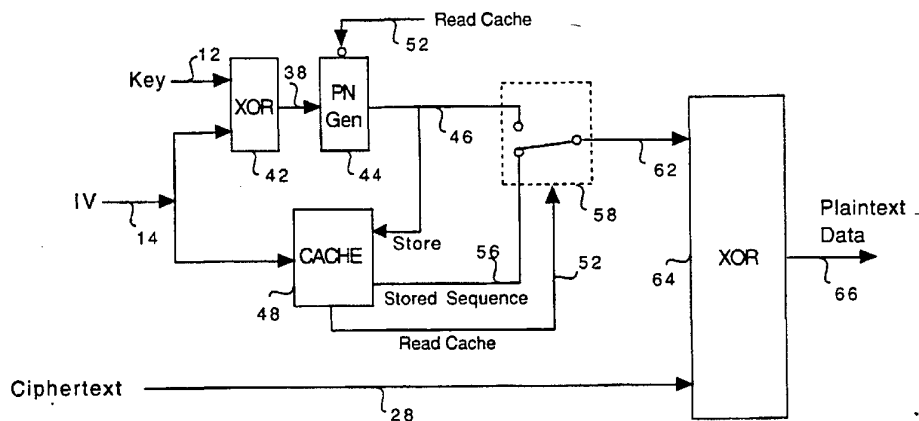




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification⁶ : H04L 9/18, 9/22, 9/12</p>	<p>A1</p>	<p>(11) International Publication Number: WO 95/06373 (43) International Publication Date: 2 March 1995 (02.03.95)</p>
<p>(21) International Application Number: PCT/US94/09509 (22) International Filing Date: 23 August 1994 (23.08.94) (30) Priority Data: 08/110,402 23 August 1993 (23.08.93) US 08/254,774 6 June 1994 (06.06.94) US (71) Applicant: APPLE COMPUTER, INC. [US/US]; One Infinite Loop, Cupertino, CA 95014 (US). (72) Inventors: LYNN, Kerry, E.; 1176 Parkwood Way, Redwood City, CA 94061 (US). ZWEIG, Jonathan, M.; 3272 St. Ignatius Place, Santa Clara, CA 95051 (US). MINCHER, Richard, W.; 1539 Miller Avenue, San Jose, CA 95129 (US). (74) Agents: FERRELL, John, S.; Carr, DeFilippo & Ferrell, Suite 200, 2225 East Bayshore Road, Palo Alto, CA 94303 (US) et al.</p>	<p>(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, GE, HU, JP, KE, KG, KP, KR, KZ, LK, LT, LU, LV, MD, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SI, SK, TJ, TT, UA, 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), ARIPO patent (KE, MW, SD). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>	

(54) Title: METHOD AND APPARATUS FOR DECRYPTION USING CACHE STORAGE



20

(57) Abstract

A method and apparatus for decryption using cache storage wherein imported ciphertext is decrypted to produce unencrypted plaintext data. As a communication sequence containing an initialization vector and a block of ciphertext is imported, the initialization vector is applied to a cache and to a decoder. The initialization vector is then compared with other initialization vectors stored in the cache to determine whether the specific initialization vector has previously been received and stored. If the specific initialization vector is found to be stored in the cache, then the PN sequence associated with that initialization vector is written to the decoder, and the stored PN sequence is used to decode the imported ciphertext. If a determination is made that the initialization vector has not been previously received, then the read cache signal instructs the multiplexer to connect the PN generator to the decoder, and the initialization vector is used to generate a new PN sequence. In order to improve the efficiency of future ciphertext decoding utilizing this specific initialization vector, the PN sequence associated with the initialization vector is then stored in the cache together with its corresponding initialization vector. When the next block of ciphertext is received using the same initialization vector, the PN sequence need not be regenerated by the PN generator, but rather may be read from the cache as a stored sequence.

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

1 METHOD AND APPARATUS FOR DECRYPTION
2 USING CACHE STORAGE

3
4 BACKGROUND OF THE INVENTION

5 1. Field of the Invention

6 This invention relates to data encryption, and more
7 particularly to a method and apparatus for varying the
8 computational overhead associated with encrypting and decrypting
9 digital data signals by selectively reusing, according to the desired
10 level of security, a pseudorandom encoding sequence at the
11 transmitter end and by storing and reusing pseudorandom
12 decoding sequences at the receiver end.

13
14 2. Description of the Background Art

15 Data encryption is a function that ensures the privacy of a
16 digital communication by preventing an unauthorized receiver
17 from understanding the contents of a transmitted message. A
18 conventional "symmetric key" cryptosystem is generally illustrated
19 in FIG. 1(a). A transmitter transforms a plaintext message into
20 ciphertext using an invertible encryption transformation. This
21 transformation is a function of the plaintext input message and a
22 secret key which is shared by both the transmitter and the
23 receiver. The ciphertext is then transmitted over an unsecured
24 public channel and the intended receiver of the message, also in
25 possession of the secret key, applies the inverse transformation to
26 decrypt the ciphertext and recover the original plaintext message.
27 The secret key is communicated to a plurality of authorized users
28 through a secure channel (for example, a secure Key Exchange
29 Algorithm may be employed) and the key effectively dictates a
30 specific encryption transformation from a family of cryptographic
31 transformations. In general, any station in possession of the secret
32 key may encrypt or decrypt messages.

33 A conventional cryptosystem can be said to exhibit
34 "unconditional security" if the secret key is as long as the
35 ciphertext message, each key is used only once, and all keys are
36 equally likely. However, since most systems can be expected to
37 transmit a large number of messages, the problem of distributing
38 the key information becomes formidable. Most practical
39 cryptosystems have short keys compared to the length of a
40 message. The lessened security resulting from short keys is
41 compensated for by relying on the complexity of the way that the
42 key is combined with the data.

43 A particular example of a conventional cryptosystem,
44 hereafter referred to as an electronic codebook, is generally
45 illustrated in FIG. 1(b). The electronic codebook involves the use of

1 a secret key that is shared by both the transmitter and the
2 receiver. The transmitter utilizes the key to generate a
3 deterministic, apparently random sequence of binary digits using a
4 Pseudorandom Number (PN) generator. An essential feature of the
5 PN generator is that with a specific key input, a unique PN
6 sequence of arbitrary length may be generated. The PN sequence
7 is then combined with the binary representation of the plaintext
8 message to be encrypted to produce a sequence of ciphertext. The
9 combination of the PN sequence and the plaintext must be
10 accomplished using an invertible function. An invertible function
11 is one that has a known inverse such that when the inverse
12 function is applied to the ciphertext the original plaintext can be
13 extracted. For example, two's complement addition or bit-wise
14 exclusive-OR (XOR) are two widely used invertible functions,
15 although other functions can be employed.

16 Decoding of the encrypted ciphertext may be performed by
17 the receiver using a method identical to that used by the
18 transmitter. Ciphertext is received from the transmitter and
19 combined using a logical XOR gate, with a pseudorandom sequence
20 generated by a PN generator identical to that used in the
21 transmitter. The essence of the electronic codebook system is that
22 an encryption key is used to generate a pseudorandom sequence in
23 the transmitter side, and the identical sequence is then generated
24 in the receiver when the same encryption key is applied to the
25 receiver PN generator. The XOR gate in the receiver provides the
26 inverse function of the XOR gate in the transmitter so that logical
27 combination of the ciphertext and the PN sequence in the receiver
28 produces the same plaintext that was originally encoded by the
29 transmitter.

30 One drawback of the prior art system described is that the
31 overhead of generating PN sequences is quite high, particularly
32 relative to the overhead of applying the combination function. In
33 practice, it is typical to generate and combine the PN sequence
34 with a plaintext message of arbitrary length one character at a
35 time, as needed. The characters of the PN sequence are discarded
36 after a single use, so there is no opportunity to spread the cost of
37 computing the sequence over several messages. The rate at which
38 messages can be encrypted and decrypted is therefore limited by
39 the speed at which the PN sequence can be produced. What is
40 needed is a method for storing and reusing PN sequences in order
41 to increase the transmission rate of messages through the
42 cryptosystem.

43 Another drawback of the prior art system is that the
44 receiver's PN generator may lose synchronization with that of the
45 transmitter under some circumstances, necessitating additional

1 recovery procedures in order for the plaintext to be recovered. For
2 example, if the next character emitted by the PN generator is a
3 function of the initial key input as well as the number of
4 characters that have been previously emitted, and if the message
5 is being communicated from the transmitter to the receiver in
6 several fragments or packets, and if any packets are lost or
7 received out of order, then it will first be necessary for the
8 receiver to receive and arrange all the fragments in the proper
9 order before decoding of the message can be accomplished. It is
10 therefore desirable that a high speed cryptosystem exhibit the
11 property of self-synchronization between transmitter and receiver
12 such that no additional recovery procedures are required to decode
13 messages.

14

15 SUMMARY OF THE INVENTION

16 In accordance with the present invention, an apparatus and
17 method are described for variable overhead cached encryption and
18 decryption. A transmitter unit is used for encoding or encrypting
19 data and a separate authorized receiver decodes or decrypts the
20 data. Both the transmitter and receiver share a common secret
21 key that has been communicated through some separate channel.
22 The transmitter combines the secret key (which serves as a
23 constant base value) with an Initialization Vector (IV), using an
24 XOR operation to produce a temporal key. This temporal key is
25 then used as an input to a pseudorandom number (PN) generator
26 to produce a unique PN sequence of binary digits, for each new
27 temporal key entered. The generated PN sequence is equal in
28 length to the longest anticipated message fragment. The
29 initialization vector together with its corresponding PN sequence is
30 then stored in a cache and the PN sequence is iteratively reused, as
31 determined by a counter, to encrypt one or more plaintext
32 messages. The counter is initialized to a maximum count value
33 whenever a new PN sequence is generated, and the counter tracks
34 reuse of the PN sequence to encrypt the number of messages
35 specified by the maximum count value. When the maximum count
36 value specifies that the PN sequence is to be used only once, the
37 security afforded by the present invention will be high, but a new
38 PN sequence must be generated for each message sequence
39 transmitted and so the computational overhead will also be high.
40 If the maximum count value specifies a maximum count value
41 greater than one, the PN sequence stored in the cache will be
42 reused to encrypt the maximum count number of message
43 sequences. The resulting ciphertext messages will be more
44 vulnerable to statistical cryptanalytic attack as the maximum
45 count value increases. The PN sequence from the cache is

1 combined with the plaintext data to be transmitted using an
2 invertable combination function. An exclusive-OR (XOR) function is
3 used in the preferred embodiment to produce a ciphertext
4 message. The unencrypted initialization vector is then
5 concatenated with the ciphertext, and together, both are exported
6 by the transmitter to the receiver for decrypting. As each
7 plaintext message is encrypted and exported, the value of the
8 counter is decremented. If the value of the counter goes to zero
9 then a new initialization vector is selected and the above steps are
10 repeated for subsequent messages. A new initialization vector
11 should be chosen with equal probability from the set of all possible
12 initialization vectors since this has the desirable result of selecting
13 a large number of different encoding sequences over the life of the
14 secret key.

15 The encoded communication is imported by the receiver and
16 the initialization vector portion is extracted. The receiver's cache
17 of previously received initialization vectors is searched using the
18 imported initialization vector as a search key to determine
19 whether an entry exists for it in the cache. If the initialization
20 vector has been previously received and stored, then the
21 corresponding PN sequence has already been computed and stored
22 and is available for decoding the imported ciphertext without
23 having to regenerate the PN sequence. If the imported
24 initialization vector is not found in the cache, then the associated
25 PN sequence is not available and the receiver then combines the
26 initialization vector with the secret key to produce a temporal key
27 and corresponding PN sequence identical to the sequence used by
28 the transmitter to encode the data. This PN sequence is then
29 combined with the ciphertext, using an XOR gate, to recover the
30 original plaintext from the ciphertext. The initialization vector and
31 corresponding newly generated PN sequence are then stored in the
32 receiver cache, to be available for comparison with subsequently
33 received initialization vectors. Utilization of this cache can greatly
34 reduce the overhead associated with generating PN sequences,
35 particularly when higher count values are used by a given
36 transmitter.

37

38 BRIEF DESCRIPTION OF THE DRAWINGS

39 FIG. 1(a) is a block diagram showing a conventional
40 symmetric key cryptosystem;

41 FIG. 1(b) is a block diagram showing an example electronic
42 codebook cryptosystem of the prior art;

43 FIG. 2 is a block diagram showing the transmitter of the
44 variable-overhead cached encryption system of the present
45 invention;

1 FIG. 3 is a block diagram showing the receiver of the
2 variable-overhead cached encryption system of the present
3 invention;

4 FIG. 4(a) is a block diagram showing a general purpose
5 computer which is used to implement the cached encryption
6 system of the present invention;

7 FIG. 4(b) is a table showing the arrangement of cached data of
8 the present invention, in which each member of a list of
9 initialization vectors is stored together with its corresponding
10 pseudorandom sequence;

11 FIG. 5 is a flow diagram showing the method steps of
12 transmitting encrypted data using the apparatus of FIG. 2; and

13 FIG. 6 is a flow diagram showing the method steps of
14 receiving encrypted data using the apparatus of FIG. 3.

1 DESCRIPTION OF THE PREFERRED EMBODIMENT

2 The encryption-decryption system of the present invention
3 consists of a unique combination of digital functional blocks, all of
4 which are separately conventional and well understood in the art.
5 The system is preferably implemented on a general purpose
6 computer using programmed instructions; however, the discussion
7 which follows teaches the invention in terms of functional blocks
8 which may be readily implemented using conventional discrete or
9 integrated digital circuitry. The preferred implementation is
10 described with reference to FIGS. 4 and 5 below.

11 Referring now to FIG. 2, a transmitter 10 is shown for
12 encrypting plaintext data 32 into ciphertext 28. Plaintext data 32
13 is digital information which may be readily understood by both a
14 sender and a receiver and may also be readily understood by other
15 unauthorized third parties having access to the communications
16 channel. The function of transmitter 10 is to encode or encrypt the
17 plaintext data 32 in such a way that the information is usable only
18 to a receiver having a bona fide access to the data. A central
19 feature of transmitter 10 is a key 12 which is secret as to third
20 parties but shared between the transmitter and a receiver 20
21 (shown in FIG. 3) of data 32. As discussed with reference to FIG.
22 1(a), key 12 would ideally be infinite in length and would be
23 unique as to every message communicated between the
24 transmitter 10 and the receiver 20. In practice, however, key 12
25 is relayed only periodically between the transmitter 10 and the
26 receiver 20 and during the periods between the relay of the key,
27 the key is used repetitively to encrypt plaintext data 32 from
28 transmitter 10 before transmission to receiver 20.

29 An initialization vector (IV) 14 is produced by IV generator
30 29 and utilized by the transmitter 10 and receiver 20 to extend
31 the usability of the key 12. The key 12 is a relatively expensive
32 component to generate and maintain. The key 12 must be
33 randomly generated and must be securely transmitted between
34 transmitter 10 and receiver 20 in a secure channel which is
35 separate from the communication system through which ciphertext
36 28 is transmitted. Consequently, even though the security of the
37 key 12 diminishes with each successive use, efficiency demands
38 that maximum utilization of the key occurs. One way of extending
39 the utilization of the key 12 is to combine the key with a local key
40 such as the initialization vector 14. IV generator 29 generates a
41 random sequence having the same length as key 12. Generator 29
42 repeats the same IV sequence until reset 25 signals that a new
43 sequence is to be generated. Initialization vector 14 is combined
44 with key 12 using a conventional exclusive-OR (XOR) gate 16 to
45 produce a temporal key 17. Various other logical functions can be

1 equivalently used in place of XOR gate 16 to mask the identity of
2 the key. This logical function need not be invertable. The XOR
3 function is applied bitwise and is defined by a logical "0" whenever
4 all inputs are the same, and a logical "1" otherwise. Initialization
5 vector 14 is transmitted to receiver 20 as part of the
6 communication sequence containing the ciphertext output 28.
7 Information transmitted from transmitter 10 to receiver 20
8 includes a block of ciphertext 28 concatenated with initialization
9 vector 14. In essence, the initialization vector 14 becomes public
10 in that it is transmitted in an unencrypted format and may be
11 more easily appropriated by third parties. However, since
12 initialization vector 14 is always encoded with key 12 to produce
13 temporal key 17, the value knowing of this initialization vector is
14 limited. Since the initialization vector 14 is merely a component of
15 temporal key 17, it would be difficult to determine the value of the
16 temporal key knowing only the value of the initialization vector.

17 Temporal key 17 acts as a seed to a Pseudorandom Number
18 (PN) generator 18. PN generator 18 is a deterministic machine,
19 conventional in the art, and characterized by the fact that given a
20 specific input or seed value, a unique and repeatable output
21 sequence of arbitrary length can be generated. This output
22 sequence from PN generator 18 is referred to in FIG. 2 as a
23 temporal sequence 23 and is equal in length to the longest
24 anticipated plaintext data 32. Once generated, the temporal
25 sequence 23, is then stored in cache 22, a conventional memory
26 register. The contents of cache 22 is then written as a PN sequence
27 to XOR gate 26. XOR gate 26 is similar in construction to XOR gate
28 16 and is used to combine the PN sequence 24 with the plaintext
29 data 32 to produce ciphertext 28.

30 An additional feature of the present invention is counter 21,
31 which controls the generation of new initialization vectors 14 and
32 thereby the security level of the encryption system. Cache 22
33 contains the temporal sequence 23 produced by the PN generator
34 18 in response to the input combination of the initialization vector
35 14 and the key 12. In the preferred embodiment, cache 22 is
36 designed to contain one or more temporal sequences 23 arranged
37 as a function of initialization vectors 14. For a specific initialization
38 vector 14, a corresponding temporal sequence 23 will be stored. A
39 further discussion of the implementation of cache 22 can be found
40 with reference to the discussion of FIGS. 4(a) and 4(b) below.
41 Counter 21 selectively resets IV generator 29, enabling the
42 iterative reuse of a specific initialization vector 14 and
43 corresponding temporal sequence 23 in order to improve the
44 efficiency of the transmitter 10. The counter 21 is operated by
45 initially loading a maximum count signal 19 into the counter 21.

1 As each new data sequence 32 is present, a decrement signal 27
2 instructs counter 21 to decrement. When counter 21 decrements
3 to zero, then a new initialization vector 14 is subsequently utilized
4 by XOR 16 in generating a new temporal key 17. With each
5 sequence of plaintext data 32 combined in XOR gate 26, a PN
6 sequence 24 of identical length is read from cache 22 by XOR 26.
7 With each new plaintext data 32 sequence, the decrement signal
8 27 reduces the counter 21 contents by one. The encrypting
9 process proceeds in XOR gate 26, reading PN sequences 24 and
10 decrementing counter 21 until the counter 21 contents reaches
11 zero causing the IV generator 29 to reset. Resetting the IV
12 generator 29 results in the generation of a new initialization vector
13 14. Counter 21 has been described with respect to FIG. 2 as a
14 plaintext data 32 sequence counter, decrementing with each
15 sequence processed. Counter 21 equivalently implements a timer
16 or clock function, resetting the IV generator 29 after a period of
17 time set by Max Count 21. In this way, initialization vector 14
18 extends the usability of the key 12 by making the corresponding
19 PN sequence 24 more difficult to determine. Use of counter 21 and
20 cache 22 serve the purpose of reducing the costly overhead
21 associated with generating PN sequences 24 by reusing the
22 sequences generated and stored in the cache 22. The counter 21
23 enables variability of the overall security of the transmitter 10 and
24 receiver 20 by providing a selection of the number of times each
25 specific temporal sequence 23 is used in the encoding of data. In
26 the preferred embodiment, the counter 21, reset 25, maximum
27 count 19, and decrement 27 signals are implemented in the central
28 processing unit of a conventional general purpose computer.

29 Referring now to FIG. 3, a receiver 20 is shown in which a
30 ciphertext 28 is decoded to produce an unencrypted plaintext data
31 66 which is identical to the plaintext data 32 sequence of
32 transmitter 10. As the communication sequence containing an
33 initialization vector 14 and a block of ciphertext 28 is imported by
34 receiver 20, the initialization vector 14 is stripped off and applied
35 to cache 48 and to XOR gate 42. Other functions may be
36 equivalently substituted in place of XOR gate 42; however, gate 16
37 and gate 42 must be identical. Initialization vector 14 is then
38 compared in cache 48 with other initialization vectors stored in
39 cache 48 to determine whether the specific initialization vector 14
40 has previously been received and stored. If the specific
41 initialization vector 14 is found to be stored in cache 48, then the
42 PN sequence associated with that initialization vector is written to
43 an XOR gate 64, and the stored PN sequence is used to decode the
44 imported ciphertext 28. When a match of the received
45 initialization vector 14 is made to a stored initialization vector in

1 cache 48, read cache signal 52 instructs multiplexer 58 to route the
2 stored sequence 56 output to the XOR gate 64. From the viewpoint
3 of the XOR gate 64, the PN sequence stored in cache 48 becomes
4 the selected sequence and is delivered through multiplexer 58 via
5 the stored sequence 56 output of the cache.

6 If a determination is made that the initialization vector 14
7 has not been previously received, then the read cache signal 52 of
8 cache 48 signals multiplexer 58 to connect the PN generator 44 to
9 the XOR gate 64. In this event, initialization vector 14 is used in
10 producing a temporal key 38 input to PN generator 44 to generate
11 a new PN sequence 46 identical to the corresponding PN sequence
12 23 used in the encoding of the ciphertext 28 by the transmitter 10.
13 The read cache signal 52 is then inverted and used to enable the
14 output of the PN generator 44. Just as in the case with the
15 transmitter 10, initialization vector 14 is combined with key 12 in
16 XOR gate 42 to produce a temporal key 38. It should be noted that
17 this temporal key 38 is identical to the corresponding temporal
18 key 17 produced in the transmitter 10 by the XOR gate 16
19 combination of key 12 and initialization vector 14. PN generator
20 44 receives temporal key 38 to produce a PN sequence 46, which is
21 then connected via multiplexer 58 to XOR gate 64 as a selected
22 sequence 62. In order to improve the efficiency of future decoding
23 of ciphertext 28 utilizing this specific initialization vector 14, the
24 PN sequence associated with the initialization vector is then stored
25 in cache 48 together with its corresponding initialization vector.
26 When the next block of ciphertext 28 is received using the same
27 initialization vector 14, the PN sequence 46 need not be
28 regenerated by PN generator 44, but rather may be read from
29 cache 48 as a stored sequence 56. It should further be noted that
30 the imported initialization vector 14 has a dual purpose: it is used
31 both as a component of the temporal key 17 for generating PN
32 sequence 46 and as an input to cache 48 for the purpose of
33 determining whether there exists a stored sequence 56
34 corresponding to the imported initialization vector 14. The XOR
35 gate 64 combines the selected sequence 62 with ciphertext 28 to
36 produce plaintext data 66 which is identical in content to the
37 corresponding plaintext data 32 originally encoded in transmitter
38 10.

39 An important benefit of the encryption system of the present
40 invention is that the transmitter 10 and receiver 20 are self-
41 synchronizing. That is, assuming the key is shared, everything
42 needed to decode a block of transmitted data is contained within
43 the message. Knowledge of prior messages or sequences is not
44 required.

1 Referring now to FIG. 4(a), a diagram is shown of a general
2 purpose computer 40 used for the preferred implementation of the
3 encryption system shown in FIGS. 2 and 3. The preferred
4 implementation of the present invention consists of programmed
5 instructions implemented on an Apple Macintosh[®] computer,
6 manufactured by Apple Computer, Inc. of Cupertino, California.
7 The general method steps, described below, can be equivalently
8 implemented on any general purpose computer and many other
9 programmable processor-based systems. The general purpose
10 computer 40 consists of a CPU 31 attached to a number of
11 processing components. CPU 31 contains a keyboard 37 and a CRT
12 35 through which a user can interact with CPU 31. The CPU 31 is
13 connected to a communication port 33 for interfacing with other
14 processors and communication devices, such as modems and area
15 networks. CPU 31 further comprises a data bus 45 for connecting
16 various memories, including program memory 39, cache memory
17 60, counter memory 43, and mass storage 41. Program memory
18 39 contains operating instructions for directing the control of CPU
19 31. Cache 60 contains high speed temporary memory for use by
20 CPU 31 in executing the encryption and decryption program
21 instructions of the present invention. Also attached to data bus 45
22 is mass storage 41 which contains stored data, utilized by CPU 31
23 in executing program instructions from program memory 39.

24 Referring also to FIGS. 2 and 3, the XOR gates 16, 26, 42 and
25 64 are implemented by CPU 31 using Boolean arithmetic; counter
26 21 is implemented using counter memory 43; and the caches 22
27 and 48 are implemented using cache 60 memory. PN generator 18
28 and 44 are implemented by the CPU 31 using a conventional
29 pseudorandom number generator algorithm. Computer system 40
30 can implement the encryption system in a number of ways. A first
31 computer system can act as a transmitter 10 and export ciphertext
32 to a second computer system via the communication port 33. In
33 this operation mode, the first computer acts as transmitter 10
34 while the second computer acts as receiver 20. This first mode of
35 operation provides for a secure transmission of sensitive data.

36 In an alternative operating mode, a single computer system
37 40 acts as both a transmitter 10 and as a receiver 20, storing
38 ciphertext to mass storage 41 and later retrieving the stored
39 ciphertext for decoding and use. The purpose of this second mode
40 of operation is to allow for the secure storage of sensitive data.

41 Referring now to FIG. 4(b), a memory map of cache 60 is
42 shown in which a list of initialization vectors 72 are paired with
43 corresponding sequences 74. The entry "IV 1" has a corresponding
44 "Sequence 1", "IV 2" has a corresponding "Sequence 2", and "IV n"
45 has a corresponding "Sequence n". Cache 60 memory provides a

1 functional implementation of cache 22 in FIG. 2, when computer
2 system 40 is operating as a transmitter 10, and provides an
3 implementation of cache 48 in FIG. 3, when the computer system is
4 operating as a receiver 20. The counter 21 output in transmitter
5 10 is implemented as a CPU 31 function in which the CPU reads
6 and decrements the contents of counter memory 43 each time a PN
7 sequence is utilized to encode a sequence of plaintext data 32.

8 Referring now to FIG. 5, a flow chart is shown outlining the
9 programmed instruction steps which are executed by the general
10 purpose computer 40, acting in the mode of a transmitter 10 (FIG.
11 2) in encrypting plaintext data to produce the ciphertext 28 of the
12 present invention. Step 61 is the entry point for the encrypting
13 instructions of FIG. 5. If step 63 determines that the routine
14 variables have not been initialized, CPU 31 initializes the routine
15 variables in step 65 by setting the packet count to Max Count
16 generating an Initialization Vector (IV), and setting the PN
17 Sequence to the value NewSeq(IV XOR Secret Key). The variable
18 IV is equal to the initialization vector 14 and the variable Secret
19 Key is a previously determined and stored value equal to key 12.
20 The function "NewSeq()" is a conventional algorithm for
21 pseudorandom number generation, using the values of IV and
22 Secret Key as seed components. For example see Blahut, Richard,
23 Digital Transmission of Information, Addison Wesley Publishing
24 Company, 1990, p 497. The variable Packet Count represents the
25 maximum number of times that a particular initialization vector
26 can be used in the generation of a PN sequence 24. The maximum
27 value (Max Count) for the variable packet count is equal to the
28 maximum count signal 19. In step 67, packet count is
29 decremented by one, and in step 71 the CPU 31 tests whether
30 Packet Count is equal to zero. If Packet Count is equal to zero, then
31 the program returns to the initialization step in 65. In the event
32 that packet count is not equal to zero, a Ciphertext sequence is
33 calculated in step 73 using the formula:

$$34 \quad \text{Ciphertext}[i] = \text{PN Sequence}[i] \text{ XOR Plaintext}[i]$$

35
36
37 where i is an indexing integer ranging from zero to one less than
38 the length of the plaintext sequence in bits. It should be noted
39 that in this preferred method, the length of the plaintext, PN, and
40 ciphertext sequences are all of equal length. Following the
41 calculation of the ciphertext sequence, data strings called
42 "message.iv" and "message.data" are generated, in which
43 message.iv is set equal to the initialization vector sequence and
44 message.data is set equal to the ciphertext sequence. The routine
45 exits 77 at which time CPU 31 transmits message.iv and

1 message.data as a concatenated data string to communication port
2 33 or to mass storage 41 for transmission or storage.

3 Referring now to FIG 6, with the computer 40 acting in the
4 mode of a receiver 20 (FIG. 3), the concatenated data string
5 containing message.iv and message.data is received by CPU 31 in
6 step 87, and the initialization vector and ciphertext sequences are
7 separated. Using the initialization vector component (message.iv),
8 a search 89 of cache 60 is made for an initialization vector
9 matching the incoming message.iv. Since each initialization vector
10 in cache 60 is matched to a PN sequence, locating a matching
11 initialization vector to the incoming message.iv provides
12 identification of the PN sequence used to encrypt the incoming
13 message.data. If the message.iv can be matched 91 to a stored IV
14 and PN sequence, the receiver 20 will not have to expend the
15 overhead of creating a new PN sequence to decode the
16 message.data sequence. If the sequence is found in the cache 60,
17 then the plaintext data is determined 95 using the formula:

$$18 \quad \text{Plaintext}[i] = \text{PN Sequence}[i] \text{ XOR Ciphertext}[i]$$

19
20
21 If the sequence is not found 91 in the cache 60, then CPU 31
22 generates 93 the sequence using the same pseudorandom number
23 generation routine used in step 65 of FIG. 5, wherein:

$$24 \quad \text{PN Sequence} = \text{NewSeq}(\text{IV XOR Secret Key})$$

25
26
27 This PN Sequence is stored in cache 48 and then used in step 95 to
28 recover the plaintext data 66. The routine exits in step 97.

29 The invention has now been explained with reference to
30 specific embodiments. Other embodiments will be apparent to
31 those of ordinary skill in the art in light of this disclosure. For
32 example, the invertable function described in the preferred
33 embodiment is an XOR function. Other invertable functions are
34 equivalently effective. Also the counter 21 is shown as a "preset
35 with decrement-to-zero" function. Alternative up-counters and
36 the CPU-implemented increment-and-compare functions are
37 viewed as equivalents with respect to the present invention.
38 Therefore, it is not intended that this invention be limited, except
39 as indicated by the appended claims.

1 What is claimed is:

2

3 1. An apparatus for decryption using cache storage, the
4 apparatus comprising:

5 a memory for storing at least one number sequence;

6 a control signal responsive to the contents of the
7 memory which indicates whether a selected number sequence is
8 stored in the memory;

9 switching means having a first sequence input coupled
10 to the output of the memory, for selectively outputting the first
11 sequence input responsive to the indication of the control signal;
12 and

13 a decoder which receives the output of the switching
14 means as a first input and receives encrypted data as a second
15 input, and combines these first and second inputs to produce
16 decrypted data.

17

18 2. The apparatus according to Claim 1, further comprising:

19 a PN generator which generates and provides as an
20 output the selected number sequence.

21

22 3. The apparatus according to Claim 2, wherein the switching
23 means further comprises a second sequence input coupled to a first
24 output of the PN generator, and the switching means outputs one
25 of the sequence inputs responsive to the indication of the control
26 signal.

27

28 4. The apparatus according to Claim 2, wherein a second output of
29 the PN generator is coupled to the memory.

30

31 5. The apparatus according to Claim 1, wherein the decoder is an
32 exclusive-OR gate.

33

34 6. The apparatus according to Claim 1, wherein the number
35 sequence is a Pseudorandom Number sequence.

36

37 7. The apparatus according to Claim 3, wherein the memory stores
38 a new number sequence from the PN generator in response to the
39 indication of the control signal when the decoder is receiving data
40 from the second sequence input.

41

42 8. A decryption system comprising:

43 means for storing unique pairs of initialization vectors
44 and pseudorandom numbers;

- 1 means for searching the storage for a unique
2 initialization vector and pseudorandom number pair having an
3 initialization vector which matches an imported initialization
4 vector; and
5 means for decrypting imported ciphertext using the
6 pseudorandom number corresponding to the matched initialization
7 vector, if such an initialization vector match is found.
8
- 9 9. The decryption system according to Claim 8, further comprising
10 means for generating the pseudorandom number corresponding to
11 the imported initialization, if no initialization vector match is found
12 in the storage.
13
- 14 10. The decryption system according to Claim 9, further
15 comprising means for using the generated pseudorandom number
16 to decrypt the imported ciphertext, if no initialization vector match
17 is found in the storage.
18
- 19 11. The decryption system according to Claim 8, further
20 comprising means for importing the ciphertext in concatenated
21 combination with an initialization vector.
22
- 23 12. The decryption system according to Claim 11, further
24 comprising means for separating the ciphertext from the
25 initialization vector in the concatenated combination.
26
- 27 13. The decryption system according to Claim 9 wherein the
28 means for generating a pseudorandom number from the imported
29 initialization vector further comprises means for storing the
30 generated pseudorandom number and its corresponding
31 initialization vector in storage for future use in decrypting.
32
- 33 14. The decryption system according to Claim 8 wherein the
34 means for decrypting the imported ciphertext, further comprises
35 means for logically combining the imported ciphertext with the
36 stored pseudorandom sequence.
37
- 38 15. A method for decryption using cache storage, the method
39 comprising the steps:
40 storing unique pairs of initialization vectors and
41 pseudorandom numbers;
42 searching the storage for a unique initialization vector
43 and pseudorandom number pair having an initialization vector
44 which matches an imported initialization vector; and

1 decrypting imported ciphertext using the
2 pseudorandom number corresponding to the matched initialization
3 vector, if such an initialization vector match is found.
4

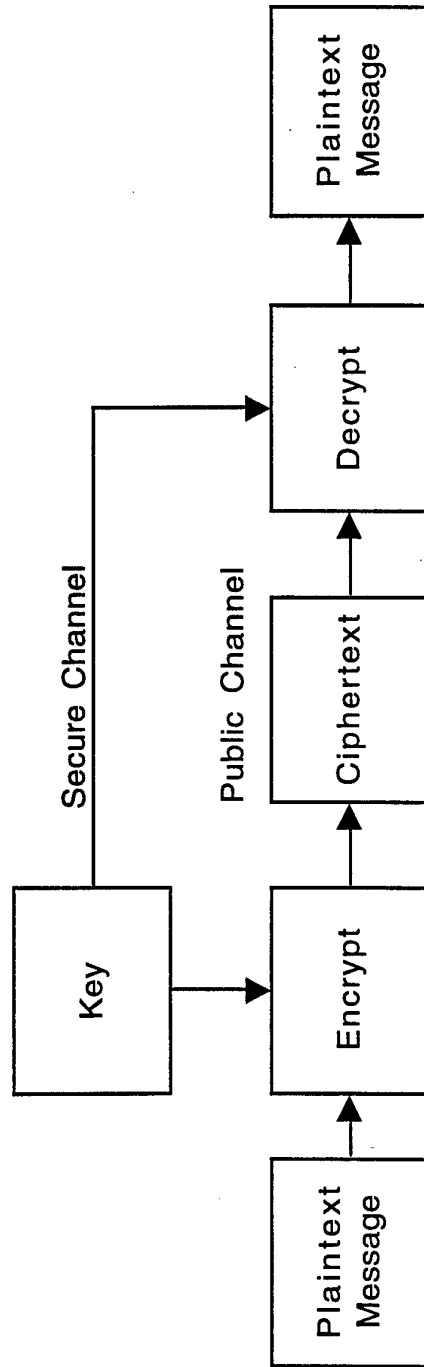
5 16. The method according to Claim 15, wherein following the step
6 of decrypting the imported ciphertext if such an initialization
7 vector match is found, the method further comprises the step of
8 generating a pseudorandom number from the imported
9 initialization vector and using the generated pseudorandom
10 number to decrypt the imported ciphertext, if no initialization
11 vector match is found in the searching step.
12

13 17. The method according to Claim 15, wherein following the step
14 of storing unique pairs of initialization vectors and pseudorandom
15 numbers, the method further comprises the step of importing the
16 ciphertext in concatenated combination with an initialization
17 vector.
18

19 18. The method according to claim 17 further including the step of
20 separating the ciphertext from the initialization vector in the
21 concatenated combination.
22

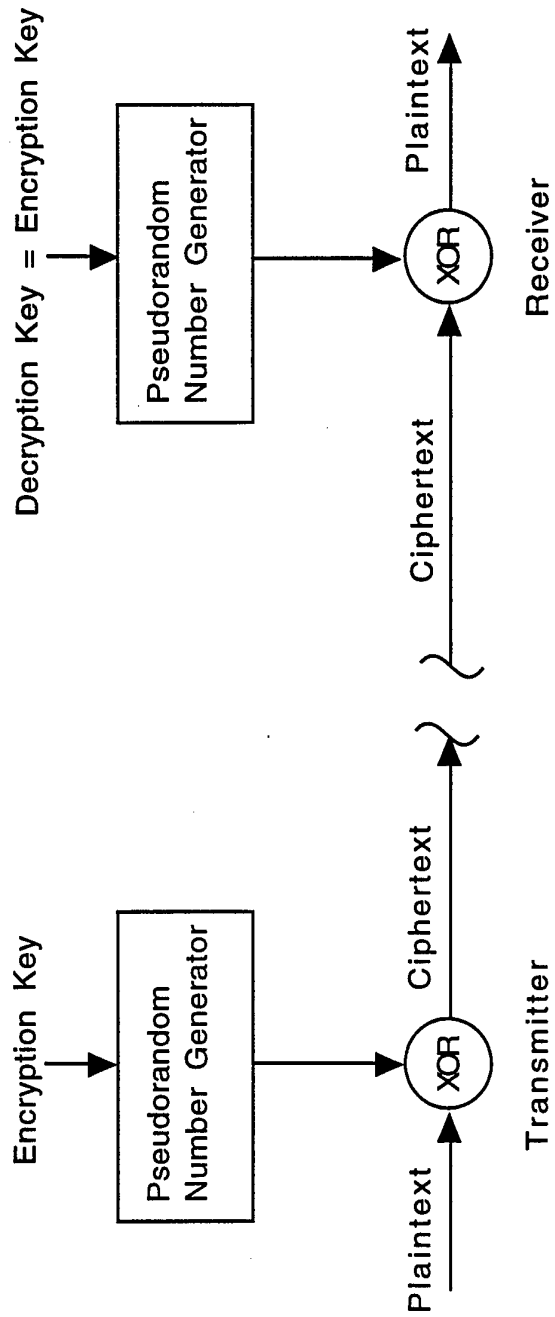
23 19. The method according to Claim 16, wherein the step of
24 generating a pseudorandom number from the imported
25 initialization vector further comprises the step of storing the
26 generated pseudorandom number and its corresponding
27 initialization vector in storage for future use in decrypting.
28

29 20. The method according to Claim 15, wherein the step of
30 decrypting the imported ciphertext if an initialization vector match
31 is found, further comprises the step of logically combining the
32 imported ciphertext with the stored pseudorandom sequence.



Prior Art

FIG. 1(a)



Prior Art

FIG. 1(b)

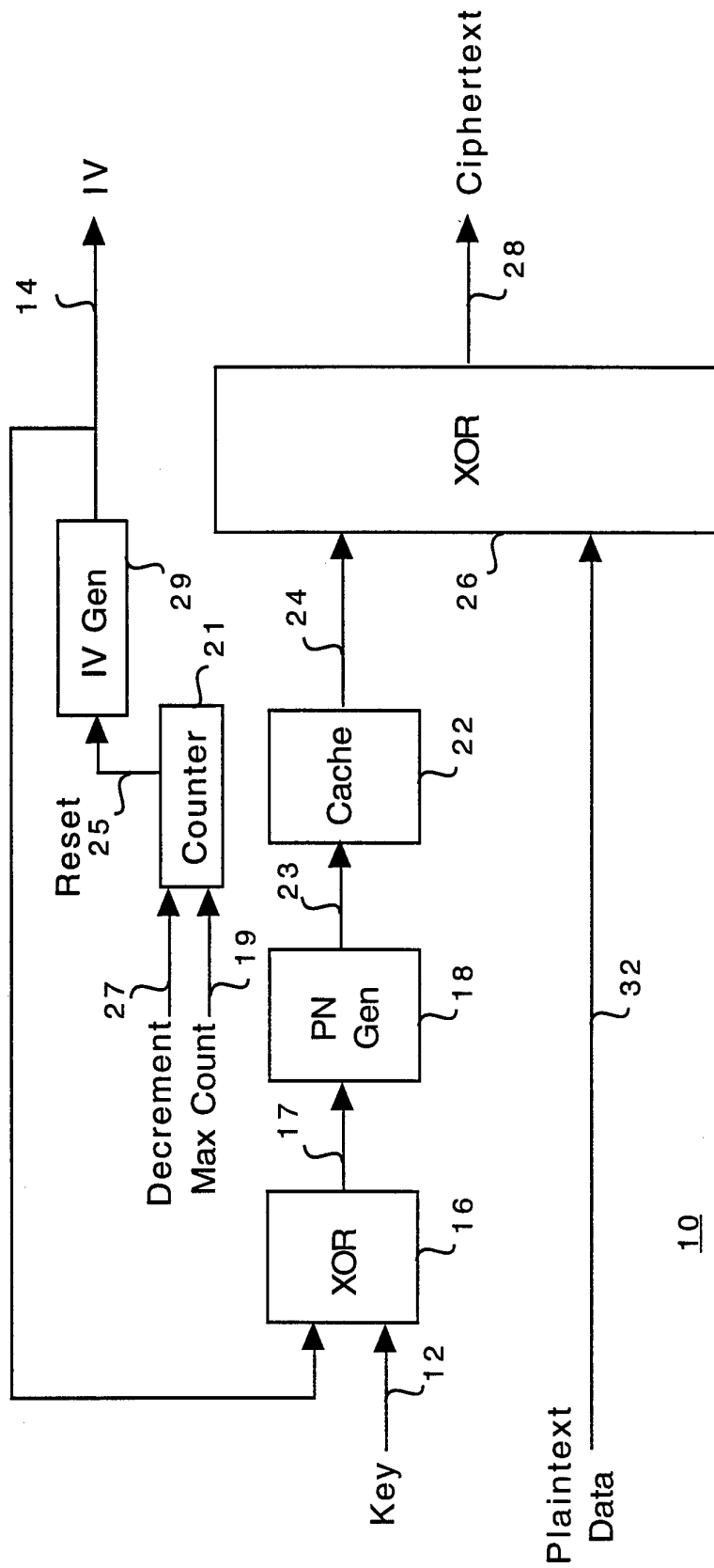


FIG. 2

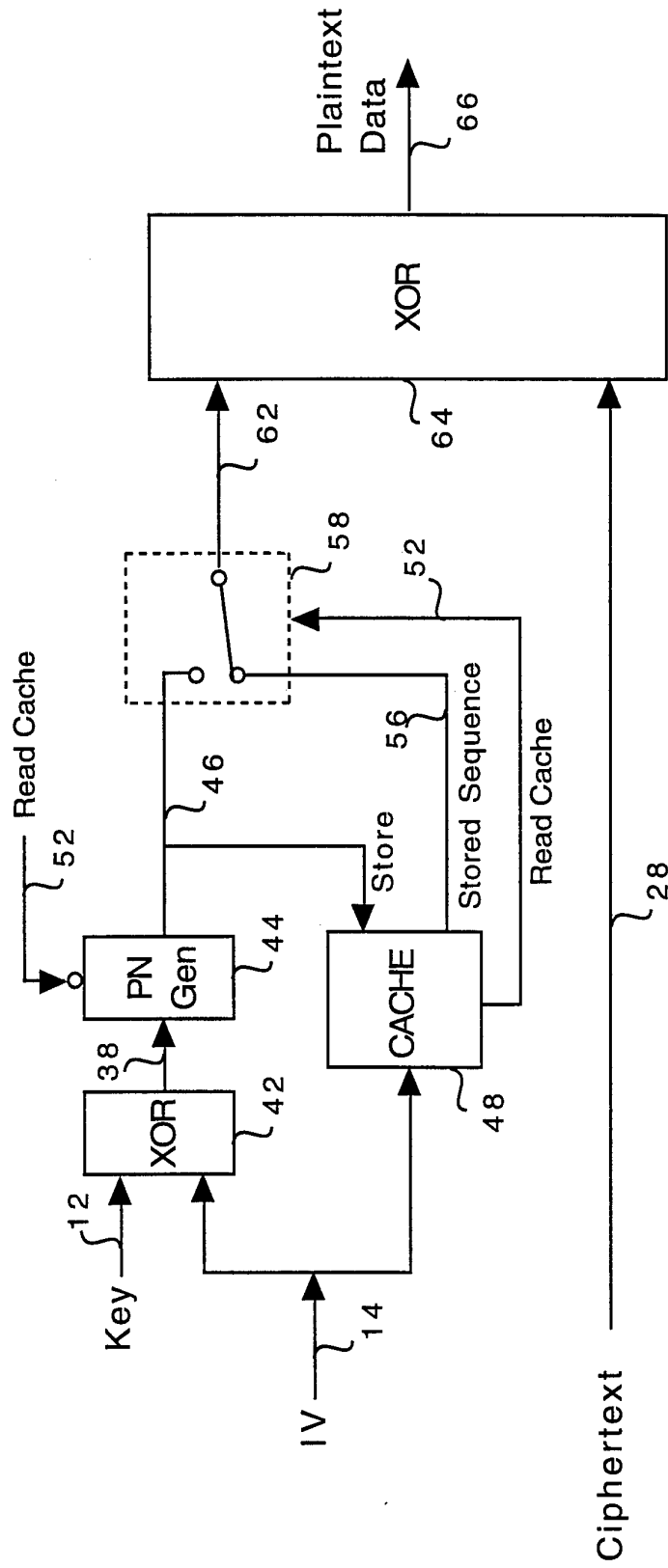


FIG. 3

20

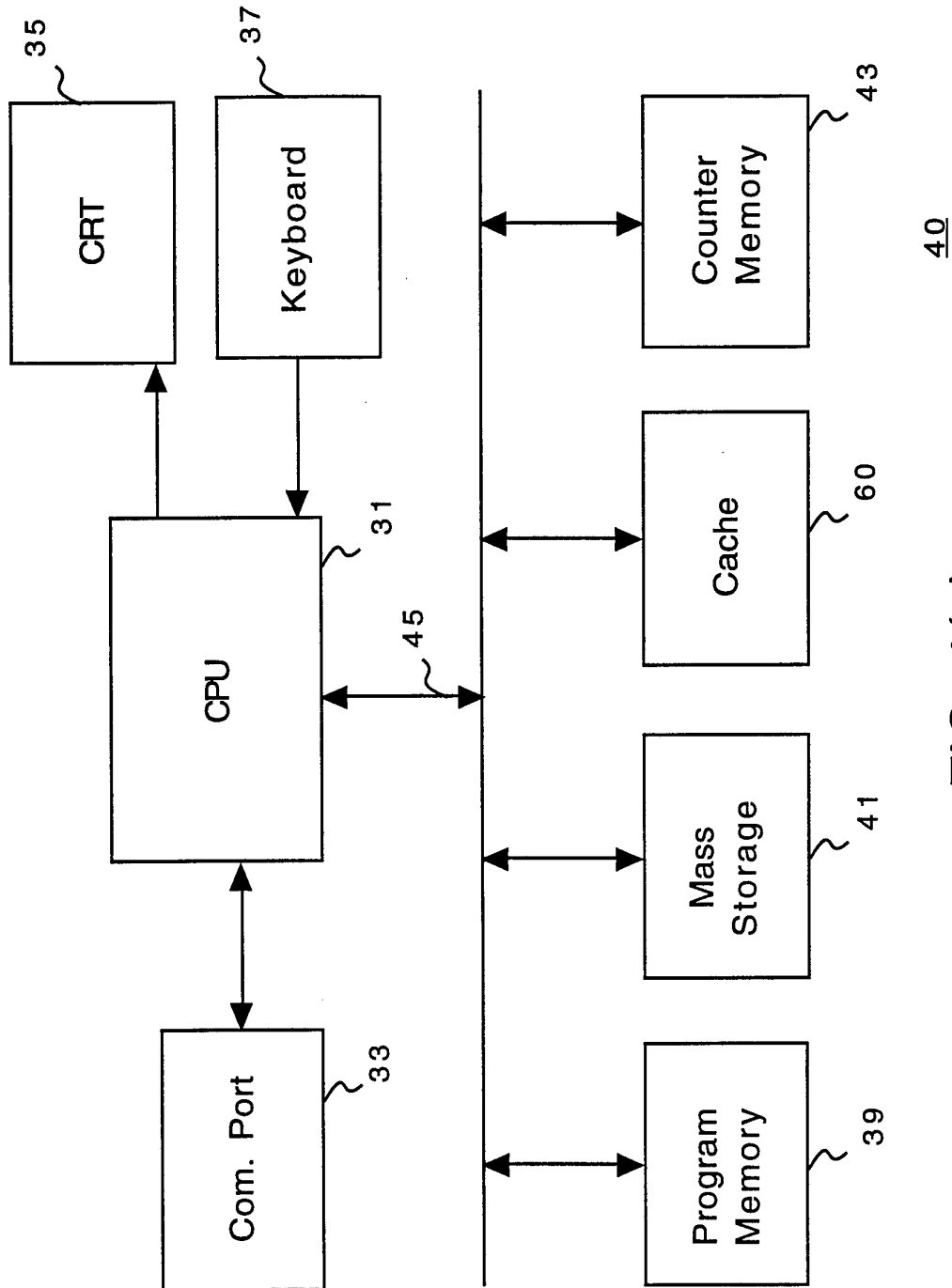


FIG. 4(a)

IV	Sequence
IV 1	Sequence 1
IV 2	Sequence 2
IV n-1	Sequence n-1
IV n	Sequence n

60

72 74

FIG. 4(b)

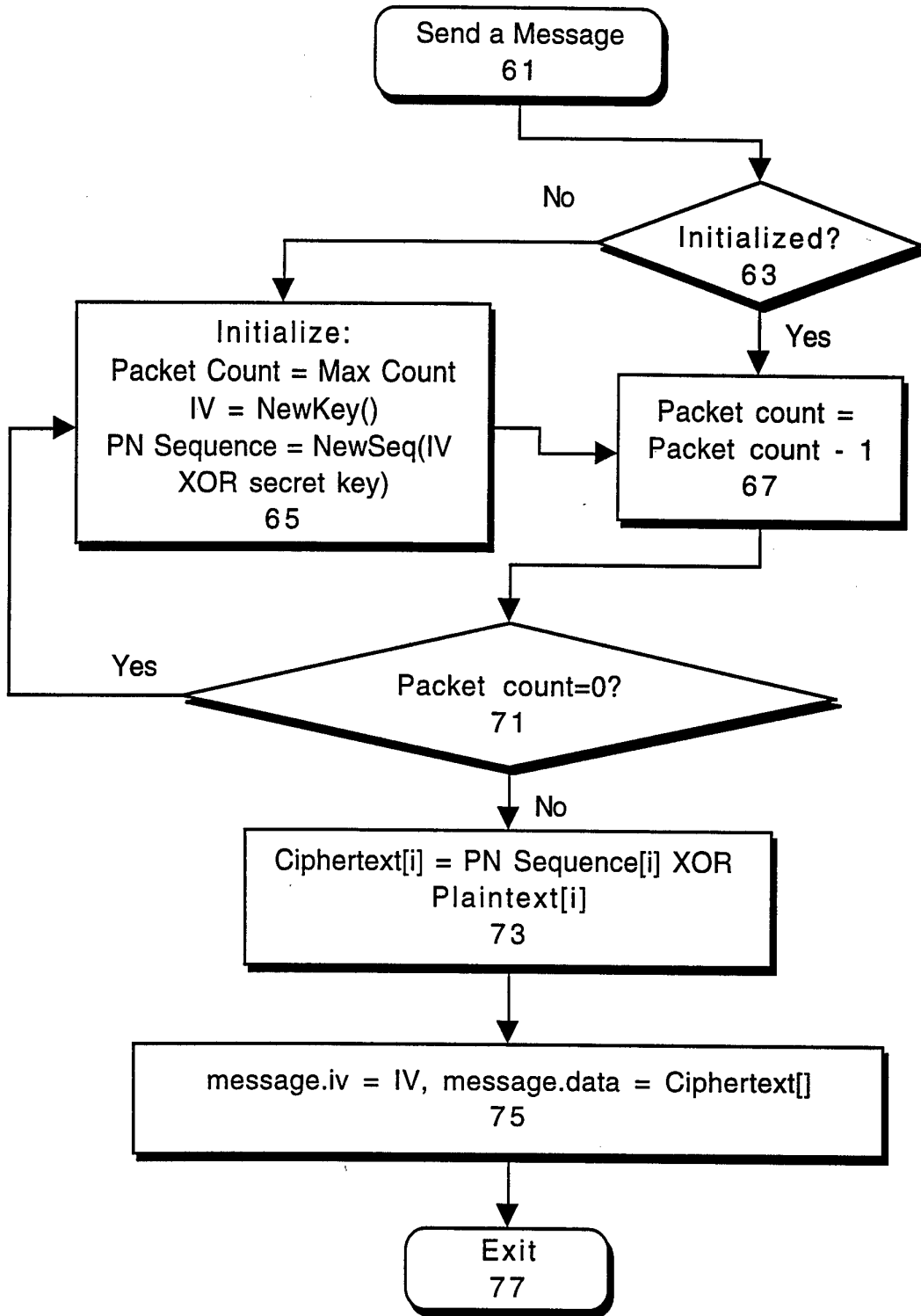


FIG. 5

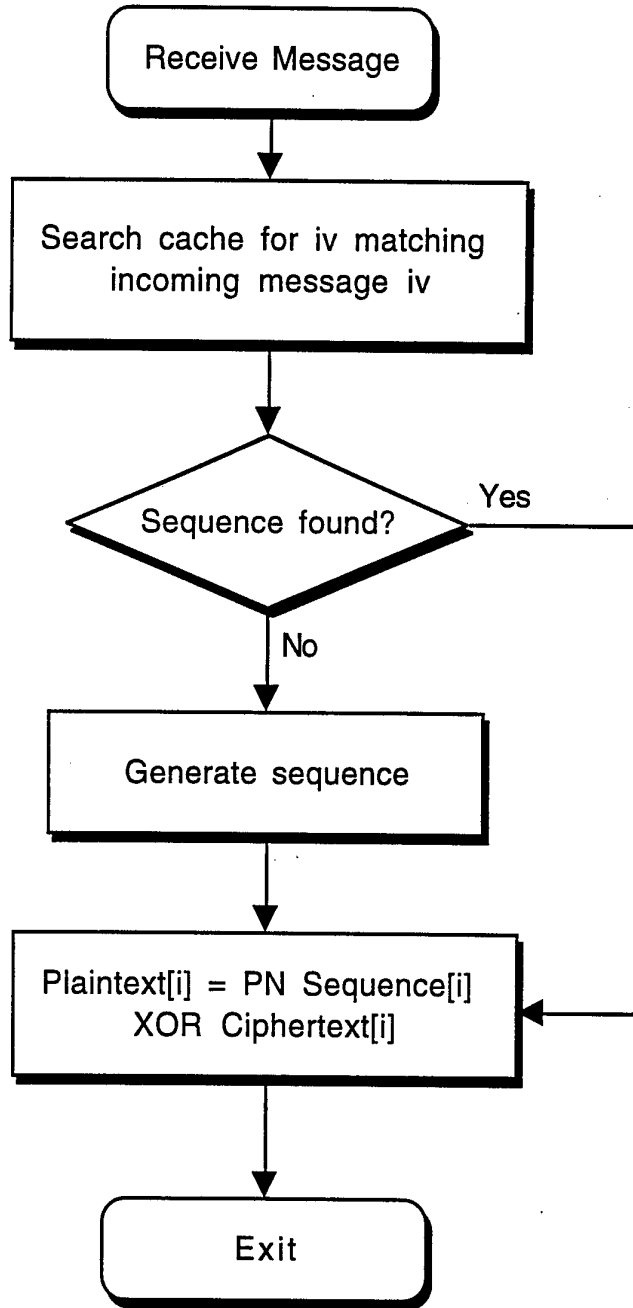


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 94/09509

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 H04L9/18 H04L9/22 H04L9/12		
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) IPC 6 H04L		
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 practical, 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.
A	DE,A,35 18 462 (STANDARD ELEKTRIK LORENZ) 27 November 1986 see abstract see page 4, line 14 - page 6, line 8 see figure ---	1
A	EP,A,0 093 525 (BRITISH TELECOMMUNICATIONS) 9 November 1983 see page 5, line 1 - page 11, line 24 see figures 1-4 -----	8,15
<input type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but 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 22 December 1994		Date of mailing of the international search report 13. 01. 95
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer Lydon, M

INTERNATIONAL SEARCH REPORT

information on patent family members

International application No.

PCT/US 94/09509

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE-A-3518462	27-11-86	US-A- 4757535	12-07-88
EP-A-0093525	09-11-83	CA-A- 1209664	12-08-86
		JP-A- 58202644	25-11-83