OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(12) (19) (CA) **Demande-Application**

(21) (A1) **2,312,358**
(86) 1998/11/25
(87) 1999/06/10

(72) GEORGIADES, JEAN, DE
(71) SIEMENS AKTIENGESELLSCHAFT, DE

(51) Int.Cl.[6] G06F 12/14, H04L 9/08
(30) 1997/12/01 (197 53 274.8) DE
(30) 1998/01/19 (198 01 776.6) DE

(54) **PROCEDE PERMETTANT DE REDUIRE L'ENCOMBREMENT DE MEMORISATION POUR UNE PREMIERE CLE ELECTRONIQUE ET DISPOSITIF DE CODAGE ET DE DECODAGE**

(54) **METHOD FOR REDUCING STORAGE SPACE REQUIREMENTS FOR A FIRST ELECTRONIC KEY AND CODING/DECODING ARRANGEMENT**

(57) Pour réduire l'encombrement de mémorisation pour une clé secrète, celle-ci est subdivisée en blocs, les blocs sont permutés et un index connecté à la permutation est conservé en mémoire. L'index est considérablement raccourci par rapport à la clé. Inversement, la clé secrète peut être retrouvée à partir de l'index en déterminant la permutation dans l'index. La clé secrète est déterminée au moyen des blocs permutés non maintenus secrets et de la permutation. L'invention concerne en outre un dispositif, par exemple une carte à puce, destiné à effectuer le codage et le décodage.

(57) In order to save the storage space required for a secret key, the latter is subdivided into blocks, the blocks are then permutated and a permutation-linked index is stored. The index is significantly shortened in relation to the secret key. On the other hand, the secret key can be recovered from the index by determining the permutation in the index. The secret key is determined by means of the non-secret permutated blocks and the permutation. The invention also relates to an arrangement, i.e. a chip card, for coding and decoding.

PCT

WELTORGANISATION FÜR GEISTIGES EIGENTUM
Internationales Büro

INTERNATIONALE ANMELDUNG VERÖFFENTLICHT NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT)

(51) Internationale Patentklassifikation 6 : G09C

A2

(11) Internationale Veröffentlichungsnummer: WO 99/28887

(43) Internationales Veröffentlichungsdatum: 10. Juni 1999 (10.06.99)

(21) Internationales Aktenzeichen: PCT/DE98/03470

(22) Internationales Anmeldedatum: 25. November 1998 (25.11.98)

(30) Prioritätsdaten:

197 53 274.8	1. Dezember 1997 (01.12.97)	DE
198 01 776.6	19. Januar 1998 (19.01.98)	DE

(71) Anmelder *(für alle Bestimmungsstaaten ausser US)*: SIEMENS AKTIENGESELLSCHAFT [DE/DE]; Wittelsbacherplatz 2, D–80333 München (DE).

(72) Erfinder; und
(75) Erfinder/Anmelder *(nur für US)*: GEORGIADES, Jean [GR/DE]; Ungererstrasse 68 A, D–80805 München (DE).

(74) Gemeinsamer Vertreter: SIEMENS AKTIENGESELLSCHAFT; Postfach 22 16 34, D–80506 München (DE).

(81) Bestimmungsstaaten: CA, JP, US, europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Veröffentlicht
Ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts.

(54) Title: METHOD FOR REDUCING STORAGE SPACE REQUIREMENTS FOR A FIRST ELECTRONIC KEY AND CODING/DECODING ARRANGEMENT

(54) Bezeichnung: VERFAHREN ZUR REDUZIERUNG VON SPEICHERPLATZBEDARF FÜR EINEN ELEKTRONISCHEN ERSTEN SCHLÜSSEL UND ANORDNUNG ZUR VER– UND ENTSCHLÜSSELUNG

(57) Abstract

In order to save the storage space required for a secret key, the latter is subdivided into blocks, the blocks are then permutated and a permutation–linked index is stored. The index is significantly shortened in relation to the secret key. On the other hand, the secret key can be recovered from the index by determining the permutation in the index. The secret key is determined by means of the non–secret permutated blocks and the permutation. The invention also relates to an arrangement, i.e. a chip card, for coding and decoding.

(57) Zusammenfassung

Um Speicherplatzbedarf für einen geheimen Schlüssel einzusparen wird dieser in Blöcke unterteilt, die Blöcke werden permutiert und ein mit der Permutation verknüpfter Index wird abgespeichert. Der Index ist gegenüber dem geheimen Schlüssel signifikant verkürzt. Umgekehrt wird aus dem Index der geheime Schlüssel wiedergewonnen, indem die Permutation aus dem Index ermittelt wird und mittels der nicht geheimzuhaltenden permutierten Blöcken und der Permutation der geheime Schlüssel ermittelt wird. Ferner wird eine Anordnung, z.B. eine Chipkarte, zur Durchführung der Ver– und Entschlüsselung angegeben.

GR 97 P 2979

Description

Method for reducing memory space requirement for an electronic first key, and arrangement for encryption and decryption

The invention relates to a method for reducing memory space requirement for an electronic first key, and to an arrangement for encryption and decryption.

A "key" is understood to mean data which is to be kept secret and is to be used in a cryptographic method, in particular.

A "hacker" is an unauthorized person aiming to get hold of the key.

Numerous cryptographic methods are known, inter alia symmetrical and asymmetrical methods. Particularly in a computer network, but also to an increasing extent in portable media, e.g. a mobile telephone or a smart card, it is necessary to ensure that a stored key cannot be accessed even if a hacker has taken control of the computer, the mobile telephone or the smart card.

To ensure adequate security of cryptographic methods, keys are determined, particularly in asymmetrical methods, which each have lengths of a plurality of 100 bits. A computer's or portable medium's memory area protected against hacking, that is to say a memory area which cannot be read by a hacker, is usually of small proportions. A length of a plurality of 100 bits for a key stored in such a protected memory area reduces free memory space within the protected memory area, which means that relatively few such keys can be stored together.

The object of the invention is to specify a method for reducing the memory space requirement for an electronic key, and to specify an arrangement for encryption and decryption, avoiding the disadvantage described above.

This object is achieved in accordance with the features of the independent patent claims.

The invention specifies a method for reducing memory space requirement for an electronic first key having a predetermined length, in which the first key contains a number of units which corresponds to the predetermined length. A plurality of units are combined into a respective block. The first key is represented by a plurality of blocks, and a publicly accessible identifier is produced from the first key by ascertaining a permutation of the blocks and storing an index for this permutation as an electronic second key.

The index characterizes that permutation according to which the blocks in the first key have been interchanged. For this, a first place in the index can contain a number which describes the place to which the first block in the first key has been shifted as a result of the permutation. Similarly, the second place in the index refers to that place (a block corresponds to a place) in the public identifier which represents the second block in the first key, and so on. Continuation of this method until all the blocks have been allocated produces a permutation number (PERM, see Figure 3) which references the first key unambiguously from the publicly accessible identifier.

The index can also be a shortened form of the permutation number. If n denotes the number of blocks, then (if

all the blocks are different on a paired basis) there are n! possibilities for the arrangement of the blocks. Blocks 1 to n can be arranged differently. If the possibilities for the arrangement are sorted on the basis of a predetermined scheme (e.g. in a table) according to size, with each block receiving a digit according to its position, then the index can be determined by searching for the permutation number's entry and storing its location within the arrangement (with a reproducible sequence) as an index (see example in Figure 3 and Table 1).

In this regard, it should be noted that, for illustrative purposes, an arrangement in a table is assumed, whereas in practice an arrangement based on a particular scheme is preferable, since, by way of example, if n = 50, the table would have to have approximately $3*10^{64}$ entries. An example of an arrangement based on a particular scheme is explained in more detail in the description of the figures.

The memory space requirement for the first key is reduced as a result of the second key now being stored. The second key comprises the index and therefore requires significantly less memory space than the first key. The memory space requirement for the second key is preferably determined by the number of possible permutations. The identifier produced can be made publicly accessible, and, accordingly, does not need to be stored in a protected memory area. A hacker learning of this identifier has, even if he knows the block size, n! possibilities (n is the number of blocks) for ascertaining the first key from the identifier. In practice, such an attempt is extremely unlikely to succeed.

One development is that the blocks each comprise an identical number of units.

Another development is that one of the following units is used:

a) number;

b) alphanumeric character;

c) byte;

d) bit.

Another development is that the second key is stored in a protected memory area, preferably on a smart card.

In this context, it is advantageous that the second key has a markedly reduced length as compared with the first key, and thus places lower requirements on the memory space requirement within the protected memory area. The protected memory area ensures that data contained in it cannot be read easily by a hacker. Since available memory space is generally low in the protected memory area, it is a significant advantage if the length of a key to be stored in the protected memory area is reduced without also having to accept reduced security of the cryptographic method.

Within the context of an additional development, the permutation of the blocks is ascertained by carrying out the following steps: a permutation is determined at random from all the possible permutations, the permutations being produced on the basis of a particular scheme, and a sequence of the permutations produced thus being reproducible. The index is then used to ascertain a location for the permutation within the sequence of the permutations.

This index thus represents a reproducible (using the convention (= predetermined scheme) to form the permutations) map of the first key. On the basis of the randomly ascertained permutation, the blocks are

interchanged and stored as a publicly accessible identifier.

In one development of the invention, a third key is ascertained using the second key, this third key being identical to the first key. To this end, the public identifier is divided into blocks, each block comprising a plurality of units in the identifier. A third key is ascertained from the identifier and the second key by producing all the possible permutations of the blocks, whose sequence is reproducible, and using the second key to ascertain that permutation amongst the permutations which represents the third key.

This is essentially equivalent to the inverse operation for forming the permutation of the blocks, whose sequence is reproducible. The index (second key) is used for addressing within the sequence of permutations, so that the associated permutation having the publicly accessible identifier defines a third key, which is identical to the first key.

It is also a development that the second key is a secret key and is stored in a protected memory area, e.g. a smart card.

The invention also specifies an arrangement for encryption and for decryption, with a medium which has a protected memory area and with an arithmetic and logic unit which is set up such that a first key is shortened in accordance with the steps of the method illustrated above.

The medium is preferably a portable medium, e.g. a smart card. The protected memory area can be arranged both on the medium and within a computer,

which, by way of example, is combined in a network interconnection with other computers. In this arrangement, the protected memory area is intended to be sufficiently safe from unauthorized access. This is ensured by suitable mechanisms which, by way of example, prevent the protected memory area from being read and use internal mechanisms, detected by a computer accessing the protected memory area, to inform the outside merely of a result of a comparison operation with an entry, particularly a key, in the protected memory area.

Developments of the invention are revealed in the dependent claims.

Illustrative embodiments of the invention are illustrated and explained below with the aid of the drawings, in which

Figure 1 shows a sketch illustrating a method for reducing memory space requirement for an electronic first key,

Figure 2 shows a sketch illustrating a method for restoring the first key from the publicly accessible identifier and the second key,

Figure 3 shows an example of a method for reducing memory space requirement for a secret key,

Figure 4 shows an arrangement for encryption and for decryption,

Figure 5 shows an arithmetic and logic unit.

Figure 1 shows a method for reducing memory space requirement for an electronic first key. In a step 101, the first key is divided into blocks, with the blocks each containing an identical number of units. Such units are preferably numbers, alphanumeric characters, bytes or bits. In a step 102, a permutation is ascertained at random from all the permutations of the blocks, whose sequence is reproducible. This random permutation is used as a publicly accessible identifier. The sequence of the blocks which corresponds to this permutation is extremely unlikely to be used to restore the first key if the first key is provided so as to have a suitable number of units. The selected permutation has a particular location (see step 103) within the sequence of all the permutations (sequence reproducible). In a step 104, the index is stored as a second key. The second key is stored in a protected memory area.

Figure 2 shows steps in a method for restoring the first key from the publicly accessible identifier with the second key. For this, permutations, whose sequence is reproducible, of the blocks in the identifier are ascertained from the identifier in a step 201. The second key is used to ascertain one permutation amongst the permutations as a third key (see step 202). The third key is identical to the first key (see step 203). This means that the first key, which was mapped in a second key in order to reduce memory space, is restored.

The example illustrated below explains how the method for reducing the

memory space requirement for an electronic key works. Figure 3 illustrates the correlations.

The first key K1 "1234567890" comprises a plurality of units EINH "1", "2", "3", "4", "5", "6", "7", "8", "9", "0" each represented using an alphanumeric character. In a step 301, the first key K1 is divided into blocks BL "1 2", "3 4", "5 6", "7 8", "9 0", which each comprise two units EINH. A subsequent step 302 determines a random combination of the blocks BL to produce an identifier KEN "3478129056", which can be publicly accessible. A step 303 produces, on the basis of the permutation of the first key K1 and of the identifier KEN, a permutation number PERM "24153", which converts the identifier KEN unambiguously to the first key K1.

The permutation number PERM maps the identifier KEN to the first key in that the first place in the permutation number PERM "2" represents the first place in the identifier KEN "3 4", which is based on two-unit blocks, and has this block as the second block in the first key K1. The second place in the identifier KEN "7 8" is accordingly the fourth place in the first key K1, and so on. When allocation is complete, the permutation number unambiguously gives the first key K1 as "1234567890".

A representation of the first key K1 which is significantly shorter than the length of the permutation number PERM is produced by a further allocation. In a step 304, the sequence of all the permutations having the same number of places as the permutation number PERM is used to ascertain from the permutation number PERM a location in this sequence, with the aid of Table 1. Table 1 comprises an extract from the total number of possibilities for the arrangement of the blocks BL.

$n=5 \Rightarrow n! = 120$ possibilities for the arrangement of the blocks

If the possible permutations are sorted according to size, an unambiguous sequence of all the permutations (from 0 to n!-1) is produced (see Table 1 as an extract of the first 47 possibilities).

000	12345	024	21345
001	12354	025	21354
002	12435	026	21435
003	12453	027	21453
004	12534	028	21534
005	12543	029	21543
006	13245	030	23145
007	13254	031	23154
008	13425	032	23415
009	13452	033	23451
010	13524	034	23514
011	13542	035	23541
012	14235	036	24135
013	14253	**037**	**24153**
014	14325	038	24315
015	14352	039	24351
016	14523	040	24513
017	14532	041	24531
018	15234	042	25134
019	15243	043	25143
020	15324	044	25314
021	15342	045	25341
022	15423	046	25413
023	15432	...	...

Table 1

Step 305 uses the table LISTE to determine the entry referencing the permutation number PERM in the table LISTE. As Table 1 shows, the 37th entry in Table 1 (LISTE in Figure 3) equates to the permutation number PERM. Accordingly, the second key K2 stored is the 37th entry, that is to say the character sequence "037". The length of the second key K2 is markedly reduced as compared with the first key K1. The second key K2 is preferably stored in a protected

memory area. The size of the second key K2 is determined by the number of possible permutations. If n is the number of blocks BL into which the first key K1 is divided, then the number of possibilities is given as "n!". In this example, there are 5 blocks, that is to say 120 possibilities. In decimal notation, three places ("000" to "119") are required as the second key K2, whereas in binary notation only 7 bits are required.

If the first key K1 comprises a plurality of 100 bits, then it is reasonable that, even given a combination into blocks of a plurality of bits, there is still a large number of blocks, and the factorial of this number is equivalent to the possibilities which the hacker would have to try out in order to arrive at the first key. The likelihood of such an occurrence is small.

Table 1 serves predominantly to illustrate the basic procedure. In practice, the number of blocks n is usually large, so that the allocation (indicated by Table 1) described preferably takes place on the basis of a particular scheme. Such a scheme will be explained below.

ALLOCATION OF THE PERMUTATION TO THE IDENTIFIER:

The following abbreviations are used:

n	number of blocks,
PERM	permutation,
L	list containing blocks in the secret key which have not yet been evaluated, initialization with L=(1,2,3,...n),

k:=0

The convention for allocating the permutation to the identifier is as follows:

```
FOR s=1 TO n-1 DO
    1.  Read s-th number in the permutation
    2.  T = position of read number in the list L
    3.  k = k+(T-1)*(n-s)!
    4.  Delete T-th position from the list L,
        advance subsequent positions by one place
    5.  s = n-1 ?
            YES? => program end
Result: k = identifier
```

The correlation is illustrated below with the aid of an example:

```
n = 5
PERM = 2,4,1,5,3
L=(1,2,3,4,5)
k=0
```

1st step:

s = 1, n-s = 4, (n-s)! = 24, k = 0, L = (1,2,3,4,5)

1. s-th number in the permutation: 2
2. T = 2
3. k = 0 + (2-1) * 24 = 24
4. L = (1,3,4,5)

2nd step:

s = 2, n-s = 3, (n-s)! = 6, k = 24, L = (1,3,4,5)

1. s-th number in the permutation: 4
2. T = 3
3. k = 24 + (3-1) * 6 = 36
4. L = (1,3,5)

3rd step:

s = 3, n-s = 2, (n-s)! = 2, k = 36, L = (1,3,5)

1. s-th number in the permutation: 1

2. T = 1
3. k = 36 + (1-1) * 2 = 36
4. L = (3,5)

4th step:

s = 4, n-s = 1, (n-s)! = 1, k = 36, L = (3,5)

1. s-th number in the permutation: 5
2. T = 2
3. k = 36 + (2-1) * 1 = 37
4. L = (3)
5. s = n-1 = 4? => YES! => program end

Result: Identifier = k = 37.

ALLOCATION OF THE IDENTIFIER TO THE PERMUTATION:

The following abbreviations are used:

n	number of blocks,
k	identifier,
L	list containing positions in the secret key which have not yet been identified, initialization with L=(1,2,3,...n).

The convention for allocating the identifier to the permutation is as follows:

FOR s=1 TO n-1 DO

1. Divide k/(n-s)!
2. E = integer result of division into 1.
3. R = remainder of division into 1.
4. The s-th block in the permutated key is that block in the secret key which is in the (E+1)th position in the list L

5. the (E+1)th entry in the list L is deleted, subsequent entries advance by one position
6. k assumes the value R
7. s = n-1?
   YES? => the n-th block of the permutated key is then that block in the secret key which has still remained in the list L => program end.

The correlation is illustrated below with the aid of an example (as above example for Table 1: identifier 37 => permutation 24153):

n = 5
k = 37

1st step:
s = 1, k = 37, (n-s)! = (5-1)! = 24, L = (1,2,3,4,5)
1. 37/24 = 1, remainder 13
2. E = 1
3. R = 13
4. the 1st block in the permutated key is that block in the secret key which is in the 2nd position in the list L = (1,2,3,4,5)
5. the 2nd entry in the list L is deleted, subsequent entries advance by one position => L = (1,3,4,5)
6. k = 13

2nd step:
s = 2, k = 13, (n-s)! = 6, L = (1,3,4,5)
1. 13/6 = 2, remainder 1
2. E = 2
3. R = 1

4. the 2nd block in the permutated key is that block in the secret key which is in the 3rd position in the list L = (1,3,4,5)
5. L = (1,3,5)
6. k = 1

3rd step:

s = 3, k = 1, (n-s)! = 2, L = (1,3,5)

1. 1/2 = 0, remainder 1
2. E = 0
3. R = 1
4. the 3rd block in the permutated key is that block in the secret key which is in the 1st position in the list L = (1,3,5)
5. L = (3,5)
6. k = 1

4th step:

s = 4, k = 1, (n-s)! = 1, L = (3,5)

1. 1/1 = 1, remainder 0
2. E = 1
3. R = 0
4. 4th block in the permutated key is that block in the secret key which is in the 2nd position in the list L = (3,5)
5. L = (3)
6. k = 1
7. s = n-1? => YES! => the 5th block in the permutated key is then that block in the secret key which still remains in the list L = (3) => program end.

Result: the permutation is: 2,4,1,5,3.

Thus, Figure 3 shows that the scheme described can be used to determine both the identifier K2 from the permutation PERM and, conversely, the permutation PERM from the identifier K2. The block LISTE ensures that the location of the permutation PERM within the set of all the permutations of the same length is allocated, with the permutations being sorted according to size.

Figure 4 shows an arrangement for encryption and decryption.

A portable medium 401, preferably a smart card, comprises a (conventional) memory area MEM 403 and a protected memory area SEC 402. Using an interface IFC 404, data is interchanged between the medium 401 and a computer network 406 over a channel 405. The computer network 406 comprises a plurality of computers which are connected to one another and communicate with one another. Data for operating the portable medium 401 is preferably available distributed in the computer network RN 406.

The protected memory area 402 is not designed to be readable. An arithmetic and logic unit accommodated on the portable medium 401 or in the computer network 406 enables the use of the data in the protected memory area 402. Thus, the result of a comparison operation can be whether or not a comparison of an entry with a key in the protected memory area 402 was successful.

Figure 5 shows an arithmetic and logic unit 501. The arithmetic and logic unit 501 comprises a processor CPU 502, a memory 503 and an input/output interface 504, which is used in different ways via an interface 505 routed out of the arithmetic and logic unit 501: a graphics interface is used to display an output on a monitor 507 and/or to output it on a printer 508. An

entry is made using a mouse 509 or a keyboard 510. The arithmetic and logic unit 501 also has a bus 506 which ensures the connection between the memory 503, the processor 502 and the input/output interface 504. It is also possible to connect additional components to the bus 506: additional memory, hard disk etc.

Patent claims

1. A method for reducing memory space requirement for an electronic first key on a mobile memory unit,

a) in which the first key contains a number of units which corresponds to the predetermined length,

b) in which a plurality of units are combined into a respective block,

c) in which the first key is represented by a plurality of blocks,

d) in which a publicly accessible identifier is produced from the first key by ascertaining a permutation for the blocks and storing an index for this permutation as an electronic second key on the mobile memory unit.

2. The method as claimed in claim 1, in which the blocks each contain an identical number of units.

3. The method as claimed in claim 1 or 2, in which one of the following units is used:

a) number,

b) alphanumeric character,

c) byte,

d) bit.

4. The method as claimed in one of the preceding claims, in which the second key is stored in a protected memory area on the mobile memory unit.

AMENDED PAGE

5. The method as claimed in one of the preceding claims, in which the permutation of the blocks comprises the following steps:

a) a permutation is determined at random from all the possible permutations, the permutations being produced on the basis of a particular scheme, and a sequence of the permutations thus being reproducible;

b) the index represents a location of the permutation within the sequence of the permutations.

6. A method for determining a third key using the second key, which has been produced using the method as claimed in one of claims 1 to 5,

a) in which the public identifier is divided into blocks, each block comprising a plurality of units in the identifier,

b) in which the third key is ascertained from the identifier and the second key by producing all the possible permutations of the blocks and using the second key to ascertain that permutation which corresponds to the third key.

7. The method as claimed in one of the preceding claims, in which the mobile memory unit is provided on a smart card.

8. An arrangement for encryption and for decryption,

a) in which a mobile memory unit is provided which has a protected memory area,

b) in which an arithmetic and logic unit is provided which is set up such that a first key is shortened and stored on the mobile memory unit in that

(1) the first key contains a number of units which corresponds to the predetermined length,

(2) a plurality of units are combined into a respective block,

(3) the first key is represented by a plurality of blocks,

(4) a publicly accessible identifier is produced from the first key by ascertaining a permutation for the blocks and storing an index for this permutation as an electronic second key on the mobile memory unit.

9. The arrangement as claimed in claim 8, in which the arithmetic and logic unit is set up such that

a) a permutation is determined at random from all the possible permutations, the permutations being produced on the basis of a particular scheme, and a sequence of the permutations thus being reproducible;

b) the index represents a location of the permutation within the sequence of the permutations.

10. The arrangement as claimed in claim 8 or 9, in which the arithmetic and logic unit is set up such that a third key is determined using the second key, which has been produced using the method as claimed in one of claims 1 to 5, by performing the following steps:

a) the public identifier is divided into blocks, each block comprising a plurality of units in the identifier,

b) the third key is ascertained from the identifier and the second key by producing all the possible permutations of the blocks and using the second key to ascertain that permutation which corresponds to the third key.

11. The arrangement as claimed in one of claims 8 to 10, in which the arithmetic and logic unit is produced on the mobile memory unit.

12. The arrangement as claimed in one of claims 8 to 10, in which the arithmetic and logic unit is a computer in a network comprising at least one computer and the mobile memory unit.

13. The arrangement as claimed in one of claims 9 to 11, in which the mobile memory unit is produced on a smart card.

Fetherstonhaugh & Co.
Ottawa, Canada
Patent Agents

AMENDED PAGE

FIG 1

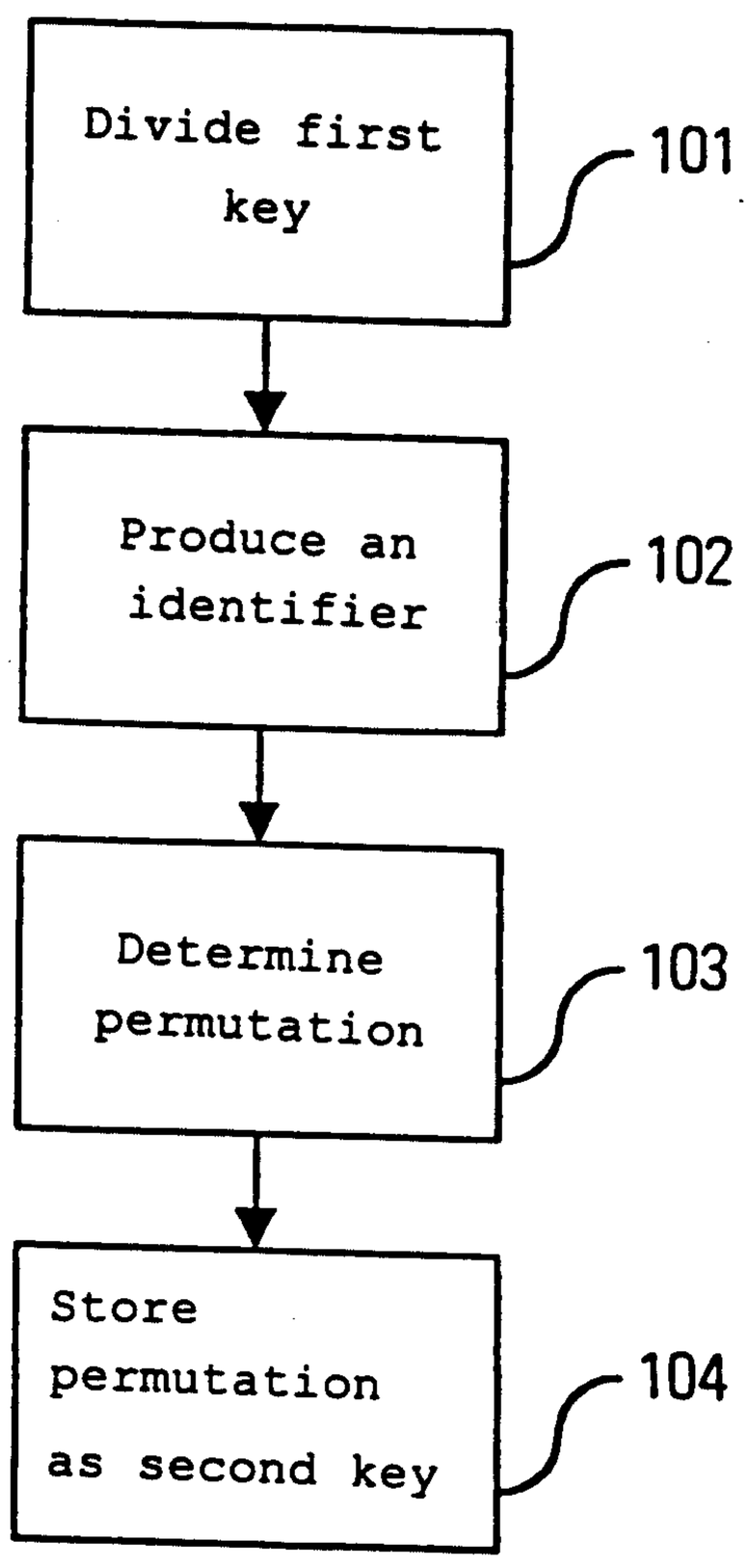

Divide first key
101
Produce an identifier
102
Determine permutation
103
Store permutation as second key
104

FIG 2

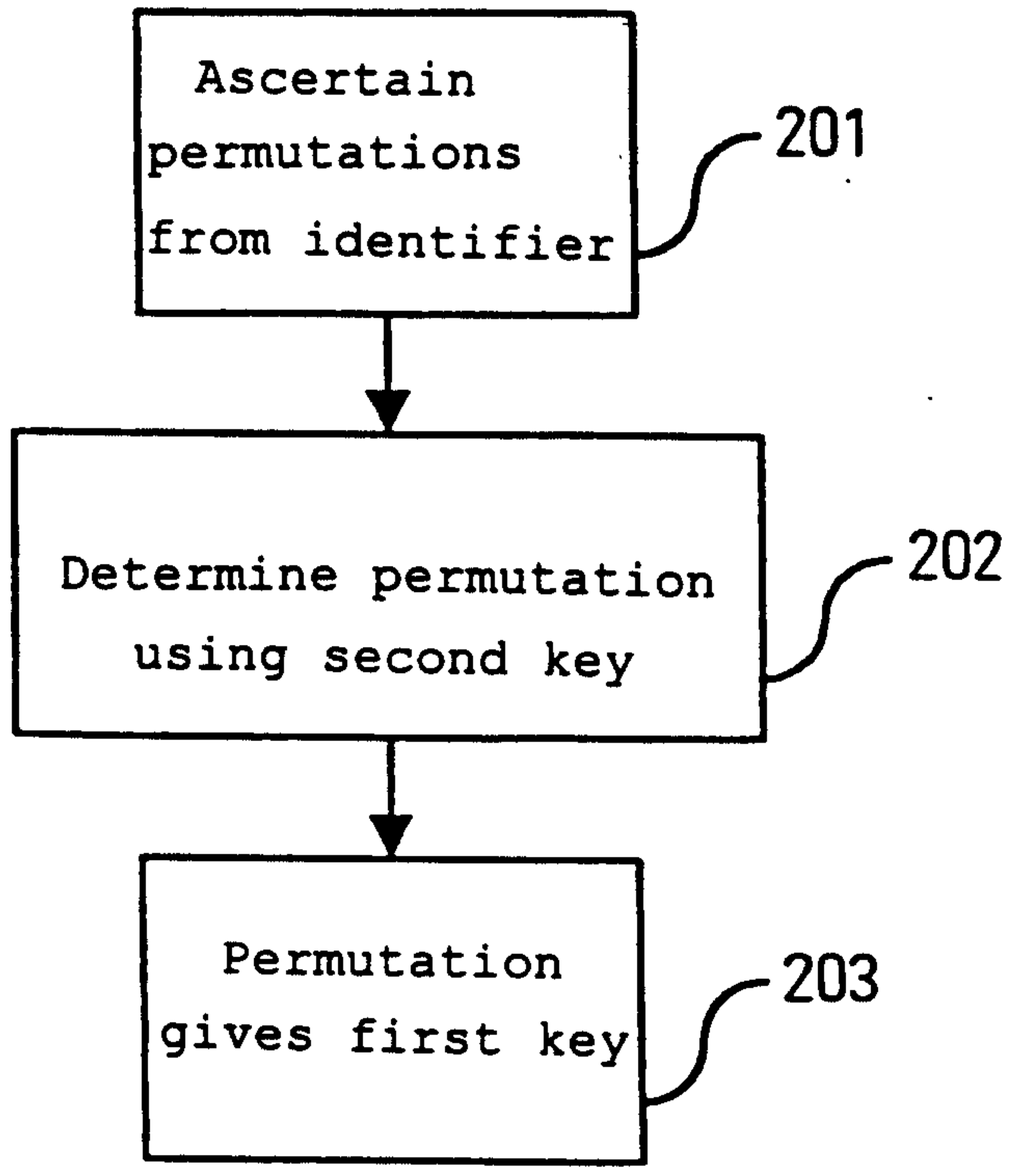

Ascertain permutations from identifier
201
Determine permutation using second key
202
Permutation gives first key
203

FIG 3

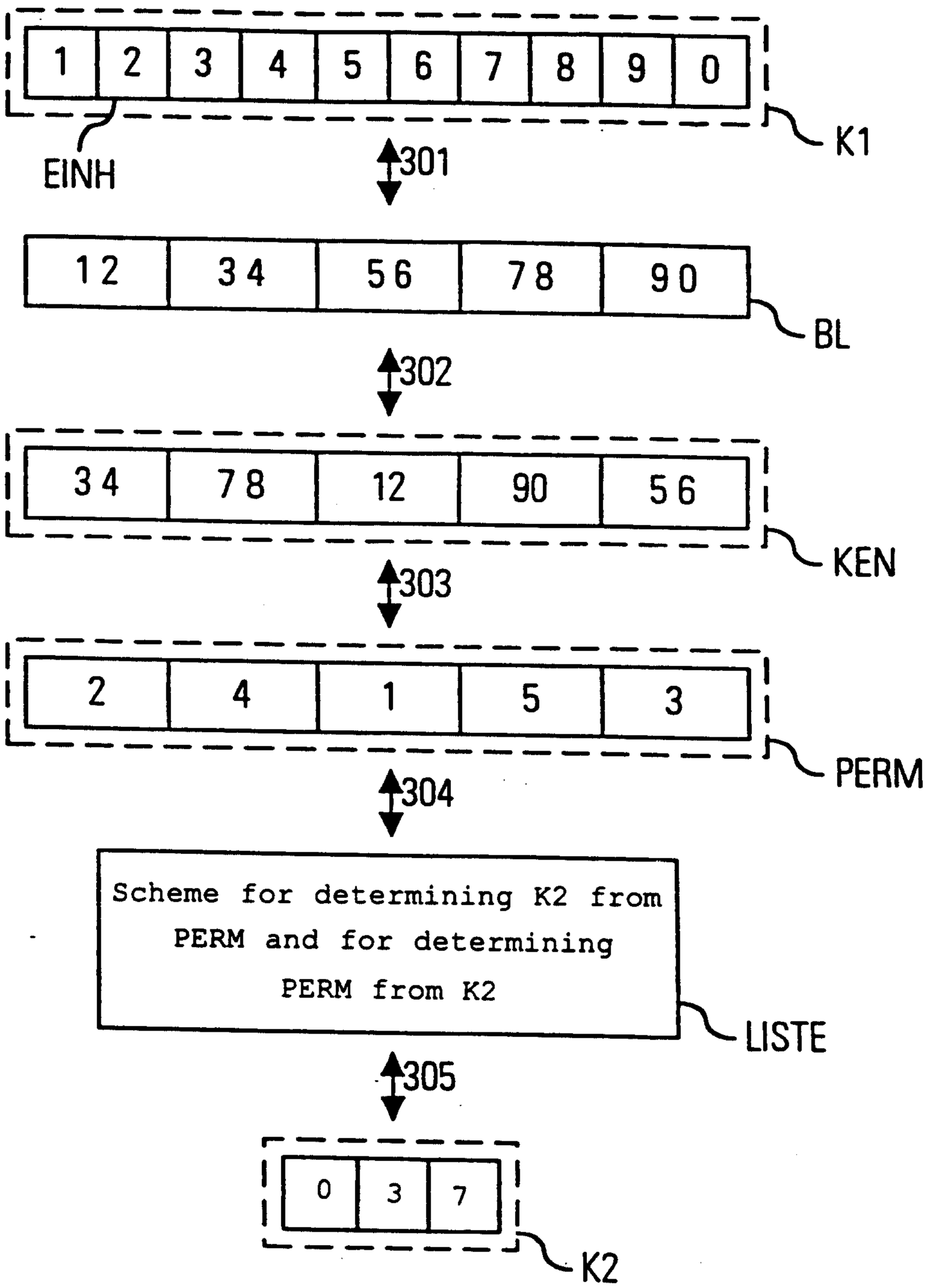

1 2 3 4 5 6 7 8 9 0
K1
EINH
301
12 34 56 78 90
BL
302
34 78 12 90 56
KEN
303
2 4 1 5 3
PERM
304
Scheme for determining K2 from PERM and for determining PERM from K2
LISTE
305
0 3 7
K2

FIG 4

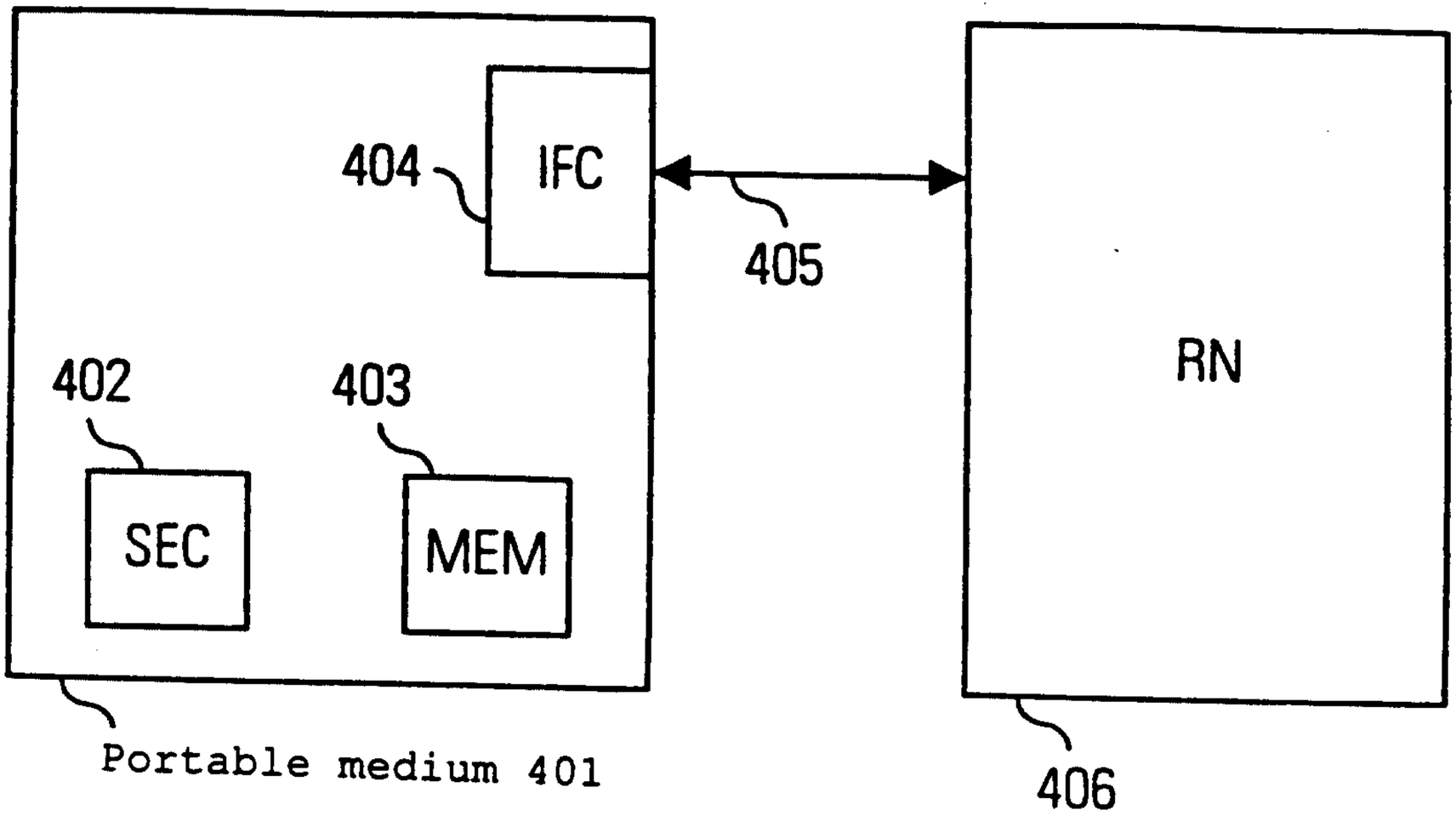
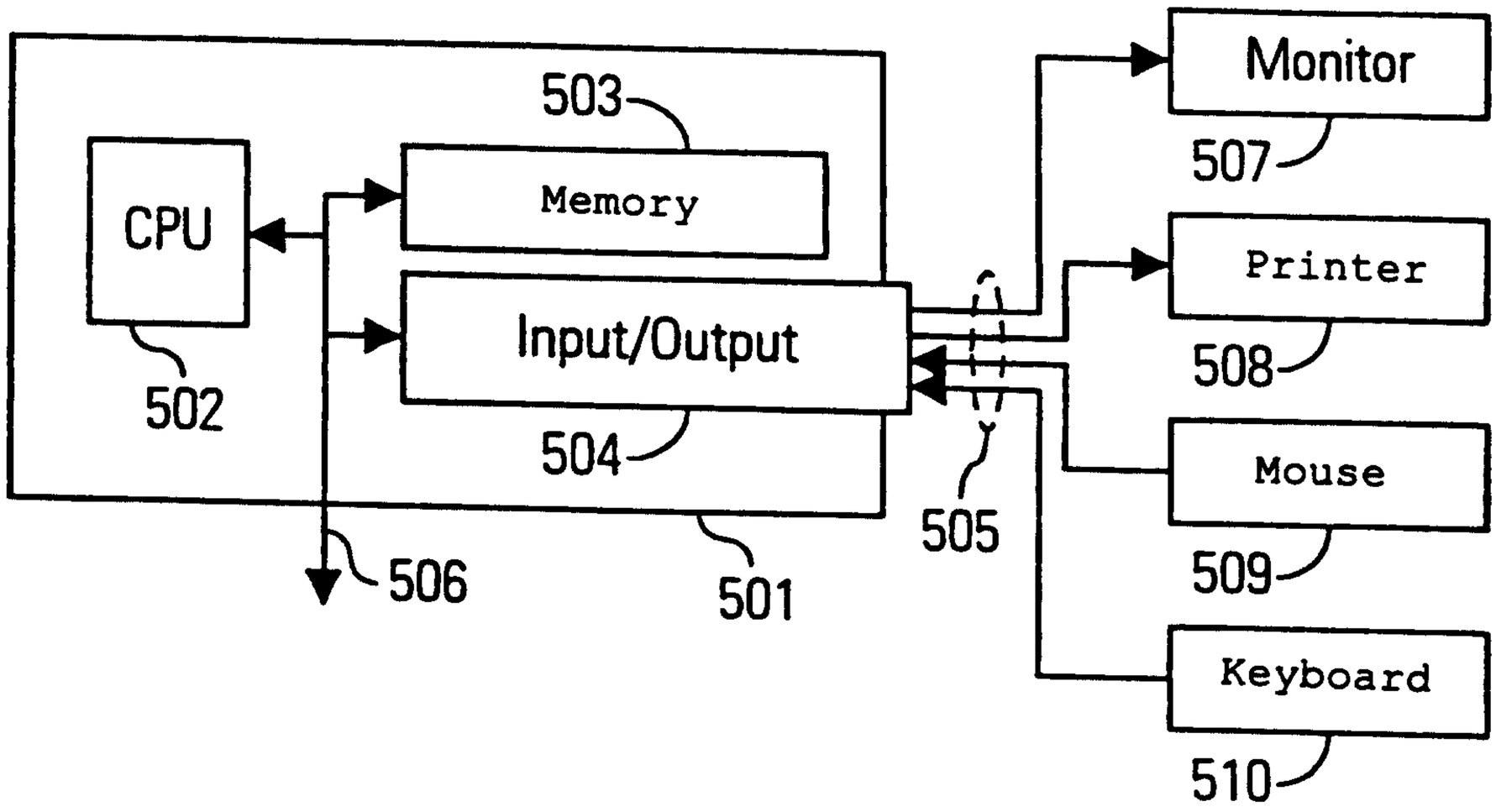

404
IFC
405
RN
402
403
SEC
MEM
Portable medium 401
406

FIG 5


503
CPU
Memory
Input/Output
502
504
506
501
505
Monitor
507
Printer
508
Mouse
509
Keyboard
510