PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : H04L 9/00 — A1

(11) International Publication Number: WO 95/26086

(43) International Publication Date: 28 September 1995 (28.09.95)

(21) International Application Number: PCT/US94/02942

(22) International Filing Date: 18 March 1994 (18.03.94)

(71) Applicant *(for all designated States except US)*: THOMSON CONSUMER ELECTRONICS, INC. [US/US]; 10330 North Meridian Street, Indianapolis, IN 46290-1024 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: MURPHY, Peter, Michael [US/US]; 235 Haldale Drive, Carmel, IN 46032 (US).

(74) Agents: TRIPOLI, Joseph, S. et al.; GE & RCA Licensing Management Operation, Inc., CN 5312, Princeton, NJ 08540 (US).

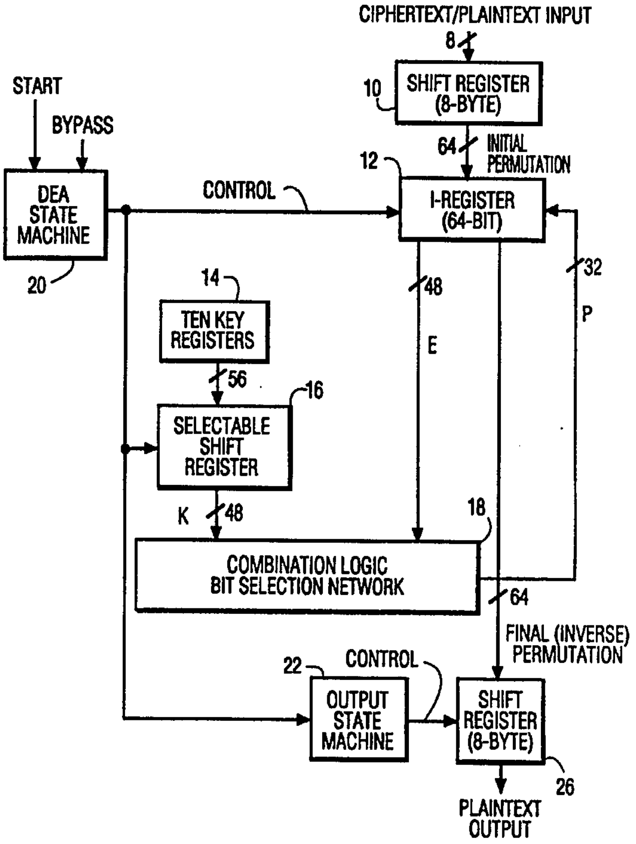
(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, LV, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
With amended claims and statement.

(54) Title: DATA DECRYPTION APPARATUS IN A SUBSCRIPTION TELEVISION SIGNAL RECEIVING SYSTEM

(57) Abstract

A receiver of satellite-broadcast signals including high definition television signals includes apparatus (12-26) for decrypting encrypted signals. When unencrypted "plaintext" signal information which is not to be decrypted is received, the plaintext information is applied to the decrypting apparatus. Normal decrypting operation is modified so that the input plaintext information appears unaltered as plaintext information at the output of the decrypting apparatus. A bit selection network (628; S1, ... S8) associated with a decipher function (f(R,K)) of the decryption apparatus employs a combinational logic network (18, 628) rather than Look up Tables.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SI	Slovenia
SK	Slovakia
SN	Senegal
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

Data Decryption Apparatus In a Subscription Television Signal Receiving System

Background of the Invention

This invention is related to the field of digital video signal processing, and more particularly to apparatus for decrypting a television signal received from a transmission channel such as a broadcast satellite, for example.

Subscriber video and television services, such as cable and satellite broadcast/receiving systems, often encrypt various broadcast services, such as video and audio, by means of a control word or "key" to limit access to only paid subscribers. The encrypting process at a transmitter may be in accordance with various techniques. Some of these techniques are described in "American National Standard Data Encryption Algorithm" ANSI X3.92-1981, and in "American National Standard for Information Systems-Data Encryption Algorithm-Modes of Operation" ANSI X3.106-1983.

Summary of the Invention

In a receiver of a signal subject to containing encrypted information, a bit selection processor associated with a decipher function of a receiver decryption processor exhibits enhanced operating speed by employing a combinational logic network rather than Look Up Tables, for example.

Brief Description of the Drawings

In the drawings:

Figure 1 is a block diagram of decrypting apparatus according to the principles of the present invention.

Figure 2 is a block diagram of a satellite video signal broadcasting and receiving system including the decrypting apparatus of Figure 1 in a transport unit of Figure 2.

Figures 3-13 depict block diagrams and tables which are helpful in understanding the decrypting operation of the apparatus shown in Figure 1.

Figures 14 and 15 respectively show a state diagram and an associated logic table for an input main state machine in the system of Figure 1.

Figures 16 and 17 respectively show a state diagram and an associated logic table for an output state machine in the system of Figure 1.

Detailed Description

The decryption apparatus shown in Figure 1 operates in accordance with the Data Encryption/Decryption Algorithm (DEA), which is a standard algorithm for encrypting and decrypting digital data as discussed in "American National Standard Data Encryption Algorithm" ANSI X3.92-1981. In particular, the Figure 1 apparatus operates in an "electronic codebook mode" as described in "American National Standard for Information Systems-Data Encryption Algorithm-Modes of Operation" ANSI X3.106-1983. In this example it is assumed that a received input datastream contains data from five sources or "services" associated with a subscription television system, and that the decryption apparatus provides ten decryption keys, two for each service. The use of two alternate keys for each service advantageously allows a given key, which is not currently in use, to be modified periodically without disrupting current decryption processing using the other key. According to the DEA standard, each key is a 64 bit word comprising 56 decryption bits and 8 parity check bits. In the disclosed system, the 8 parity bits are not used.

An input signal may comprise either encrypted ciphertext data, or unencrypted plaintext data. In accordance with the DEA standard, "plaintext" is intelligible text or signals that have meaning and that can be read and used, and decryption is the process of transforming ciphertext into plaintext by means of a standard algorithm. The input signal is in the form of a parallel 8 byte datastream of 8 bits/byte. An input 8 byte shift register 10 acquires input data one byte at a time in response to a clock. The clock signals are not shown to simplify the drawing. Thus 8 bits (1 byte) are clocked into register 10 during each clock cycle, whereby 8 clock cycles are required to fully load register 10 with 64 bits. Elements 12, 14, 16 and 18 form a decryption processor. The loading of input register 10 is accomplished while the decryption processor is performing the data decryption algorithm on 64 bits of data from a previous input cycle. Since the standard data decryption algorithm (as will be discussed) is composed of 16 iterations, and 8 clock cycles are required to load register 10, the clocking speed of decryption processor elements 12, 14, 16 and 18 is twice as fast as that of input register 10, ie., twice as fast as the speed of acquiring the next 8 bytes of input data.

The decrypting operation is under the control of a state machine 20, eg., a microcontroller. State machine 20 may be programmed to cause the system to wait between decrypting data while the rest of the next 64 bits of input data is acquired, or it may be programmed to cause the system to proceed directly to decrypting new data after completing the decryption of the previous data. State machine 20 responds to input control signals Start and Bypass, and produces an output Control signal.

The Bypass signal signifies that the normal decryption operation is to be modified so that input unscrambled plaintext data remains in the same form at an output of the decryption network, ie., the decryption process is to be bypassed. An output Control signal from state machine 20 conveys this bypass instruction to decryption processing elements 12 and 16, and to output state machine 22. To bypass the decryption operation for a plaintext data input, state machine 20 causes "I" register 12 to be loaded from input register 10, then register 12 waits ("idles") for an interval encompassing the 16 decryption iterations (as will be discussed in connection with Figure 3) without performing the iterations, after which the contents of register 12 are passed to output register 26. Since a final data permutation (inverse permutation) at the input of output register 26 is predetermined to be the inverse of the initial permutation at the input of register 12, the plaintext output data from register 26 is identical to the input plaintext data. Thus the decryption bypass function is advantageously accomplished without resorting to the circuit and interface complexities which would be associated with the use of switching networks for switching the plaintext data around the decryption processing network.

An input ciphertext/plaintext digital signal received and processed by the disclosed system is in the form of data packets including a data component which is subject to encryption, and an associated header component which contains data-identifying information. The header is not encrypted. The input Bypass control signal to unit 20 is produced in reponse to a control bit included in the header of a received data packet that is sensed by an input signal processing network (not shown). The Control signal from unit 20 contains information that instructs register 12 to idle for sixteen iterations in the plaintext bypass mode, as noted above. The Control signal from unit 20 also contains information that instructs a multiplexer in register 12 to select either the output of unit 10 or the output of unit 18 for processing. The Start signal input to unit 20 is generated by an input signal processing network preceding unit 20 (not shown) that senses the header of a received 8 byte data packet and, after eight bytes are sensed (indicating that input register 10 is full), generates information on the Control signal line that causes the output of input register 10 to be loaded into register 12. This information also informs output state machine 22 that another block of data has entered the data processing pipeline, which data block will be output from shift register 26 under control of unit 22 after the sixteen-step iterative decryption process has been completed as will be discussed.

Figures 14 and 15 respectively show a state diagram and an associated logic table for state machine 20. When the Start signal is received, unit 20 first initializes the decryption system by loading 64 bit input data from unit 10 to unit 12. Unit 20 then counts the 16 iterations of the decryption, after which unit 20 waits for the next Start signal. In anticipation of the last Start signal still being present at the end of the decryption process, unit 20 goes to state "Done-Wait" and waits for the Start signal to disappear, after which unit 20 waits for the next Start signal. The 13 states associated with the 5-bit data words between 10001 and 11111 are not used and do not occur, since only 19 states are needed as shown in this example.

Figures 16 and 17 respectively show a state diagram and an associated logic table for output state machine 22. When unit 22 receives a Done (Not done) control signal from state machine 20, that signal indicates that data is being decrypted or is being passed through the apparatus in the bypass mode. Unit 22 then advances to a Ready state. When state machine 20 signals that the decryption process is Done, output state machine 22 counts out the 8 bytes (states S1 through S8). Unit 22 then waits for the $\overline{\text{Done}}$ then Done signal sequence. If a $\overline{\text{Done}}$ signal is received while the 8 bytes are being distributed, unit 22 keeps track of this by finishing the count with state R1 through R8. If at the end of the count the decryption is still not done, unit 22 goes to the Ready state and waits until the decryption process is completed. If a Done signal is received while states R1 through R8 are being used, then at the end of the count state machine 22 goes directly to distributing the next 8 bytes.

The codeword permutations defined by the DEA standard are implemented by appropriately arranging the data bits on the data links interconnecting units 12, 14, 16 and 18 in Figure 1. A decipher bit selection function performed by unit 18 is advantageously implemented by a combinational logic network. Specifically, the bit selection function may be implemented by means of a 6-input, 4-output combinational logic arrangement as will be explained subsequently. The use of combinational logic for performing the bit selection function is considered to result in economy of hardware and faster operation of the selection function compared to the use of a ROM-based Look-Up table, for example.

Output state machine 22 allows the plaintext output to be conveyed to the output channel via register 26 one byte at a time while the next 64 bit sequence is being processed by units 12-18, and while another 64 bit sequence is being acquired by input register 10. Output state machine 22 may be programmed to either wait after the last byte of a given sequence has been conveyed by output register 26 (since the next 64 bits may not have been completely decrypted), or state machine 22 may immediately begin distributing the next 64 bit plaintext output sequence. Thus state machine 22 and output register 26 may wait in the case of data occurring at a slow or non-uniform rate, or unit 22 may cause data blocks to be shifted out of register 26 as they appear in the case of data occurring at a maximum rate. Output state machine 22 determines whether or not new data is being processed by the decryption network, if new data is being processed but is not completely decrypted, or if new data is decrypted and is waiting to be distributed to the output channel via register 26. This is determined in response to, for example, Control signal information from unit 20 indicating that a given block has been decrypted after sixteen iterations.

In the "bypass" mode, when unencrypted plaintext input information is present, the input plaintext information is conveyed from input register 10 to output register 26 via decryption processor units 12 and 18, which exhibit modified operation in this mode. Specifically, the output of unit 18, corresponding to block 622 in Figure 6, is not used during the 16 iteration cycle over which decryption is otherwise performed.

The bypass mode is facilitated by causing register 12 to idle during the bypass mode, as follows. When a 64 bit input has been received in the bypass mode, the decryption processor and state machine 20 commence operation in substantially the same way as during a decryption operating mode. Data is transfered from shift register 10 to register 12 with a permutation of bit positions as described previously. In each of the 16 iterations of a decryption, the right half of register 12 gets the previous left half of register 12, and the left half of register 12 gets the output of combinational logic circuit 18. However, there is only one iteration in the bypass mode. In this one bypass iteration, the right half of register 12 gets the previous left half of register 12, and the left half of register 12 gets the previous right half of register 12. Register 12 maintains its value until output register 26 is ready to accept data from register 12. Register 12 maintains its value by conveying output to input on each clock.

More specific information concerning the DEA Data Encryption/Decryption Algorithm performed by the apparatus of Figure 1 follows immediately below, based on the DEA publication

"American National Standard Data Encryption Algorithm," American National Standard X3.92-1981. The decryption process is essentially the inverse of the encryption process which is described in more detail below.

The data encryption/decryption algorithm is designed to encipher and decipher 64 bit data blocks, under control of a 64 bit key. Deciphering uses the same key that was used for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process.

A block to be enciphered is subjected to an initial permutation "IP", then to a complex key-dependent computation, and finally to a permutation IP^{-1} that is the inverse of the initial permutation. The key-dependent permutation may be defined in terms of a cipher function "f" and a key schedule function "KS." Descriptions of computation and enciphering operations are provided below. The following notation is convenient to an understanding of the following material. Given two blocks L and R of bits, LR denotes a block consisting of the bits of L followed by the bits of R. Since concatenation is associative, B1, B2...B8, for example, denotes a block consisting of the bits of B1 followed by the bits of B2 ... followed by the bits of B8.

The enciphering computation is illustrated by Figure 3. The 64 bits of an input block to be enciphered are first subjected to initial permutation IP, as given in table 1 of Figure 4, before being received by register 12 in Figure 1. The permuted input has bit 58 of the input as its first bit, bit 50 as its second bit, and so on, with bit 7 as the last bit. The permuted input block is then input to a complex, key-dependent computation described by the equations that follow below. The output of that computation, called the "preoutput", is then subjected to the permutation given in table 2 of Figure 5, which is the inverse of the of the initial permutation. Thus, the output of the algorithm has bit 40 of the preoutput block as its first bit, bit 8 as its second bit, and so on, until bit 25 of the preoutput block is the last bit of the output. The initial permutation performed at the input of register 12 may be accomplished by rearranging the wiring that connects the output of unit 10 to the input of unit 12. Alternatively, this could be accomplished by using a logic network. The inverse permutation performed at the input of output register 26 may be accomplished similarly.

The computation uses the permuted input block as input to produce the preoutput block. Except for a final interchange of blocks, the computation consists of 16 iterations of a set of operations including calculation of cipher function f, which operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits. For example, let the 64 bits of the input block to an iteration consist of a 32 bit block L followed by a 32 bit block R, so that the input block is designated as LR. If K is a block of 48 bits chosen from the 64 bit key, then the output L'R' of an iteration with input LR is defined by

$$L' = R$$
$$R' = L + f(R,K) \qquad \text{(Equation 1)}$$

where in this example "+" denotes bit-by-bit modulo 2 addition.

As stated earlier, the input of the first iteration of the calculation is the permuted input block. If L'R' is the output of the sixteenth iteration, then R'L' is the preoutput block. At each iteration a different block K of key bits is selected from the 64 bit key designated as KEY. This is accomplished by selectable 56 bit shift register 16 in response to the Control signal from unit 20. Specifically, shift register 16 shifts the 56 bits of the then active key one or two places for each iteration, as a function of a predetermined key schedule, and 48 bits are selected each time as indicated by function "K" at the output of shift register 16. In this example the 48 bits are selected by appropriately configuring the wiring bus between units 16 and 18. An example of such a key schedule is described in "American National Standard Data

Encryption Algorithm" ANSI X3.92-1981 mentioned previously. The iterations of the computation can now be described in more detail. Let KS be a function that takes an integer "n" in the range from 1 to 16 and a 64 bit block KEY as input and yields as output a 48 bit block K_n, which is a permuted selection of bits from KEY.

$$K_n = KS(n,KEY) \qquad \text{(Equation 2)}$$

K_n is determined by the bits in 48 distinct bit positions of KEY. KS is called the key schedule because the block K used in the nth iteration of Equation 1 above is the block K_n determined by Equation 2. As before, let the permuted input block be LR. Let L_0 and R_0 be respectively L and R, and let L_n and R_n be respectively L' and R' of equation 1, when L and R are respectively L_{n-1} and R_{n-1} and K is K_n. That is, when n is in the range 1 to 16

$$L_n = R_{n-1}$$

$$R_n = L_{n-1} + f(R_{n-1},K_n) \qquad \text{(Equation 3)}$$

The preoutput block is then $R_{16}L_{16}$. The key schedule KS produces the 16 values of Kn that are required for the algorithm, as discussed in greater detail in the DEA publication "American National Standard Data Encryption Algorithm," American National Standard X3.92-1981.

During the deciphering operation performed by the apparatus of Figure 1, the inverse permutation IP^{-1} (the final permutation in Figure 1) which is applied to the preout block is the inverse of the initial permutation IP applied to the input. From Equation 1 it follows that

$$R = L'$$

$$L = R' + f(L',K) \qquad \text{(Equation 4)}$$

Consequently, to decipher it is only necessary to apply the same algorithm to an enciphered message block, taking care that at each iteration of the computation the same block of key bits K is used during deciphering as was used during the enciphering of the block. This concept can be expressed as follows:

$R_{n-1} = L_n$

$L_{n-1} = R_n + f(L_n,K_n)$ (Equation 5)

where now $R_{16}L_{16}$ is the permuted input block for the deciphering calculation and L_0R_0 is the preoutput block. That is, for the deciphering calculation with R_{16} L_{16} as the permuted input, K_{16} is used in the first iteration, K_{15} in the second, and so on, with K_1 used in the sixteenth iteration. In connection with the above it is noted that permutations performed at the transmitter/encryptor are the inverse of permutation performed at the receiver/decryptor. Thus initial permutation (IP) at the decryptor in Figure 1 has a corresponding inverse permutation performed at the encryptor.

The sixteen step iteration process (Figure 3) involves the calculation of 16 key-dependent cipher functions f(R,K) performed by combinational logic bit selection network 18 in Figure 1. It should be understood that f(R,K) are actually decipher functions in the context of the decryption apparatus of Figure 1. The decipher functions are the inverse of the cipher functions performed at the transmitter/encoder. Figure 6 shows additional details of network 18. Referring to Figure 6, each calculation is performed with respect to a 32 bit block "R" designated as 610, and a 48 bit block "K" designated as 616. Block R is one-half of an input 64 bit data block, and K is a block of 48 bits chosen from a 64 bit key. Block R is expanded to 48 bits (block 614) by function "E" performed by unit 612, to be compatible with the length of block K when blocks R and K are combined by unit 626. As noted previously, for each iteration a different block K of 48 key bits is chosen (permuted) from the (shifted) 64 bit key according to a predetermined schedule.

A combinational logic bit selection network 628 includes a plurality of unique selection functions S1, ... S8 which essentially form the basis of the cipher/decipher function. Each selection function S1, ... S8 produces a unique combination of 4 output bits in response to 6 input bits received from an exclusive-OR logic network 626. That is, each of the selection functions substitutes one set of original bits for another set of bits. The 6 bit to 4 bit substitution is in accordance with the DEA standard. The original bits for which the substitution is made are either plaintext bits or encrypted bits, depending on whether the operation is being performed at a transmitter/encryptor or a receiver/decryptor.

More specifically, in Figure 6, block 610 represents an input data block to unit 12 of Figure 1, and element E represents an expansion function performed within unit 12 in Figure 1. Block 614 represents a 48 bit output block from unit 12 in Figure 1. Block 616 represents an output data block from unit 16 of Figure 1 as applied to an input of unit 18 in Figure 1. Network 626, processor 628 and a permutation function P indicated by element 620 are included within unit 18 in Figure 1, which produces an output 32 bit data block designated by 622 in Figure 6. Elements 612, 614, 620 and 628 of Figure 6 perform the cipher function "f" shown in Figure 3. The elements of Figure 6, particularly elements 612, 614, 626, 628 and 620, may be employed in both the encryption and decryption processes.

In Figure 6, element E denotes an expansion function that receives an input block of 32 bits and produces an output block of 48 bits. Function E is such that the 48 output bits, written as 8 blocks of six bits each, are obtained by selecting the input bits in the order indicated by table 3 in Figure 7. Thus, the first three bits of E(R) are the bits in positions 32, 1, and 2 of R, and the last two bits of E(R) are the bits in positions 32 and 1. Each of the unique bit selection functions S1, S2, ...S8 receives a 6 bit input block and produces a 4 bit output block. This process is illustrated by table 4 of Figure 8, which contains values for function S1. If S1 is the function defined by table 4, and B is a block of 6 bits, then S1(B) is determined as follows. The first and last bits of B represent, in base 2 binary form, a number in the range 0 to 3.

Let that number be "i." The middle four bits of B represent, in base 2, a number in the range 0 to 15. Let that number be "j." In table 4, the number in the ith row and the jth column is a number in the range 0 to 15 and is uniquely represented by a 4 bit block. That block is the output of S1(B) of S1 for input B. For example, for binary input 011011 the row is binary 01 (ie., row 1) and the column is determined by binary 1101 (ie., column 13). In row 1, column 13, the number 5 appears, so the binary output is 0101. The complete set of selection functions S1, S2, ... S8 is shown in table 6 in Figure 9.

Table 4, which defines selection function S1, may be used as shown, ie., as shown in the specification of the DEA standard noted above. However, in the illustrated Figure 1 system, table 4 in Figure 8 was rearranged as shown in Figure 11 to facilitate the use of a combinational logic network rather than Look-Up tables. Specifically, as shown by the table in Figure 11, the table was rearranged so that 6 bit input B represents (in base 2) a number in the range 0 to 63, without altering the order of the bits in B. In the table of Figure 11, the "Output" represents the quantity S1(B) discussed above.

The table of Figure 11 was further arranged, as indicated by the table of Figure 12, so that the unique 4 bit ouputs (representing numbers in the range 0-16) could be used to determine the four possible 6 bit B inputs (representing numbers in the range 0-63) that produce the output. That is, the table of Figure 12 represents the relationship between a 4 bit output and possible 6 bit B inputs. Finally, a Boolean algebra expression was created that describes the function represented by the table shown in Figure 12. This Boolean expression is used to synthesize a combinational logic circuit, using conventional logic circuit design techniques, for the selection function indicated by the table of Figure 12. Figure 13 shows the VHDL code for implementing the Boolean expression of the table in Figure 12. Techniques similar to that described above for the tables shown in Figures 8 and 11-13 for selection function S1 are used to create Boolean expressions for synthesizing combinational logic circuits for each of the other bit selection functions S2, ... S8, which selection functions are shown in table 6 of Figure 9. The described re-arranging of decryption tables could also be done at a transmitter/encryptor, but it is not necessary to do so.

The permutation function P yields a 32 bit output block from a 32 bit input block by permuting the bits of the input block. The permutation function is defined by table 5 in Figure 10. The output P(L) for the function P defined by this table is obtained from the input L by taking the 16th bit of L as the first bit of P(L), the seventh bit of L as the second bit of P(L), and so on until the 25th bit of L is taken as the 32nd bit of P(L). Assuming S1, ... S8 are eight distinct selection functions, P is the permutation function function, and E is the expansion function. To define f(R,K), blocks B1, ... B8 are defined as blocks of 6 bits each for which

$$B1B2 \ldots B8 = K + E(R). \qquad \text{(Equation 6)}$$

The block f(R,K) is then defined to be

$$P(S1(B1)S2(B2) \ldots S8(B8)). \qquad \text{(Equation 7)}$$

Thus, K+E(R) is first divided into the eight blocks as indicated in Equation 6. Then each Bi is taken as an input to Si, and the 8 blocks S1(B1),S2(B2), ... S8(B8) of 4 bits each are consolidated into a single block of 32 bits that forms the input to P. The output (shown in Equation 7) is then the output of the function f for inputs R and K.

In the satellite transmitter/receiver system of Figure 2, a transmitter section processes signals from a source 30. In this example source 30 includes a plurality of audio and video sources, eg., including television signal sources which contain information in the form of transport packets including a payload data component and an associated header component which describes the contents of an associated data component. The data packets from the respective sources are subjected to asynchronous time division multiplexing (ATDM) onto an output path before being processed by units 32 and 34.

Signals from source 30 are encoded and compressed by a unit 32, which in this example includes MPEG coding apparatus. MPEG is an international standard developed by the Moving Pictures Expert Group of the International Standards Organization for coded representation of moving pictures and associated audio stored on a digital storage medium. Encoded signals from unit 32 are provided to a Quaternary Phase Shift Keyed (QPSK) modulator and FEC (Forward Error Corrector) 34, which encodes the signal with error correction data and QPSK modulates the encoded signal onto a carrier. Unit 34 performs both convolutional and Reed Solomon (RS) coding. Uplink unit 36 transmits the compressed and encoded signal to satellite 40, which broadcasts the signal to a selected geographic reception area. In this example satellite 40 operates in two modes, which trade off channel capacity and transmission power. In one mode, satellite 40 transmits 16 channels at 120 watts each, and in other mode transmits 8 channels at 240 watts each.

The signal from satellite 40 is received by an antenna (not shown) and is coupled to an input tuner circuit 44 of a receiver. An output signal from tuner 44 is QPSK demodulated by unit 46, decoded by means of units 48, 50 and 52, and applied to a transport processor 56. A QPSK demodulator suitable for use as unit 46 is commercially available from Hughes Network Systems of Germantown, MD (integrated circuit type No. 1016212), and from Comstream Corp., San Diego, California (No. CD 2000). Transport processor 56 transports a decoded output signal from unit 52 to appropriate decoders within unit 62 depending upon the content of the signal from unit 52, eg., audio or video information. Transport unit 56 receives corrected data packets from unit 52 and examines the header of each packet to determine its routing. Transport unit 56 includes the decryption apparatus shown in Figure 1. In this satelllite subscription system, unencrypted plaintext information includes header data, decryption keys, listings of available program material for each of the several sources, audio and miscellaneous items. A satellite system typically provides many more channels than broadcast or cable systems, with many more program listings which advantageously should not be decrypted.

Audio and video output signals from unit 62 are respectively applied to an audio processor 66, and to an NTSC television signal video encoder 64 which encodes the video signal to a format suitable for use by signal processing circuits in a standard NTSC consumer television receiver 68. The audio signals from unit 66 are applied to an audio input of receiver 68.

A microcontroller 60 responds to an input User Control signal, eg., from a remote control device, and operates interactively with tuner 44, demodulator 46, decoder units 48 and 50, and transport processor 56, as described in detail in a copending PCT patent application (RCA 87,182) of John S. Stewart. Briefly, microcontroller 60 provides a Frequency control signal to tuner 44 in response to a user's channel selection, causing tuner 44 to tune to the appropriate channel. QPSK demodulator 46 synchronizes with the tuned channel, provides a demodulated signal to decoder 48, and also provides a Signal Quality control signal to microcontroller 60 indicative of the quality (eg., signal to noise ratio) of a received signal. Demodulator 46 also provides a Demodulator Lock control signal to microcontroller 60 indicating whether or not demodulator 46 is synchronized with the input signal.

Decoder 48 uses a Viterbi algorithm to decode and correct bit errors in the demodulated signal from unit 46. Decoder 48 includes internal networks, as known, to synchronize its operation to the incoming demodulated signal in order to effectively decode the demodulated signal. Decoder 48 operates at one of two error correction decoding rates, which correspond to error correction coding rates provided at the transmitter. When satellite 40 operates in a low power mode, the transmitted signal uses a rate 2/3 error correction code, for example. When satellite 40 operates in a high power mode, the transmitted signal uses a rate 6/7 error correction code. A Code Rate control signal, eg., a binary signal developed by a comparator network in microcontroller 60, signifies either that the code rate used by decoder 48 should remain unchanged, or should be switched to another programmed code rate. The Code Rate control signal may instruct decoder 48 to change the code rate as a function of the Signal Quality signal which indicates a low quality received signal, or as a function of the Demodulator Lock signal which indicates that demodulator 46 is not locked to (synchronized with) the received signal, coupled with an Error signal from Reed-Solomon decoder 52 indicating the occurrence of a decoding error.

If decoder 48 is using an incorrect error correction code rate for a given input signal, it is unlikely that RS decoder 52 will provide a normal output. An Error signal from decoder 52 will be analysed with regard to the Signal Quality and Demodulator Lock signals from demodulator 46. If the latter two signals indicate that the input signal is of acceptable quality and that demodulator 46 is synchronized with the input signal, it is likely that a decoding error manifested by the Error signal is caused by the fact that decoder 48 is using a code rate different from that of the received signal, ie., the error correction code rate of the transmitted signal was changed at the transmitter. If the Signal Quality or Demodulator Lock signal indicate a poor quality received signal or lack of demodulator synchronism, the Error signal may be due to these factors (eg., caused by rain fade) rather than to an incorrect code rate being used by decoder 48. Microprocessor 60 may then wait a predetermined time before examining the control signals again.

De-interleaver 50 restores the ordering of data signal packets to an original sequence, and forms Reed-Solomon blocks in accordance with known techniques. For this purpose de-interleaver 27 relies upon an 8 bit sync word inserted by the encoder at the beginning of each RS block, thereby providing RS block synchronization. The de-interleaved signal is supplied to Reed-Solomon decoder 28.

What Is Claimed Is:

1. In a system for processing digital video signals including signals containing encrypted information, digital signal processing apparatus comprising:

input means (10) responsive to digital signals subject to containing encrypted information;

decryption means (12-18) responsive to signals containing encrypted information from said input means for providing decrypted information at an output of said decryption means; and

output means (26) for conveying signals from said output of said decryption means to an output channel; wherein

said decryption means includes decipher function means (18) employing combinational logic means (628) for implementing a plurality of bit selection functions (S1, ... S8).

2. Apparatus according to claim 1, wherein said decryption means includes

means for performing a plurality of iterative calculations each including calculation of a decipher function (f(R,K)) and a predetermined bit selection function.

3. Apparatus according to claim 2, wherein

each said cipher function is key dependent.

4. Apparatus according to claim 2, wherein

said decryption means includes a plurality of combinational logic means respectively associated with respective ones of said bit selection functions.

5. Apparatus according to claim 1, wherein

said decryption means operates in accordance with DEA standards.

6. Apparatus according to claim 1, wherein said apparatus in included in a subscription satellite broadcast system.

AMENDED CLAIMS

[received by the International Bureau on 22 August 1994 (22.08.94); original claims 1 and 6 amended; remaining claims unchanged (2 pages)]

1. In a system for processing digital video signals including signals containing encrypted information, digital signal processing apparatus comprising:

input means (10) responsive to digital signals subject to containing encrypted information;

decryption means (12-18) responsive to signals containing encrypted information from said input means for providing decrypted information at an output of said decryption means; and

output means (26) for conveying signals from said output of said decryption means to an output channel; wherein

said decryption means includes decipher function means (18) employing combinational logic means (628) for implementing a plurality of bit selection functions (S1, ... S8) that provide a predetermined first number of output bits in response to a predetermined second number of input bits.

2. Apparatus according to claim 1, wherein said decryption means includes

means for performing a plurality of iterative calculations each including calculation of a decipher function (f(R,K)) and a predetermined bit selection function.

3. Apparatus according to claim 2, wherein

each said cipher function is key dependent.

4. Apparatus according to claim 2, wherein

said decryption means includes a plurality of combinational logic means respectively associated with respective ones of said bit selection functions.

5. Apparatus according to claim 1, wherein said decryption means operates in accordance with DEA standards.

6. Apparatus according to claim 1, wherein said apparatus is included in a subscription satellite broadcast system.

STATEMENT UNDER ARTICLE 19

Claim 1 has been amended to further specify and clarify the structure and operation of the claimed decryption means, consistent with the disclosure of Figure 6 and the corresponding explanatory text in the specification (cf page 11 et seq.).

Claim 6 is amended to correct a typographical error.

Please acknowledge receipt of the above noted claim amendments.

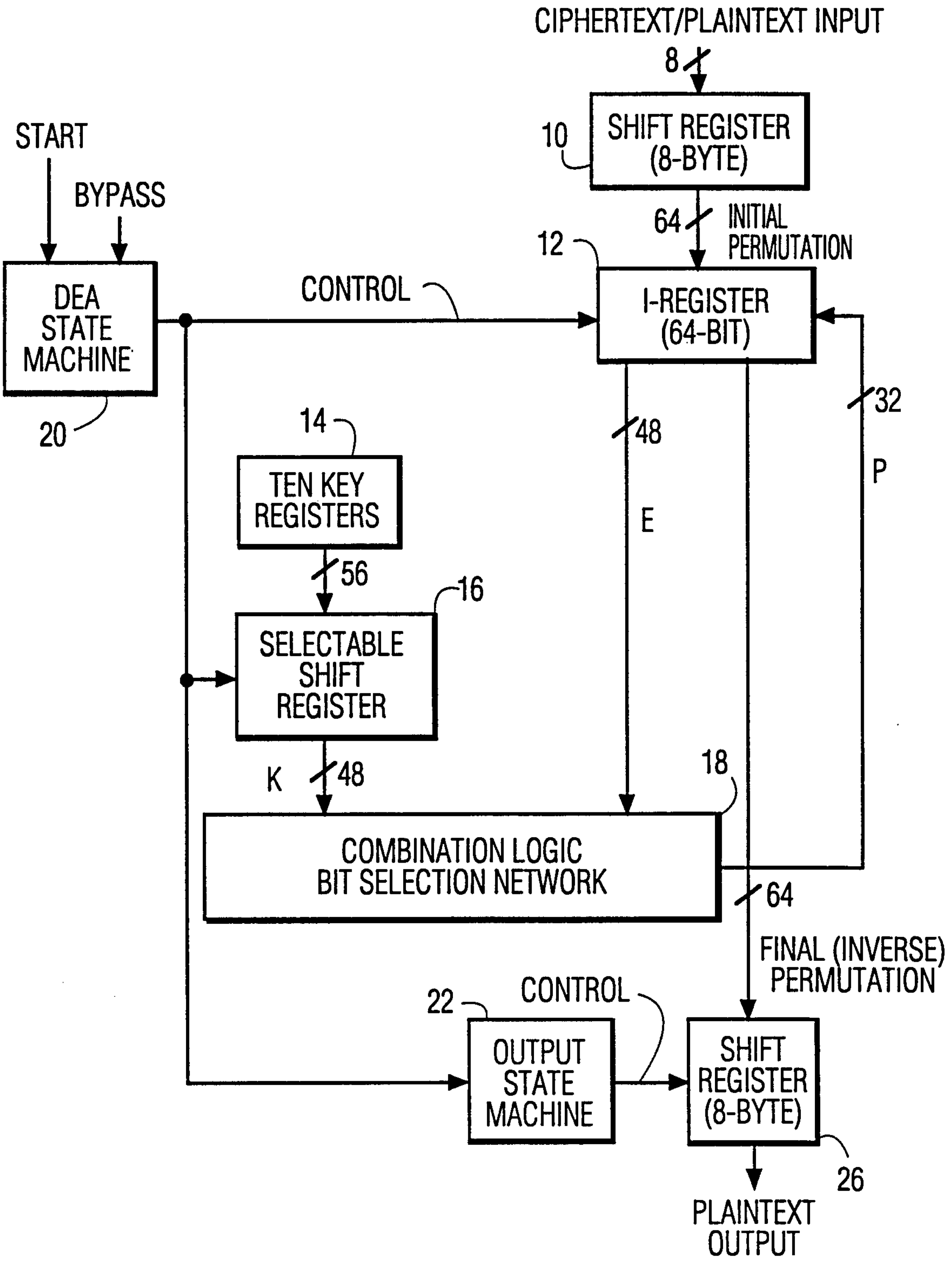

CIPHERTEXT/PLAINTEXT INPUT
8
SHIFT REGISTER (8-BYTE)
10
64
INITIAL PERMUTATION
12
START
BYPASS
DEA STATE MACHINE
20
CONTROL
I-REGISTER (64-BIT)
32
48
P
E
14
TEN KEY REGISTERS
56
16
SELECTABLE SHIFT REGISTER
K
48
18
COMBINATION LOGIC BIT SELECTION NETWORK
64
FINAL (INVERSE) PERMUTATION
22
CONTROL
OUTPUT STATE MACHINE
SHIFT REGISTER (8-BYTE)
26
PLAINTEXT OUTPUT

FIG.1

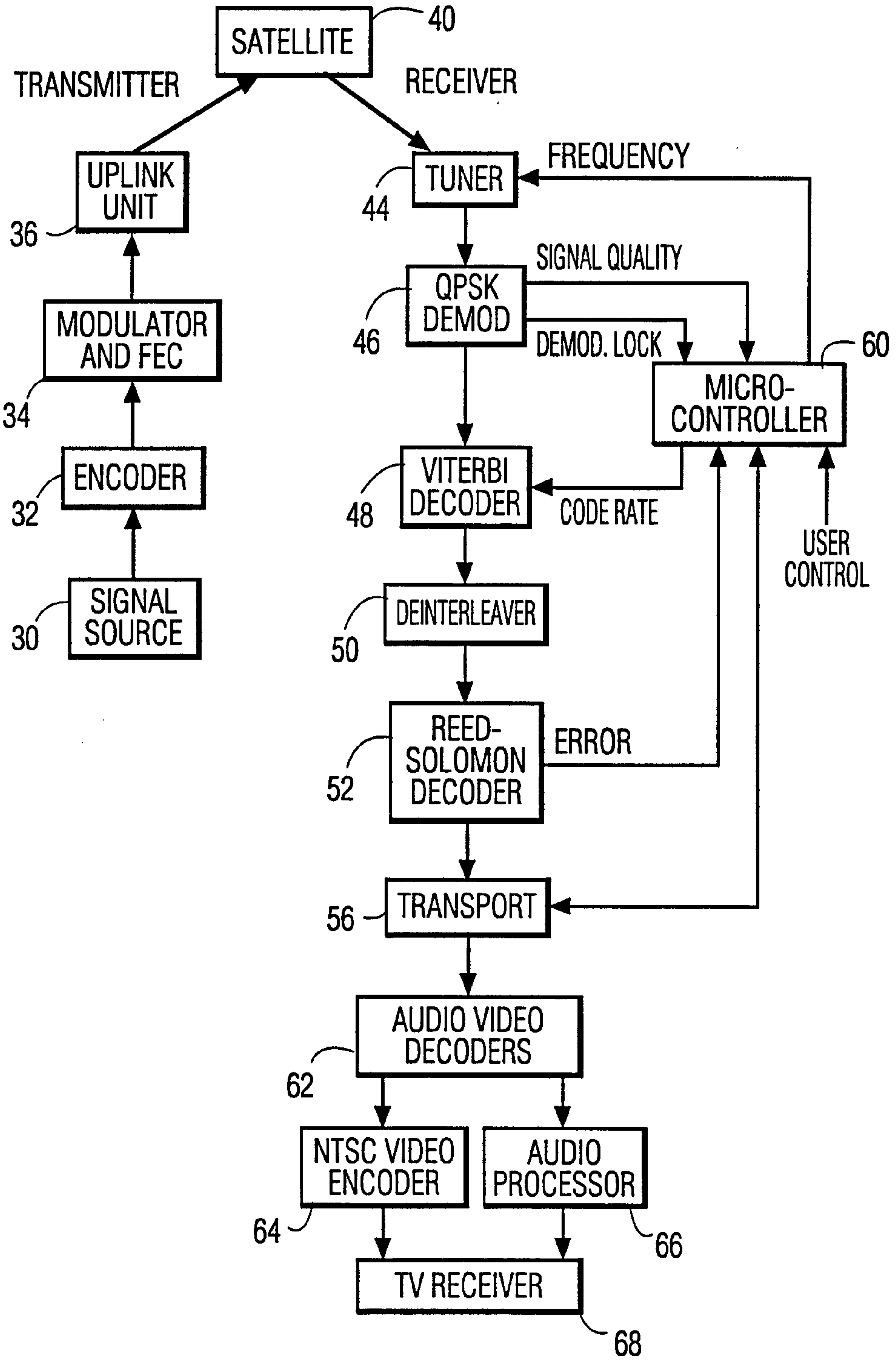

SATELLITE
40
TRANSMITTER
RECEIVER
UPLINK UNIT
36
TUNER
44
FREQUENCY
MODULATOR AND FEC
34
QPSK DEMOD
46
SIGNAL QUALITY
DEMOD. LOCK
MICRO-CONTROLLER
60
ENCODER
32
VITERBI DECODER
48
CODE RATE
USER CONTROL
SIGNAL SOURCE
30
DEINTERLEAVER
50
REED-SOLOMON DECODER
52
ERROR
TRANSPORT
56
AUDIO VIDEO DECODERS
62
NTSC VIDEO ENCODER
64
AUDIO PROCESSOR
66
TV RECEIVER
68

FIG.2

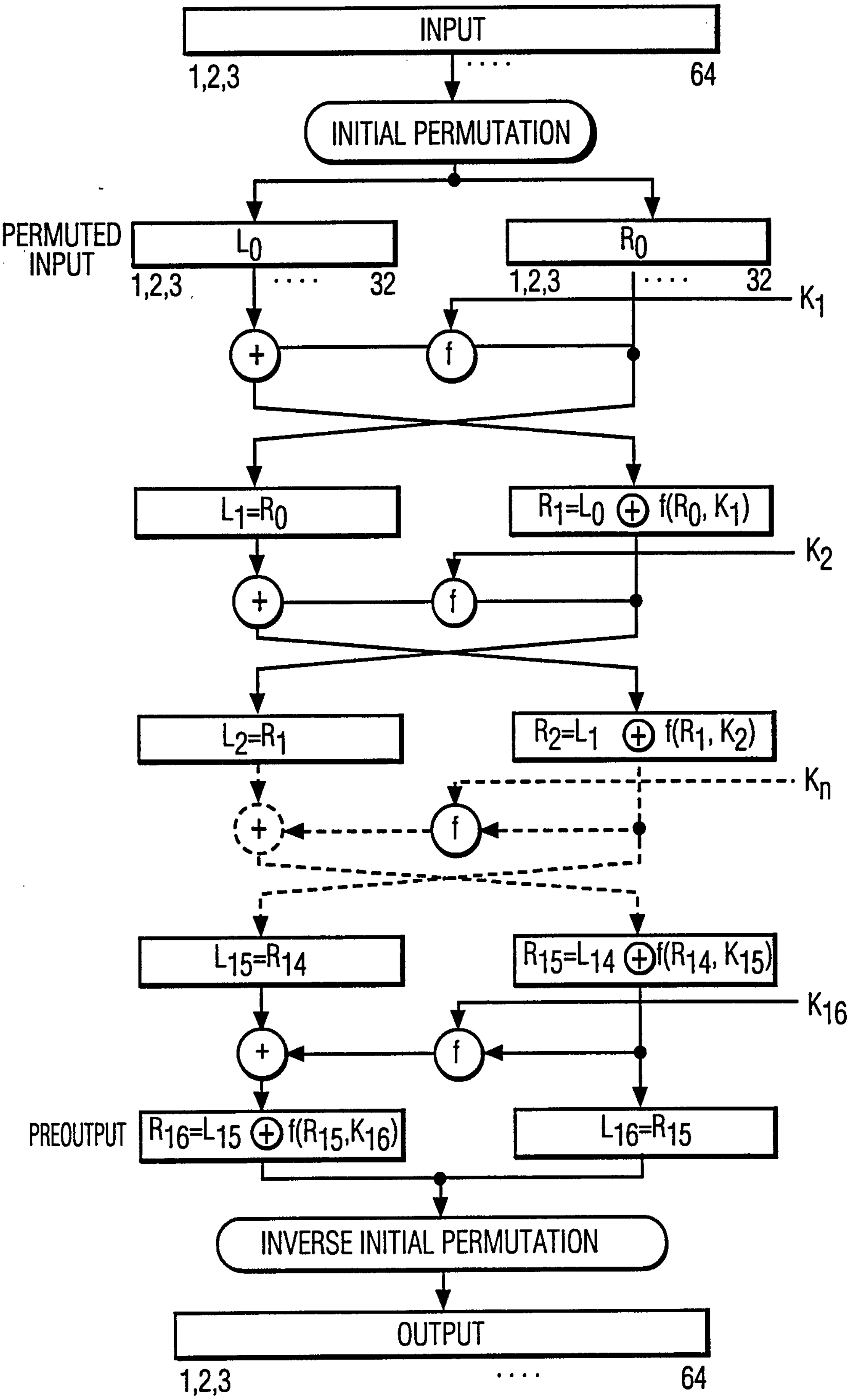

INPUT
1,2,3
....
64
INITIAL PERMUTATION
PERMUTED INPUT
L_0
1,2,3
....
32
R_0
1,2,3
....
32
K_1
+
f
$L_1=R_0$
$R_1=L_0 \oplus f(R_0, K_1)$
K_2
+
f
$L_2=R_1$
$R_2=L_1 \oplus f(R_1, K_2)$
K_n
+
f
$L_{15}=R_{14}$
$R_{15}=L_{14} \oplus f(R_{14}, K_{15})$
K_{16}
+
f
PREOUTPUT
$R_{16}=L_{15} \oplus f(R_{15},K_{16})$
$L_{16}=R_{15}$
INVERSE INITIAL PERMUTATION
OUTPUT
1,2,3
....
64

FIG. 3

TABLE 1
INITIAL PERMUTATION IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

FIG.4

TABLE 2
INVERSE OF INITIAL PERMUTATION IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

FIG.5

TABLE 3
EXPANSION FUNCTION E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

FIG.7

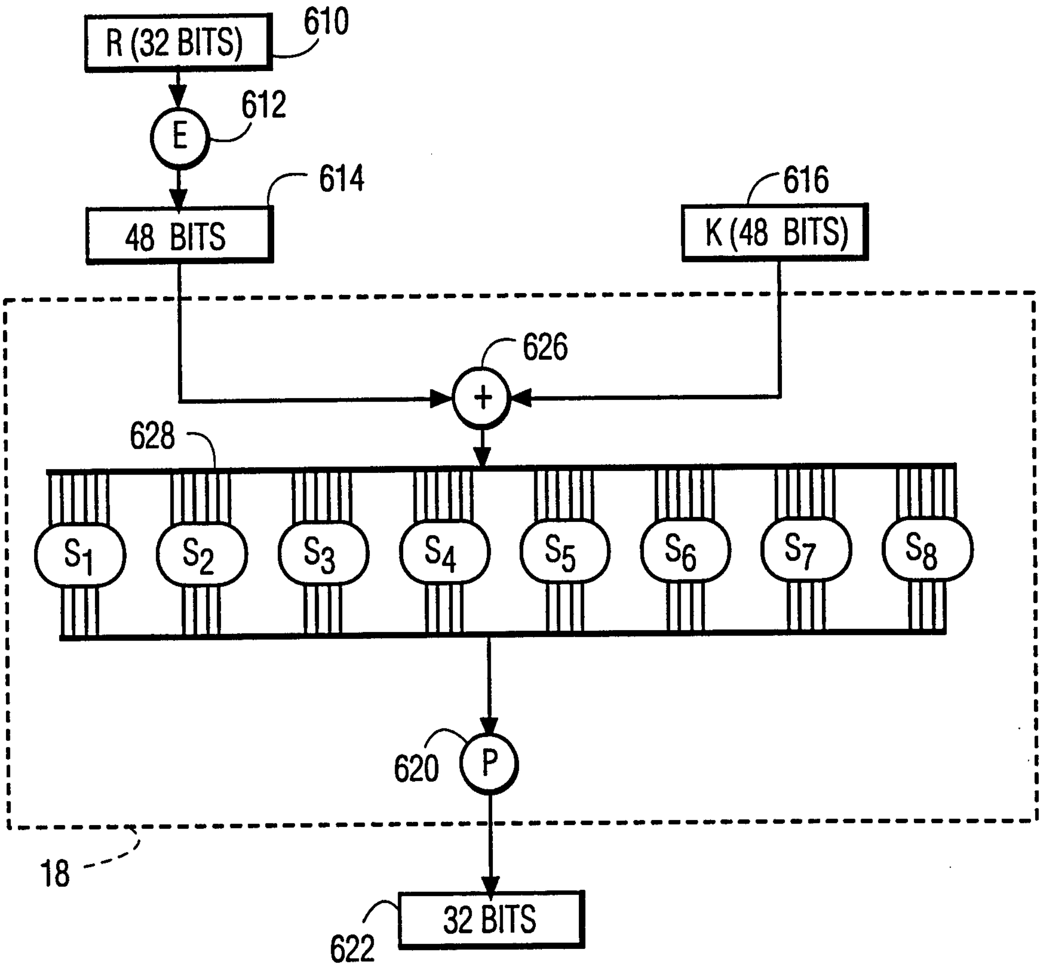

R (32 BITS)
610
E
612
48 BITS
614
K (48 BITS)
616
+
626
628
S1
S2
S3
S4
S5
S6
S7
S8
P
620
18
32 BITS
622

FIG.6

TABLE 4
SELECTION FUNCTION S_1

ROW NO.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

FIG.8

TABLE 5
PERMUTATION FUNCTION P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

FIG.10

TABLE 6
SELECTION FUNCTION $S_1,...,S_8$

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

FIG.9

B	OUTPUT	B	OUTPUT	B	OUTPUT	B	OUTPUT
0	14	16	13	32	4	48	15
1	0	17	10	33	15	49	5
2	4	18	10	34	1	50	12
3	15	19	6	35	12	51	11
4	13	20	6	36	14	52	9
5	7	21	12	37	8	53	3
6	1	22	12	38	8	54	7
7	4	23	11	39	2	55	14
8	2	24	5	40	13	56	3
9	14	25	9	41	4	57	10
10	15	26	9	42	6	58	10
11	2	27	5	43	9	59	0
12	11	28	0	44	2	60	5
13	13	29	3	45	1	61	6
14	8	30	7	46	11	62	0
15	1	31	8	47	7	63	13

4-BIT OUTPUT
6-BIT INPUT B

FIG.11

4-BIT OUTPUT	6-BIT POSSIBLE B			
0	28	1	62	59
1	6	15	34	45
2	8	39	44	11
3	29	16	53	56
4	2	32	41	7
5	27	49	24	60
6	19	20	42	61
7	5	30	47	54
8	14	31	37	38
9	25	26	43	52
10	17	18	57	58
11	12	23	46	51
12	21	22	51	46
13	13	4	40	63
14	9	0	55	36
15	3	10	53	48

FIG.12

```
WITH B SELECTED
P(1 TO 4) <= "0000" WHEN "011100" OR "000001" OR "111110" OR "111011",
             "0001" WHEN "000110" OR "001111" OR "100010" OR "101101",
             "0010" WHEN "001000" OR "001011" OR "101100" OR "100111",
             "0011" WHEN "010000" OR "011101" OR "111000" OR "110101",
             "0100" WHEN "000010" OR "000111" OR "100000" OR "101001",
             "0101" WHEN "011000" OR "011011" OR "111100" OR "110001",
             "0110" WHEN "010100" OR "010011" OR "101010" OR "111101",
             "0111" WHEN "011110" OR "000101" OR "110110" OR "101111",
             "1000" WHEN "001110" OR "011111" OR "100110" OR "100101",
             "1001" WHEN "011010" OR "011001" OR "110100" OR "101011",
             "1010" WHEN "010010" OR "010001" OR "111010" OR "111001",
             "1011" WHEN "001100" OR "010111" OR "101110" OR "110011",
             "1100" WHEN "010110" OR "010101" OR "110010" OR "100011",
             "1101" WHEN "000100" OR "001101" OR "101000" OR "111111",
             "1110" WHEN "000000" OR "001001" OR "100100" OR "110111",
             "1111" WHEN "001010" OR "000011" OR "110000" OR "100001";
```

FIG.13

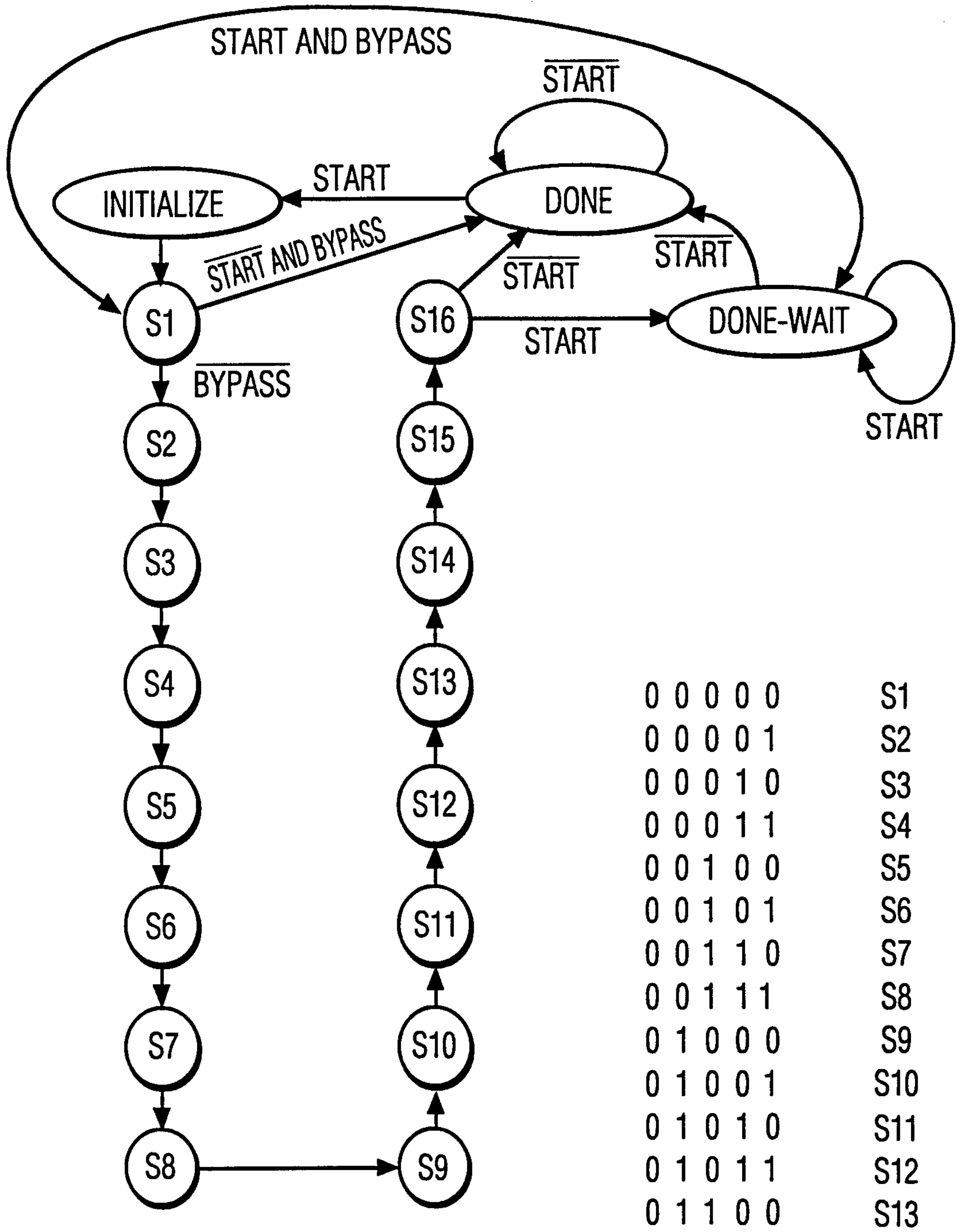

START AND BYPASS
$\overline{\text{START}}$
START
INITIALIZE
DONE
$\overline{\text{START}}$ AND BYPASS
$\overline{\text{START}}$
$\overline{\text{START}}$
S1
$\overline{\text{BYPASS}}$
S16
START
DONE-WAIT
START
S2
S15
S3
S14
S4
S13
S5
S12
S6
S11
S7
S10
S8
S9

FIG.14

0 0 0 0 0	S1
0 0 0 0 1	S2
0 0 0 1 0	S3
0 0 0 1 1	S4
0 0 1 0 0	S5
0 0 1 0 1	S6
0 0 1 1 0	S7
0 0 1 1 1	S8
0 1 0 0 0	S9
0 1 0 0 1	S10
0 1 0 1 0	S11
0 1 0 1 1	S12
0 1 1 0 0	S13
0 1 1 0 1	S14
0 1 1 1 0	S15
0 1 1 1 1	S16
1 0 0 0 0	INITIALIZE
1 0 0 0 1	DONE-WAIT
⋮	
1 1 1 1 1	DONE

FIG.15

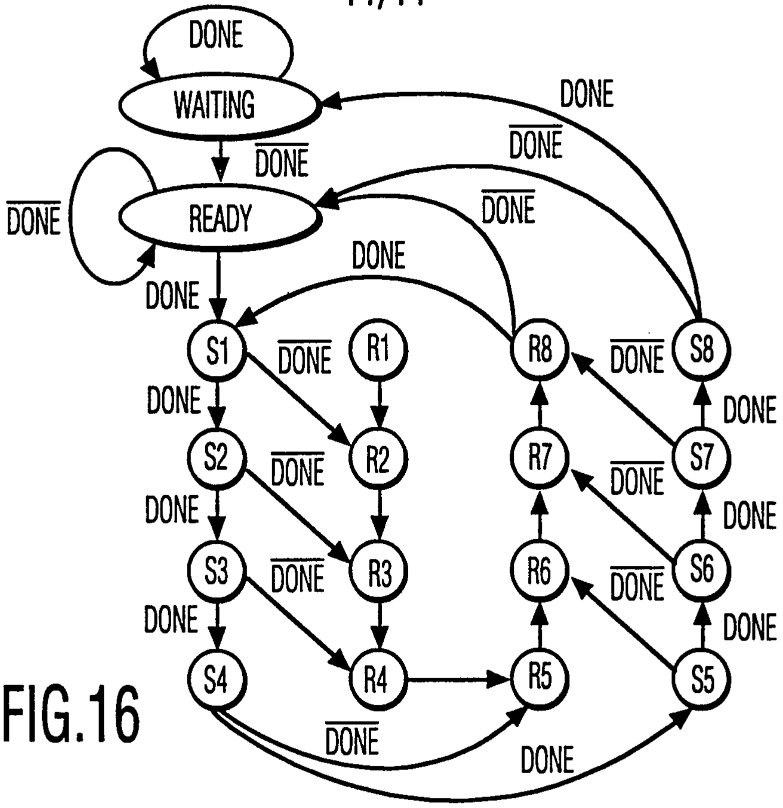


FIG.16

0	0	0	0	0	S1
0	0	0	0	1	S2
0	0	0	1	0	S3
0	0	0	1	1	S4
0	0	1	0	0	S5
0	0	1	0	1	S6
0	0	1	1	0	S7
0	0	1	1	1	S8
0	1	0	0	0	R1
0	1	0	0	1	R2
0	1	0	1	0	R3
0	1	0	1	1	R4
0	1	1	0	0	R5
0	1	1	0	1	R6
0	1	1	1	0	R7
0	1	1	1	1	R8
1	0	0	0	0	WAITING
1	0	0	0	1	NOT USED
		⋮			
1	0	1	1	1	
1	1	0	0	0	READY
1	1	0	0	1	NOT USED
		⋮			
1	1	1	1	1	

FIG.17

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/02942

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :H04L 9/00
US CL :380/20
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/20,29,49

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 4,172,213 (BARNES ET AL) 23 OCTOBER 1979. SEE FIGS. 1, 10 AND 11.	1-6
Y	US, A, 4,274,085 (MARINO, JR) 16 JUNE 1981 SEE FIGS. 1-7	1-6
Y	US, A, 4,484,027 (LEE ET AL) 20 NOVEMBER 1984 SEE FIGURE.	1-6
X	US, A, 4,613,901 (GILHOUSEN ET AL) 23 SEPTEMBER 1986 SEE ENTIRE DOCUMENT.	1-6
Y	US, A, 4,731,843 (HOLMQUIST) 15 MARCH 1988 SEE FIGS. 2-4.	1-6

☒ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 08 JUNE 1994	Date of mailing of the international search report 05.07.94
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Diane Goodwyn for SALVATORE CANGIALOSI Telephone No. (703) 305-0482

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/02942

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, E, RE 33,189 (LEE ET AL) 27 MARCH 1990 SEE FIGURE.	1-6
X	US, A, 5,159,633 (NAKAMURA) 27 OCTOBER 1992 SEE FIG. 2.	1-4, 6
X	US, A, 5,237,610 (GAMMIE ET AL) 17 AUGUST 1993 SEE FIGS. 7-13	1-6
X	US, A, 5,285,497 (THATCHER, JR) 08 FEBRUARY 1994 SEE ENTIRE DOCUMENT	1-6