-parameter is sufficient for a given time advance.

[0047] The techniques described herein may be implemented by various means. For example, these techniques may be implemented in hardware, software, or a combination thereof. For a hardware implementation, the processing units may be implemented within one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), processors, controllers, micro-controllers, microprocessors, other electronic units designed to perform the functions described herein, or a combination thereof. With software, implementation can be through modules (e.g., procedures, functions, and so on) that perform the functions described herein. The software codes may be stored in memory unit and executed by the processors.

[0048] Turning now to **Figs. 6 and 7**, a system is provided that relates to wireless signal processing. The systems are represented as a series of interrelated functional blocks, which can represent functions implemented by a processor, software, hardware, firmware, or any suitable combination thereof.

[0049] Referring to **Fig. 6**, a wireless communication system 600 is provided. The system 600 includes a logical module 602 for shifting register output values with one or more vectors and a logical module 604 for generating one or more m-parameters in accordance with the vectors. The system 600 also includes a logical module 606 for setting a future state based in part on the output values, the m-parameters, and the vectors.

[0050] Referring to **Fig. 7**, a wireless communication system 700 is provided. The system 700 includes a logical module 702 for generating a first masking vector from a sequence generator polynomial and a first cyclic shift and a logical module 704 for generating a second masking vector from the sequence generator polynomial and a second cyclic shift. The system 700 also includes a logical module 706 for processing the first and second masking vector to mask shift register output values to obtain a first output value and a second output value that are employed to generate a Gold sequence.

[0051] **Fig. 8** illustrates a communications apparatus 800 that can be a wireless communications apparatus, for instance, such as a wireless terminal. Additionally or alternatively, communications apparatus 800 can be resident within a wired network. Communications apparatus 800 can include memory 802 that can retain instructions for performing a signal analysis in a wireless communications terminal. Additionally, communications apparatus 800 may include a processor 804 that can execute instructions within memory 802 and/or instructions received from another network device, wherein the instructions can relate to configuring or operating the communications apparatus 800 or a related communications apparatus.

[0052] Referring to **Fig. 9**, a multiple access wireless communication system 900 is illustrated. The multiple access wireless communication system 900 includes multiple cells, including cells 902, 904, and 906. In the aspect the system 900, the cells 902, 904, and 906 may include a Node B that includes multiple sectors. The multiple sectors can be formed by groups of antennas with each antenna responsible for communication with UEs in a portion of the cell. For example, in cell 902, antenna groups 912, 914, and 916 may each correspond to a different sector. In cell 904, antenna groups 918, 920, and 922 each correspond to a different sector. In cell 906, antenna groups 924, 926, and 928 each correspond to a different sector. The cells 902, 904 and 906 can include several wireless communication devices, e.g., User Equipment or UEs, which can be in communication with one or more sectors of each cell 902, 904 or 906. For example, UEs 930 and 932 can be in communication with Node B 942, UEs 934 and 936 can be in communication with Node B 944, and UEs 938 and 940 can be in communication with Node B 946.

[0053] Referring now to **Fig. 10**, a multiple access wireless communication system according to one aspect is illustrated. An access point 1000 (AP) includes multiple antenna groups, one including 1004 and 1006, another including 1008 and

1010, and an additional including 1012 and 1014. In Fig. 10, only two antennas are shown for each antenna group, however, more or fewer antennas may be utilized for each antenna group. Access terminal 1016 (AT) is in communication with antennas 1012 and 1014, where antennas 1012 and 1014 transmit information to access terminal 1016 over forward link 1020 and receive information from access terminal 1016 over reverse link 1018. Access terminal 1022 is in communication with antennas 1006 and 1008, where antennas 1006 and 1008 transmit information to access terminal 1022 over forward link 1026 and receive information from access terminal 1022 over reverse link 1024. In a FDD system, communication links 1018, 1020, 1024 and 1026 may use different frequency for communication. For example, forward link 1020 may use a different frequency than that used by reverse link 1018.

[0054] Each group of antennas and/or the area in which they are designed to communicate is often referred to as a sector of the access point. Antenna groups each are designed to communicate to access terminals in a sector, of the areas covered by access point 1000. In communication over forward links 1020 and 1026, the transmitting antennas of access point 1000 utilize beam-forming in order to improve the signal-to-noise ratio of forward links for the different access terminals 1016 and 1024. Also, an access point using beam-forming to transmit to access terminals scattered randomly through its coverage causes less interference to access terminals in neighboring cells than an access point transmitting through a single antenna to all its access terminals. An access point may be a fixed station used for communicating with the terminals and may also be referred to as an access point, a Node B, or some other terminology. An access terminal may also be called an access terminal, user equipment (UE), a wireless communication device, terminal, access terminal or some other terminology.

[0055] Referring to **Fig. 11**, a system 1100 illustrates a transmitter system 210 (also known as the access point) and a receiver system 1150 (also known as access terminal) in a MIMO system 1100. At the transmitter system 1110, traffic data for a number of data streams is provided from a data source 1112 to a transmit (TX) data processor 1114. Each data stream is transmitted over a respective transmit antenna. TX data processor 1114 formats, codes, and interleaves the traffic data for each data stream based on a particular coding scheme selected for that data stream to provide coded data.

[0056] The coded data for each data stream may be multiplexed with pilot data using OFDM techniques. The pilot data is typically a known data pattern that is processed in a known manner and may be used at the receiver system to estimate the channel response. The multiplexed pilot and coded data for each data stream is then modulated (i.e., symbol mapped) based on a particular modulation scheme (e.g., BPSK, QSPK, M-PSK, or M-QAM) selected for that data stream to provide modulation symbols. The data rate, coding, and modulation for each data stream may be determined by instructions performed by processor 1130.

[0057] The modulation symbols for all data streams are then provided to a TX MIMO processor 1120, which may further process the modulation symbols (e.g., for OFDM). TX MIMO processor 1120 then provides NT modulation symbol streams to NT transmitters (TMTR) 1122a through 1122t. In certain embodiments, TX MIMO processor 1120 applies beam-forming weights to the symbols of the data streams and to the antenna from which the symbol is being transmitted.

[0058] Each transmitter 1122 receives and processes a respective symbol stream to provide one or more analog signals, and further conditions (e.g., amplifies, filters, and up-converts) the analog signals to provide a modulated signal suitable for transmission over the MIMO channel. NT modulated signals from transmitters 1122a through 1122t are then transmitted from NT antennas 1124a through 1124t, respectively.

[0059] At receiver system 1150, the transmitted modulated signals are received by NR antennas 1152a through 1152r and the received signal from each antenna 1152 is provided to a respective receiver (RCVR) 1154a through 1154r. Each receiver 1154 conditions (e.g., filters, amplifies, and down-converts) a respective received signal, digitizes the conditioned signal to provide samples, and further processes the samples to provide a corresponding “received” symbol stream.

[0060] An RX data processor 1160 then receives and processes the NR received symbol streams from NR receivers 1154 based on a particular receiver processing technique to provide NT “detected” symbol streams. The RX data processor 1160 then demodulates, de-interleaves, and decodes each detected symbol stream to recover the traffic data for the data stream. The processing by RX data processor 1160 is complementary to that performed by TX MIMO processor 1120 and TX data processor 1114 at transmitter system 1110.

[0061] A processor 1170 periodically determines which pre-coding matrix to use (discussed below). Processor 1170 formulates a reverse link message comprising a matrix index portion and a rank value portion. The reverse link message may comprise various types of information regarding the communication link and/or the received data stream. The reverse link message is then processed by a TX data processor 1138, which also receives traffic data for a number of data streams from a data source 1136, modulated by a modulator 1180, conditioned by transmitters 1154a through 1154r, and transmitted back to transmitter system 1110.

[0062] At transmitter system 1110, the modulated signals from receiver system 1150 are received by antennas 1124, conditioned by receivers 1122, demodulated by a demodulator 1140, and processed by a RX data processor 1142 to extract the reserve link message transmitted by the receiver system 1150. Processor 1130 then determines which pre-coding matrix to use for determining the beam-forming weights then processes the extracted message.

[0063] In an aspect, logical channels are classified into Control Channels and Traffic Channels. Logical Control Channels comprises Broadcast Control Channel (BCCH) which is DL channel for broadcasting system control information. Paging Control Channel (PCCH) which is DL channel that transfers paging information. Multicast Control Channel (MCCH) which is Point-to-multipoint DL channel used for transmitting Multimedia Broadcast and Multicast Service (MBMS) scheduling and control information for one or several MTCHs. Generally, after establishing RRC connection this channel is only used by UEs that receive MBMS (Note: old MCCH+MSCH). Dedicated Control Channel (DCCH) is Point-to-point bi-directional channel that transmits dedicated control information and used by UEs having an RRC connection. Logical Traffic Channels comprise a Dedicated Traffic Channel (DTCH) which is Point-to-point bi-directional channel, dedicated to one UE, for the transfer of user information. Also, a Multicast Traffic Channel (MTCH) for Point-to-multipoint DL channel for transmitting traffic data.

[0064] Transport Channels are classified into DL and UL. DL Transport Channels comprises a Broadcast Channel (BCH), Downlink Shared Data Channel (DL-SDCH) and a Paging Channel (PCH), the PCH for support of UE power saving (DRX cycle is indicated by the network to the UE), broadcasted over entire cell and mapped to PHY resources which can be used for other control/traffic channels. The UL

Transport Channels comprises a Random Access Channel (RACH), a Request Channel (REQCH), an Uplink Shared Data Channel (UL-SDCH) and plurality of PHY channels. The PHY channels comprise a set of DL channels and UL channels.

[0065] The DL PHY channels comprises: Common Pilot Channel (CPICH), Synchronization Channel (SCH), Common Control Channel (CCCH), Shared DL Control Channel (SDCCH), Multicast Control Channel (MCCH), Shared UL Assignment Channel (SUACH), Acknowledgement Channel (ACKCH), DL Physical Shared Data Channel (DL-PSDCH), UL Power Control Channel (UPCCH), Paging Indicator Channel (PICH), and Load Indicator Channel (LICH), for example.

[0066] The UL PHY Channels comprises : Physical Random Access Channel (PRACH), Channel Quality Indicator Channel (CQICH), Acknowledgement Channel (ACKCH), Antenna Subset Indicator Channel (ASICH), Shared Request Channel (SREQCH), UL Physical Shared Data Channel (UL-PSDCH), and Broadband Pilot Channel (BPICH), for example.

[0067] Other terms/components include: 3G 3rd Generation, 3GPP 3rd Generation Partnership Project, ACLR Adjacent channel leakage ratio, ACPR Adjacent channel power ratio, ACS Adjacent channel selectivity, ADS Advanced Design System, AMC Adaptive modulation and coding, A-MPR Additional maximum power reduction, ARQ Automatic repeat request, BCCH Broadcast control channel, BTS Base transceiver station, CDD Cyclic delay diversity, CCDF Complementary cumulative distribution function, CDMA Code division multiple access, CFI Control format indicator, Co-MIMO Cooperative MIMO, CP Cyclic prefix, CPICH Common pilot channel, CPRI Common public radio interface, CQI Channel quality indicator, CRC Cyclic redundancy check, DCI Downlink control indicator, DFT Discrete Fourier transform, DFT-SOFDM Discrete Fourier transform spread OFDM, DL Downlink (base station to subscriber transmission), DL-SCH Downlink shared channel, D-PHY 500 Mbps physical layer, DSP Digital signal processing, DT Development toolset, DVSA Digital vector signal analysis, EDA Electronic design automation, E-DCH Enhanced dedicated channel, E-UTRAN Evolved UMTS terrestrial radio access network, eMBMS Evolved multimedia broadcast multicast service, eNB Evolved Node B, EPC Evolved packet core, EPRE Energy per resource element, ETSI European Telecommunications Standards Institute, E-UTRA Evolved UTRA, E-UTRAN Evolved UTRAN, EVM Error vector magnitude, and FDD Frequency division duplex.

[0068] Still yet other terms include FFT Fast Fourier transform, FRC Fixed reference channel, FS1 Frame structure type 1, FS2 Frame structure type 2, GSM Global system for mobile communication, HARQ Hybrid automatic repeat request, HDL Hardware description language, HI HARQ indicator, HSDPA High speed downlink packet access, HSPA High speed packet access, HSUPA High speed uplink packet access, IFFT Inverse FFT, IOT Interoperability test, IP Internet protocol, LO Local oscillator, LTE Long term evolution, MAC Medium access control, MBMS Multimedia broadcast multicast service, MBSFN Multicast/broadcast over single-frequency network, MCH Multicast channel, MIMO Multiple input multiple output, MISO Multiple input single output, MME Mobility management entity, MOP Maximum output power, MPR Maximum power reduction, MU-MIMO Multiple user MIMO, NAS Non-access stratum, OBSAI Open base station architecture interface, OFDM Orthogonal frequency division multiplexing, OFDMA Orthogonal frequency division multiple access, PAPR Peak-to-average power ratio, PAR Peak-to-average ratio, PBCH Physical broadcast channel, P-CCPCH Primary common control physical channel, PCFICH Physical control format indicator channel, PCH Paging channel, PDCCH Physical downlink control channel, PDCP Packet data convergence protocol, PDSCH Physical downlink shared channel, PHICH Physical hybrid ARQ indicator channel, PHY Physical layer, PRACH Physical random access channel, PMCH Physical multicast channel, PMI Pre-coding matrix indicator, P-SCH Primary synchronization signal, PUCCH Physical uplink control channel, and PUSCH Physical uplink shared channel.

[0069] Other terms include QAM Quadrature amplitude modulation, QPSK Quadrature phase shift keying, RACH Random access channel, RAT Radio access technology, RB Resource block, RF Radio frequency, RFDE RF design environment, RLC Radio link control, RMC Reference measurement channel, RNC Radio network controller, RRC Radio resource control, RRM Radio resource management, RS Reference signal, RSCP Received signal code power, RSRP Reference signal received power, RSRQ Reference signal received quality, RSSI Received signal strength indicator, SAE System architecture evolution, SAP Service access point, SC-FDMA Single carrier frequency division multiple access, SFBC Space-frequency block coding, S-GW Serving gateway, SIMO Single input multiple output, SISO Single input single output, SNR Signal-to-noise ratio, SRS Sounding reference signal, S-SCH Secondary

synchronization signal, SU-MIMO Single user MIMO, TDD Time division duplex, TDMA Time division multiple access, TR Technical report, TrCH Transport channel, TS Technical specification, TTA Telecommunications Technology Association, TTI Transmission time interval, UCI Uplink control indicator, UE User equipment, UL Uplink (subscriber to base station transmission), UL-SCH Uplink shared channel, UMB Ultra-mobile broadband, UMTS Universal mobile telecommunications system, UTRA Universal terrestrial radio access, UTRAN Universal terrestrial radio access network, VSA Vector signal analyzer, W-CDMA Wideband code division multiple access

[0070] It is noted that various aspects are described herein in connection with a terminal. A terminal can also be referred to as a system, a user device, a subscriber unit, subscriber station, mobile station, mobile device, remote station, remote terminal, access terminal, user terminal, user agent, or user equipment. A user device can be a cellular telephone, a cordless telephone, a Session Initiation Protocol (SIP) phone, a wireless local loop (WLL) station, a PDA, a handheld device having wireless connection capability, a module within a terminal, a card that can be attached to or integrated within a host device (*e.g.*, a PCMCIA card) or other processing device connected to a wireless modem.

[0071] Moreover, aspects of the claimed subject matter may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer or computing components to implement various aspects of the claimed subject matter. The term "article of manufacture" as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media. For example, computer readable media can include but are not limited to magnetic storage devices (*e.g.*, hard disk, floppy disk, magnetic strips...), optical disks (*e.g.*, compact disk (CD), digital versatile disk (DVD)...), smart cards, and flash memory devices (*e.g.*, card, stick, key drive...). Additionally it should be appreciated that a carrier wave can be employed to carry computer-readable electronic data such as those used in transmitting and receiving voice mail or in accessing a network such as a cellular network. Of course, those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope or spirit of what is described herein.

[0072] As used in this application, the terms “component,” “module,” “system,” “protocol,” and the like are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

[0073] What has been described above includes examples of one or more embodiments. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the aforementioned embodiments, but one of ordinary skill in the art may recognize that many further combinations and permutations of various embodiments are possible. Accordingly, the described embodiments are intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.

CLAIMS

What is claimed is:

1. A wireless communications method, comprising:
employing a processor executing computer executable instructions stored on a computer readable storage medium to implement the following acts:
generating cyclic shifts for a sequence generator by masking shift register output values with one or more vectors; and
forwarding the sequence generator to a future state based in part on the output values and the vectors.
2. The method of claim 1, the vectors are associated with a multi-bit polynomial.
3. The method of claim 1, further comprising generating one or more m-parameters for the sequence generator.
4. The method of claim 3, the m-parameters are combined as a set to form a Gold sequence.
5. The method of claim 4, the Gold sequence is formed by an exclusive OR operation of at least two sets of the m-parameters.
6. The method of claim 1, further comprising generating the cyclic shift via a modulo-2 adder.
7. The method of claim 6, generating an additional polynomial value via the modulo-2 adder.
8. The method of claim 1, further comprising applying differing mask values to at least two sets of m-sequences.

9. The method of claim 1, further comprising generating one or more sequence hopping functions.
10. The method of claim 1, further comprising generating one or more cell-specific sequence hopping functions.
11. The method of claim 1, further comprising generating one or more resource specific sequence hopping functions.
12. The method of claim 1, further comprising generating one or more orthogonal cover hopping functions.
13. The method of claim 1, further comprising automatically enabling or disabling sequence index hopping.
14. A communications apparatus, comprising:
 - a memory that retains instructions for generating cyclic shifts for a sequence generator by combining shift register output values with one or more vectors and starting the sequence generator at a future state based in part on the output values and the vectors; and
 - a processor that executes the instructions.
15. The apparatus of claim 14, further comprising one or more m-parameters that are employed for the sequence generator.
16. The apparatus of claim 15, the m-parameters are combined as a set to form a Gold sequence.
17. The apparatus of claim 16, the Gold sequence is formed by an exclusive OR operation of at least two sets of the m-parameters.
18. The apparatus of claim 14, further comprising a modulo-2 adder to generate the cyclic shifts.

19. A communications apparatus, comprising:
 - means for shifting register output values with one or more vectors;
 - means for generating one or more m-parameters in accordance with the vectors;
 - and
 - means for setting a future state based in part on the output values, the m-parameters, and the vectors.
20. The apparatus of claim 19, the m-parameters are associated with a set of sequences that are employed to form a Gold sequence.
21. A computer-readable medium, comprising:
 - masking register output values with one or more vectors;
 - adding one or more m-parameters in accordance with the vectors; and
 - setting a sequence generator to future state based in part on the output values, the m-parameters, and the vectors.
22. The computer-readable medium of claim 21, employing an adder to perform a cyclic shift operation.
23. The computer-readable medium of claim 21, further comprising configuring the sequence generator via a polynomial value.
24. A processor that executes the following instructions:
 - adjusting register output values according to one or more vectors;
 - sequencing one or more m-parameters with the vectors; and
 - clocking a sequence generator to future state based in part on the output values, the m-parameters, and the vectors.
25. The processor of claim 24, further comprising generating a Gold sequence from at least two m-sequences.

26. A wireless communication method, comprising:
generating a first masking vector from a sequence generator polynomial and a first cyclic shift;
generating a second masking vector from the sequence generator polynomial and a second cyclic shift; and
employing the first and second masking vector to mask shift register output values to obtain a first output value and a second output value, the first output value and the second output value employed to generate a future sequence state.
27. The method of claim 26, further comprising adding results of the first output value to create a first random sequence.
28. The method of claim 26, further comprising adding results of the second output value to create a second random sequence.
29. An apparatus operable in wireless communication system, comprising:
means for generating a first masking vector from a sequence generator polynomial and a first cyclic shift;
means for generating a second masking vector from the sequence generator polynomial and a second cyclic shift; and
means for processing the first and second masking vector to mask shift register output values to obtain a first output value and a second output value that are employed to generate a Gold sequence.
30. The apparatus of claim 29, further comprising a component to add results of the first output value to create a first random sequence and for adding results of the second output value to create a second random sequence.
31. A computer-readable medium comprising instructions which, when executed by a computer, cause the computer to perform operations including:
generating a first masking vector from a sequence generator polynomial and a first cyclic shift;

generating a second masking vector from the sequence generator polynomial and a second cyclic shift;

employing the first and second masking vector to mask shift register output values to obtain a first output value and a second output value;

adding components of the first output value to create a first random sequence; and

adding components of the second output values to create a second random sequence.

32. The computer-readable medium of claim 31, further comprising generating a Gold sequence from at least two m-sequences.

33. An apparatus operable in a wireless communication system, comprising:
a processor, configured for:

generating a first masking vector from a sequence generator polynomial and a first cyclic shift;

generating a second masking vector from the sequence generator polynomial and a second cyclic shift;

shifting the first and second masking vector with shift register output values to obtain a first output value and a second output value; and

generating at least one pseudo-random sequence from the masking vector and the shift register output values.

34. The apparatus of claim 33, further comprising processing at least one m-sequence value in accordance with the first and second masking vector.

35. The apparatus of claim 34, further comprising generating at least one Gold sequence value.

36. A communications method, comprising:

employing a processor executing computer executable instructions stored on a computer readable storage medium to implement the following acts:

generating cyclic shifts for a sequence generator; and

forwarding pseudorandom sequences within the sequence generator by setting a shift register stage to a desired future state.

37. The method of claim 36, the future state depends on a generating polynomial, a number of steps of forwarding required, or an initial state.

38. The method of claim 36, further comprising using one or more m-parameters to generate a consecutive sequence of bits occurring before the desired future state and employing the bits and to initialize the shift register stage.