

(19)



(11)

EP 1 364 279 B9

(12)

CORRECTED EUROPEAN PATENT SPECIFICATION

Note: Bibliography reflects the latest situation

(15) Correction information:

Corrected version no 1 (W1 B1)

Corrections, see

Claims EN

(51) Int Cl.:

G06F 7/58 (2006.01)

(86) International application number:

PCT/GB2002/000300

(48) Corrigendum issued on:

17.10.2007 Bulletin 2007/42

(87) International publication number:

WO 2002/063462 (15.08.2002 Gazette 2002/33)

(45) Date of publication and mention
of the grant of the patent:

31.05.2006 Bulletin 2006/22

(21) Application number: **02715561.3**

(22) Date of filing: **28.01.2002**

(54) **GENERATING RANDOM DATA**

ERZEUGUNG VON ZUFALLSZAHLN

GENERATION DE DONNEES ALEATOIRES

(84) Designated Contracting States:

**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**

(30) Priority: **05.02.2001 GB 0102840**

(43) Date of publication of application:

26.11.2003 Bulletin 2003/48

(73) Proprietor: **Cambridge Silicon Radio Limited**

Cambridge CB4 0WH (GB)

(72) Inventor: **COLLIER, James, Digby, Yarlet,**

Church Farm

Ely,

Cambridgeshire CB6 1SB (GB)

(74) Representative: **Slingsby, Philip Roy et al**

Page White & Farrer

Bedford House

John Street

London, WC1N 2BF (GB)

(56) References cited:

WO-A-00/16182

WO-A-00/75761

GB-A- 2 333 652

US-A- 4 694 412

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 1 364 279 B9

Description

[0001] This invention relates to generating random data, for example for use in encryption and authentication systems.

[0002] Numerous encryption and authentication systems call for the use of random numbers, for example for generating challenges during authentication. Examples are DES and RSA. In systems that offer typical levels of security, random numbers in the range from around 50 to 150 bits in length are required. One way to form truly random numbers of this type is to digitise a noisy analogue value, such as a voltage level, and to use the least significant bits of the digitised result to form the random number. However, this method requires some time to gather enough bits to form a random number of the length that is required for typical encryption systems. Therefore, in most situations an algorithm that generates pseudo-random numbers is used instead. The numbers generated by such an algorithm are not truly random, but are deterministic. Thus this method has the disadvantage that if the algorithm and its seed are known the pseudo-random numbers can be predicted, permitting a third party to break the encryption or authentication scheme.

[0003] There is therefore a need for a method that can quickly produce random numbers that have the property that the next number produced can not be predicted from the previous numbers.

[0004] WO00/75761 discloses a random number generator in which a pseudo random number is produced from a digital random number generator, a first random number is produced from an analogue random number generator, and the first random number is combined with the pseudo random number to produce a second random number that is a result of both generators outputs. US 4,694,412 discloses a random number generator in which the output of two digitally controlled oscillators are exclusively ORed and gated by a counter which has a counting rate determined by a relatively slow digitally controlled oscillator.

[0005] According to one aspect of the present invention there is provided a method for generating random data, as claimed in claim 1.

[0006] According to the second aspect of the present invention there is provided a device for generating random data as claimed in claim 15.

[0007] Preferably the said modifying step comprises modifying the seed value prior to generating the resulting value. Alternatively, the modifying step could comprise modifying a further value, or modifying the resulting value. The said modifying step preferably comprises performing exclusive-OR or addition operations on at least some of the bits of the seed value with corresponding bits of the new truly random data. The seed value could be modified in a single variable or in transformation from one variable to another.

[0008] The output random data is preferably generated by processing the resulting value.

[0009] The truly random data is suitably generated by measurement of a random process at least partially external to the random data generator, for example by comparing the rates of two oscillators.

5 **[0010]** Preferably only one of the oscillators is a crystal-controlled oscillator. The other oscillator could be embodied on an integrated circuit. Preferably one of the oscillators is more accurate and/or stable and/or immune to environmental variation than the other oscillator. The truly random data may be generated by counting the number of oscillations of the one of the oscillators in a predetermined number of oscillations of the other of the oscillators. Preferably the said other of the oscillators is the slower of the oscillators.

10 **[0011]** The step of processing a seed value could comprise processing the seed value by a first modulo exponentiation step to generate a resulting value for use as the seed value in a

[0012] The truly random data may be one or more of the least significant bits of the said number of oscillations.

15 **[0013]** The present invention will now be described by way of example, with reference to the accompanying drawing, in which:

25 figure 1 is a block diagram of a communications device including a random number generator; and figure 2 illustrates the steps of an algorithm for generating random numbers.

30 **[0014]** The device of figure 1 includes a pseudo random number generator 1. The random number generator is shown as including a processor 2 that includes general purpose processing hardware 3, non-volatile program memory 4 for storing program code for the processing hardware 3 and volatile temporary store memory 5 for use by the processing hardware in performing processing operations. However, any suitable means of data processing, including hard-wired processing apparatus and mixed hardware/software embodiments could be used. The pseudo random number generator has an input 6 by means of which it can be invoked to output random data at output 7. The random number generator has access to a source 8 of truly random data. In the embodiment of figure 8 the source is external to the random number generator and has another function in the communications device. However, the source could be internal to the random number generator and/or could have a dedicated function of forming random data.

35 **[0015]** The source 8 suitably includes a store 10 for storing truly random bits as they become available for use by the random number generator. The random number generator has access to that store over link 11 for determining how many truly random bits are available, for reading the truly random bits that are available, and for resetting the store once the bits in it have been used.

40 **[0016]** When a call for random data is received at input 6 by the random number generator 1 the processing means 2 performs a series of processing steps as de-

scribed below to generate a random number. The random number is then output by the random number generator at output 7. When the random number generator is called the processing means is arranged to access a seed value stored in temporary store 5, perform an algorithm which takes the seed value as input and based on that seed value to generate random data and a seed value which is stored for use by the next iteration of the algorithm. As part of the algorithm the processing means determines whether new truly random data, which has not been used in a previous iteration of the algorithm, is available from the source 8. If such data is available the processing means modifies the seed value originally taken for the present iteration in accordance with the new truly random data, and uses that modified seed as the basis for the present iteration; otherwise the seed as originally taken is used as the basis for the present iteration. In this way, truly random data can be used as it comes available in order to randomise the formation of the random data, without the formation of the data having to wait for truly random data to be available. This has the key advantages that by the time a sufficient number of outputs 7 have been collected to allow prediction of the next output, the seed will have changed in an unpredictable way.

[0017] One example of an algorithm that could be used will now be described.

[0018] Before the algorithm is executed, a number of constant values must be defined. These constant values may suitably be defined when the random number generator is designed or constructed - i.e. at system build time. The constant values are as follows:

N represents an integer of around 800.

L is a number of truly random bits that may be available from the source 8.

K_1 and K_2 are small Fermat primes such as 3, 17, 257 and 65537 (K_1 and K_2 may be equal).

p_1 and p_2 are distinct prime numbers of length $N/2$. If a "strong-S-prime", where S is a non-negative integer, is defined as a prime p such that $p-1$ has a strong-(S-1)-prime factor of at least $3/4$ as many bits' length as p , a strong-0-prime being simply a prime; p_1 and p_2 are selected to be strong-2-primes.

M is the product of p_1 and p_2 .

[0019] Once M has been calculated p_1 and p_2 are preferably discarded irretrievably or stored with high security.

[0020] The constant values that are called upon during the performance of the algorithm: L, K_1 , K_2 and M; are preferably stored in the non-volatile memory 4.

[0021] At each iteration of the algorithm a seed value is taken as input to the algorithm. The seed value is modified by the algorithm, and the modified value is taken as the seed for the next iteration of the algorithm. The seed to be used for the next iteration to be performed is stored at a specified location 9 in the temporary store 5. An initial seed is required by the algorithm for input on the first iteration for which it is called to generate random data.

One way to form the initial seed is to collect a series of random bits from the source 8, and to store those in successive bit positions in the specified location 9. Another way to form the initial seed is to store a further constant value in the non-volatile memory 4; to load that value into the specified location 9; and then to perform a specified number of iterations of the algorithm, making use of random data from the source 8 to modify the seed as successive iterations are called. Either of these methods is suitably employed when the communication device is initialised (e.g. at power-up) so that the random number generator is then ready for use, holding a truly random seed.

[0022] The algorithm is illustrated generally in figure 2.

[0023] When the algorithm is called upon to generate a random number, the seed stored in the specified location 9 is obtained (step 30 in figure 2). The seed as obtained from the specified location is stored as a variable x. Then a check is made on store 10 to find whether L truly-random bits are available from the source 8 (step 31).

[0024] If L truly-random bits are available then those bits are loaded into a variable z and the store 11 is reset. Then step 32 of the algorithm is executed to modify the variable x in accordance with those truly random bits. The variable x is modified by being set equal to:

$$(x \oplus z) \bmod M$$

where the symbol \oplus represents exclusive-ORing of each of L predetermined bits of x with the corresponding bits of z. The result of the exclusive-OR operation is reduced to modulo M in order to keep it within arithmetic bounds of the algorithm. The specification of which bits of x are to be exclusive-ORed with which bits of z may suitably be defined at system build time.

[0025] If no truly-random bits are available then x is not modified.

[0026] The seed for the next iteration of the algorithm is formed in variable W, which is set equal to $(x + 1)^{K_2} \bmod M$ (step 33).

[0027] Then the value of w is stored in the specified location 9 so as to replace the previous seed value, and allow it to serve as the initial value of x for the next iteration (step 34).

[0028] Finally, the random data that is to be output from the random number generator is generated and stored in variable v. Variable v is set equal to $X^{K_1} \bmod M$ (step 35). The value of v is used as a supply of random bits which are made available at output 7.

[0029] Instead of exclusive-ORing the seed value with the truly random bits, other approaches could be used. For example, in a suitable algorithm another value than the input seed could be modified in dependence on the random data. Specifically, in the above algorithm the output seed value could be modified. Other modifications

than exclusive-ORing, such as arithmetic shifting, or addition, could be used.

[0030] Numerous methods are available for generating the truly random data. Examples include digitising noisy analogue values from analogue sensors in the device, such voltage levels from a temperature sensor 14; or timing intervals between keypresses by a user on a keypad 16. A preferred method makes use of a pair of oscillators 12, 13 having different levels of short-term accuracy, i.e. 'jitter', and preferably having substantially different rates.

[0031] The device of figure 1 is a radio communication device. Oscillator 12 is relatively fast, relatively accurate and has relatively low jitter. Oscillator 12 is used for modulation of signals for transmission at radio frequency and suitably has a frequency of a few tens of megahertz, for example 13 or 26 MHz for a GSM-based system. Oscillator 13 is a slower, less accurate and more jittery oscillator, which could be used, for example, for interval timing between periods when the faster oscillator is switched off. The frequency of oscillator 13 could be a few kHz. The high accuracy oscillator 12 is suitably timed from a crystal, whereas the low accuracy oscillator 13 is suitably based on a simple resistor and capacitor circuit which may be on the same integrated circuit as the processor 2. Thus there is likely to be drift between the frequencies of the oscillators 12 and 13 due, for instance, to temperature variations and due to random fluctuations due to thermal noise in the resistor.

[0032] In order to generate random data a counter 16 counts the number of transitions of the fast oscillator in a predetermined, preferably small, number of periods of the slow oscillator. Because the jitter in the slow oscillator is in the same order as, or larger than, the period of the oscillator 12, the lowest significant bits of this count will be truly random. Therefore, one or more of those lowest significant bits are used as the random data. For example, where the frequency of the fast oscillator is 16MHz and the frequency of the slower oscillator is 1 kHz, in typical conditions the bottom 1 or 2 bits of this count may be used as the random data. This count is performed periodically, and the resulting bits stored in the store 10. Thus, after the count has been performed new data will be available to the random number generator. When the count has not been performed since the last call of the random, number generator new data will not be available, and in that case the random number generator can still provide an output since it can operate whether or not new truly random data is available.

[0033] The processor could be dedicated to the formation of random numbers, or could perform other functions too. In the latter case the random numbers described herein as being generated by the processor could be subsequently processed in further operations carried out by the processor itself.

[0034] In view of the foregoing description it will be evident to a person skilled in the art that various modifications may be made within the scope of the invention

as defined by the appended claims.

Claims

1. A method for generating random data, the method comprising repeatedly performing a series of operations, and the series of operations comprising processing a seed value of a first predetermined number of bits to generate a resulting value for use as the seed value in a subsequent performance of the series of operations and to generate output random data (7);

characterised in that:

the step of processing a seed value to generate a resulting value comprises processing by means of a modulo exponentiation algorithm; the method comprises the further step of collecting new truly random data at a source; the series of operations comprises the further steps of:

- (i) determining whether the newly collected truly random data comprises a second predetermined number of bits; and
 - (ii) only if the second predetermined number of bits is available, modifying the generation of at least the resulting value in dependence on the newly collected truly random data by performing an operation on each of the second predetermined number of predetermined bits of the seed value, the said predetermined bits being at least some of the bits of the seed value, with corresponding bits of the newly collected truly random data.
2. A method as claimed in claim 1, wherein the said modifying step comprises modifying the seed value prior to generating the resulting value.
 3. A method as claimed in claim 2, wherein the said modifying step comprises performing an exclusive - OR operation on the said predetermined bits with corresponding bits of the newly collected truly random data.
 4. A method as claimed in any preceding claim, wherein the output random data (7) is generated by processing the resulting value.
 5. A method as claimed in any preceding claim, wherein the truly random data is generated by comparing the rates of two oscillators (12,13).
 6. A method as claimed in claim 5, wherein only one of the oscillators (12) is a crystal-controlled oscillator.

7. A method as a method as claimed in claim 5 or 6, wherein the truly random data is generated by counting the number of oscillations of the one of the oscillators (12,13) in a predetermined number of oscillations of the other of the oscillators (12,13). 5
8. A method as claimed in claim 7, wherein the truly random data is one or more of the least significant bits of the said number of oscillations. 10
9. A method as claimed in claim 7 or 8, comprising adjusting the rate of one of the oscillators in dependence on at least one of the most significant bits of the said number of oscillations. 15
10. A method as claimed in any of claims 5 to 9, wherein the rate of one of the oscillators is at least 100 times that of the other oscillator.
11. A method as claimed in any of claims 5 to 10, wherein one of the oscillators (12,13) generates a signal for radio frequency modulation or demodulation of communication data. 20
12. A method as claimed in claim 11, wherein the other of the oscillators generates a signal for timing of idle communication periods. 25
13. A method as claimed in any of claims 1 to 12 wherein the step of processing a seed value comprises: 30
- processing the seed value by a first modulo exponentiation step to generate a resulting value for use as the seed value in a subsequent performance of the series of operations; and 35
- processing the seed value by a second modulo exponentiation step to generate output random data.
14. A method as claimed in any of claims 1-13 comprising collecting truly random data to form an initial seed of the first predetermined number of truly random bits. 40
15. A device (1) for generating random data, the device comprising: 45
- a source (8) of truly random data;
- a first store (10) for storing a seed value of a first predetermined number of bits; and 50
- processing means (2) for performing a series of operations comprising processing the seed value to generate a resulting value for storage in the first store for use as the seed value in a subsequent performance of the series of operations 55
- and to generate output random data;

characterised in that:

the step of processing the seed value to generate a resulting value comprises processing by means of a modulo exponentiation algorithm; the device comprises a second store for storing new truly random data from the source; the series of operations also comprises:

- (i) determining whether a second predetermined number of bits of new truly random data is available from the second store; and
- (ii) only if the second predetermined number of bits is available, modifying the generation of at least the resulting value in dependence on the new truly random data by performing an operation on each of the second predetermined bits of the seed value, the said predetermined bits being at least some of the bits of the seed value, with corresponding bits of the newly collected truly random data.

16. A device as claimed in claim 15, wherein the step of processing the seed value comprises:

processing the seed value by a first modulo exponentiation step to generate a resulting value for use as the seed value in a subsequent performance of the series of operations; and processing the seed value by a second modulo exponentiation step to generate output random data.

17. A communication device comprising a device as claimed in claim 15 or 16.

Patentansprüche

1. Verfahren zum Erzeugen von Zufallsdaten, wobei das Verfahren das wiederholte Ausführen einer Reihe von Operationen umfasst, wobei die Reihe von Operationen das Verarbeiten eines Startwertes einer ersten vorgegebenen Anzahl von Bits umfasst, um einen resultierenden Wert zu erzeugen, der als der Startwert in einer nachfolgenden Ausführung der Reihe von Operationen dient, und um Ausgangszufallsdaten (7) zu erzeugen;

dadurch gekennzeichnet, dass:

der Schritt des Verarbeitens eines Startwertes für die Erzeugung eines resultierenden Wertes eine Verarbeitung mittels eines Modulo-Potenzierungs-Algorithmus umfasst;

das Verfahren den weiteren Schritt des Sammelns neuer, wirklich zufälliger Daten bei einer Quelle umfasst;

die Reihe von Operationen die folgenden weiteren Schritte umfasst:

- (i) Bestimmen, ob die neu gesammelten, wirklich zufälligen Daten eine zweite vorgegebene Anzahl von Bits enthalten; und
(ii) nur dann, wenn die zweite vorgegebene Anzahl von Bits verfügbar ist, Modifizieren der Erzeugung wenigstens des resultierenden Wertes in Abhängigkeit von den neu gesammelten, wirklich zufälligen Daten durch Ausführen einer Operation an jedem Bit der zweiten vorgegebenen Anzahl von vorgegebenen Bits des Startwertes, die wenigstens einige der Bits des Startwertes sind, mittels entsprechender Bits der neu gesammelten, wirklich zufälligen Daten.
2. Verfahren nach Anspruch 1, bei dem der Modifikationsschritt das Modifizieren des Startwertes vor der Erzeugung des resultierenden Wertes umfasst.
3. Verfahren nach Anspruch 2, bei dem der Modifizierungsschritt das Ausführen einer Exklusiv-ODER-Operation an den vorgegebenen Bits mit entsprechenden Bits der neu gesammelten, wirklich zufälligen Daten umfasst.
4. Verfahren nach einem vorhergehenden Anspruch, bei dem die Ausgangszufallsdaten (7) durch Verarbeiten des resultierenden Wertes erzeugt werden.
5. Verfahren nach einem vorhergehenden Anspruch, bei dem die wirklich zufälligen Daten durch Vergleichen der Raten zweier Oszillatoren (12, 13) erzeugt werden.
6. Verfahren nach Anspruch 5, bei dem nur einer der Oszillatoren (12) ein durch einen Kristall gesteuerter Oszillator ist.
7. Verfahren nach Anspruch 5 oder 6, bei dem die wirklich zufälligen Daten durch Zählen der Anzahl von Oszillationen des einen der Oszillatoren (12, 13) während einer vorgegebenen Anzahl von Oszillationen des anderen der Oszillatoren (12, 13) erzeugt werden.
8. Verfahren nach Anspruch 7, bei dem die wirklich zufälligen Daten eines oder mehrere der niedrigstwertigen Bits der Anzahl von Oszillationen sind.
9. Verfahren nach Anspruch 7 oder 8, das das Einstellen der Rate eines der Oszillatoren in Abhängigkeit von wenigstens einem der höchstwertigen Bits der Anzahl von Oszillationen umfasst.
10. Verfahren nach einem der Ansprüche 5 bis 9, bei dem die Rate eines der Oszillatoren wenigstens gleich der 100-fachen Rate des anderen Oszillators ist.
11. Verfahren nach einem der Ansprüche 5 bis 10, bei dem einer der Oszillatoren (12, 13) ein Signal für eine Hochfrequenzmodulation oder -demodulation von Kommunikationsdaten erzeugt.
12. Verfahren nach Anspruch 11, bei dem der andere der Oszillatoren ein Signal für die Taktung von inaktiven Kommunikationsperioden erzeugt.
13. Verfahren nach einem der Ansprüche 1 bis 12, bei dem der Schritt des Verarbeitens eines Startwertes umfasst:
- Verarbeiten des Startwertes durch einen ersten Modulo-Potenzierungs-Schritt, um einen resultierenden Wert zu erzeugen, der als der Startwert in einer nachfolgenden Ausführung der Reihe von Operationen verwendet wird; und Verarbeiten des Startwertes durch einen zweiten Modulo-Potenzierungs-Schritt, um Ausgangszufallsdaten zu erzeugen.
14. Verfahren nach einem der Ansprüche 1 bis 13, das das Sammeln wirklich zufälliger Daten umfasst, um einen anfänglichen Startwert der ersten vorgegebenen Anzahl von wirklich zufälligen Bits zu bilden.
15. Vorrichtung (1) für die Erzeugung von Zufallsdaten, wobei die Vorrichtung umfasst:
- eine Quelle (8) für wirklich zufällige Daten; einen ersten Speicher (10) zum Speichern eines Startwertes einer ersten vorgegebenen Anzahl von Bits; und Verarbeitungsmittel (2) für die Ausführung einer Reihe von Operationen, die die Verarbeitung des Startwertes umfassen, um einen resultierenden Wert für die Speicherung in dem ersten Speicher zu erzeugen, der als der Startwert in einer nachfolgenden Ausführung der Reihe von Operationen verwendet wird, und um Ausgangszufallsdaten zu erzeugen;
- dadurch gekennzeichnet, dass:**
- der Schritt des Verarbeitens des Startwertes, um einen resultierenden Wert zu erzeugen, das Verarbeiten mittels eines Modulo-Potenzierungs-Algorithmus umfasst;
- die Vorrichtung einen zweiten Speicher umfasst, um neue wirklich zufällige Daten von der Quelle zu speichern;
- die Reihe von Operationen außerdem umfasst:
- (i) Bestimmen, ob eine zweite vorgegebene Anzahl von Bits von neuen, wirklich zufälligen Daten von dem zweiten Speicher verfügbar ist; und

- (ii) nur dann, wenn die zweite vorgegebene Anzahl von Bits verfügbar ist, Modifizieren der Erzeugung wenigstens des resultierenden Wertes in Abhängigkeit von den neuen, wirklich zufälligen Daten durch Ausführen einer Operation an jedem Bit der zweiten vorgegebenen Anzahl vorgegebener Bits des Startwertes, die wenigstens einige der Bits des Keimwertes sind, mittels entsprechender Bits der neu gesammelten, wirklich zufälligen Daten.
16. Vorrichtung nach Anspruch 15, bei der der Schritt des Verarbeitens des Startwertes umfasst:
- Verarbeiten des Startwertes durch einen ersten Modulo-Potenzierungs-Schritt, um einen resultierenden Wert zu erzeugen, der als der Startwert in einer nachfolgenden Ausführung der Reihe von Operationen verwendet wird; und Verarbeiten des Startwertes durch einen zweiten Modulo-Potenzierungs-Schritt, um Ausgangszufallsdaten zu erzeugen.
17. Kommunikationsvorrichtung, die eine Vorrichtung nach Anspruch 15 oder 16 enthält.
- Revendications**
1. Procédé de génération de données aléatoires, ce procédé incluant la réalisation répétée d'une succession d'opérations et de la succession d'opérations comprenant le traitement d'une valeur germe d'un premier nombre de bits prédéterminé pour produire une valeur résultante destinée à être utilisée en tant que valeur germe dans une réalisation suivante de la succession d'opérations et pour produire des données de sortie aléatoires (7) ; **caractérisé en ce que** :
- l'étape de traitement d'une valeur germe pour produire une valeur résultante comprend le traitement au moyen d'un algorithme d'exponentiation modulo ;
- le procédé comprend en outre l'étape de recueil de nouvelles données vraiment aléatoires au niveau d'une source ;
- la succession d'opérations comprend les étapes supplémentaires suivantes :
- (i) déterminer si les données vraiment aléatoires nouvellement recueillies comprennent un second nombre de bits prédéterminé ; et
- (ii) seulement si le second nombre de bits prédéterminé est disponible, modifier la production d'au moins la valeur résultante
- en fonction des données vraiment aléatoires nouvellement recueillies en réalisant une opération sur chacun du second nombre de bits prédéterminé de la valeur germe, lesdits bits prédéterminés étant au moins certains des bits de la valeur de germe, avec des bits correspondants des données vraiment aléatoires nouvellement recueillies.
2. Procédé selon la revendication 1, dans lequel l'étape de modification comprend la modification de la valeur germe avant la production de la valeur résultante.
3. Procédé selon la revendication 2, dans lequel ladite étape de modification comprend la réalisation d'une opération OU-Exclusif sur les bits prédéterminés avec des bits correspondants des données vraiment aléatoires nouvellement recueillies.
4. Procédé selon l'une quelconque des revendications précédentes, dans lequel les données de sortie aléatoires (7) sont produites en traitant la valeur résultante.
5. Procédé selon l'une quelconque des revendications précédentes, dans lequel les données vraiment aléatoires sont produites en comparant les cadences de deux oscillateurs (12, 13).
6. Procédé selon la revendication 5, dans lequel seulement l'un des oscillateurs (12) est un oscillateur à commande par cristal.
7. Procédé selon la revendication 5 ou 6, dans lequel les données vraiment aléatoires sont produites en comptant le nombre d'oscillations de l'un des oscillateurs (12, 13) pendant un nombre prédéterminé d'oscillations de l'autre des oscillateurs (12, 13).
8. Procédé selon la revendication 7, dans lequel les données vraiment aléatoires correspondent à un ou plusieurs des bits les moins significatifs dudit nombre d'oscillations.
9. Procédé selon la revendication 7 ou 8, comprenant le réglage de la cadence de l'un des oscillateurs en fonction d'au moins un des bits les plus significatifs dudit nombre d'oscillations.
10. Procédé selon l'une quelconque des revendications 5 à 9, dans lequel la cadence de l'un des oscillateurs est au moins cent fois celle de l'autre oscillateur.
11. Procédé selon l'une quelconque des revendications 5 à 10, dans lequel l'un des oscillateurs (12, 13) produit un signal pour une modulation ou une démodulation radiofréquence de données de communica-

tion.

12. Procédé selon la revendication 11, dans lequel l'autre des oscillateurs produit un signal de temporisation de durées de communication au repos. 5
13. Procédé selon l'une quelconque des revendications 1 à 12, dans lequel l'étape de traitement d'une valeur germe comprend : 10
- le traitement de la valeur germe par une première étape d'exponentiation modulo pour produire une première valeur résultante à utiliser comme valeur germe dans une réalisation suivante de la succession d'opérations ; et 15
- le traitement de la valeur germe par une seconde étape d'exponentiation modulo pour produire des données aléatoires de sortie.
14. Procédé selon l'une quelconque des revendications 1 à 13, comprenant le recueil de données vraiment aléatoires pour former un germe initial du premier nombre prédéterminé de bits vraiment aléatoires. 20
15. Dispositif (1) pour produire des données aléatoires, le dispositif comprenant : 25

une source (8) de données vraiment aléatoires ;
une première mémoire (10) pour mémoriser une valeur germe d'un premier nombre prédéterminé de bits ; et 30

un moyen de traitement (2) pour réaliser une succession d'opérations comprenant le traitement de la valeur germe pour produire une valeur résultante pour mémorisation dans la première mémoire pour utilisation en tant que valeur germe dans une réalisation ultérieure de la succession d'opérations et pour produire des données aléatoires de sortie ; 35

40

caractérisé en ce que :

l'étape de traitement de la valeur germe pour produire une valeur résultante comprend le traitement par un algorithme d'exponentiation modulo ; 45

le dispositif comprend une seconde mémoire pour mémoriser de nouvelles données vraiment aléatoires à partir de la source ;

la succession d'opérations comprend également : 50

- (i) déterminer si un second nombre prédéterminé de bits de nouvelles données vraiment aléatoires est disponible à partir de la seconde mémoire ; et 55
- (ii) seulement si le second nombre de bits prédéterminé est disponible, modifier la

production d'au moins la valeur résultante en fonction des nouvelles données vraiment aléatoires en réalisant une opération sur le second nombre de bits prédéterminé de la valeur de germe, lesdits bits prédéterminés étant au moins certains des bits de la valeur de germe, avec des bits correspondants des données vraiment aléatoires nouvellement recueillies.

16. Dispositif selon la revendication 15, dans lequel l'étape de traitement de la valeur germe comprend :

le traitement de la valeur germe par une première d'exponentiation modulo pour produire une première valeur résultante à utiliser comme valeur germe dans une réalisation suivante de la succession d'opérations ; et

le traitement de la valeur germe par une seconde étape d'exponentiation modulo pour produire des données aléatoires de sortie.

17. Dispositif de communication comprenant un dispositif selon la revendication 15 ou 16.

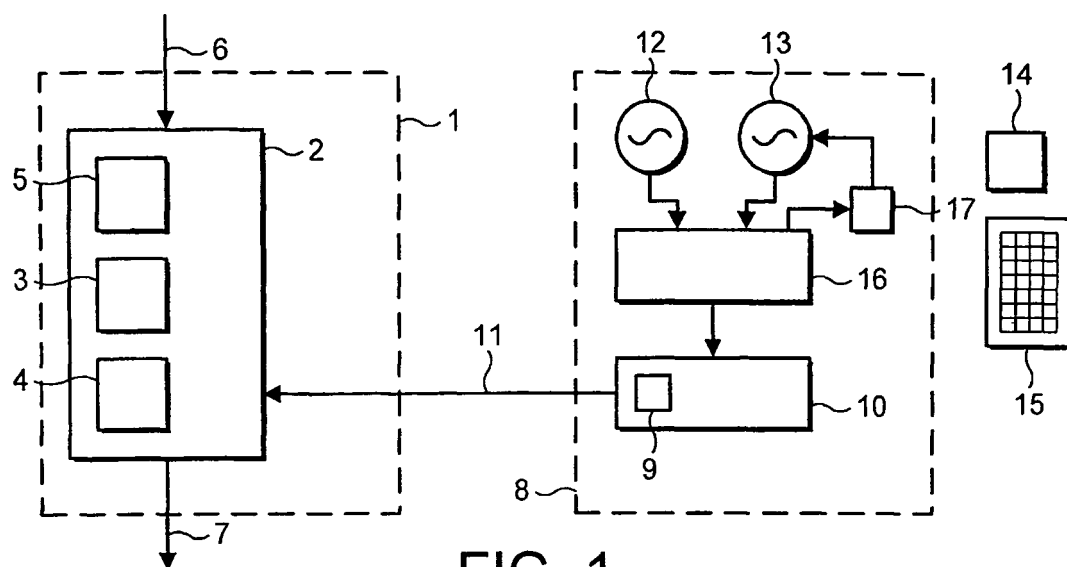


FIG. 1

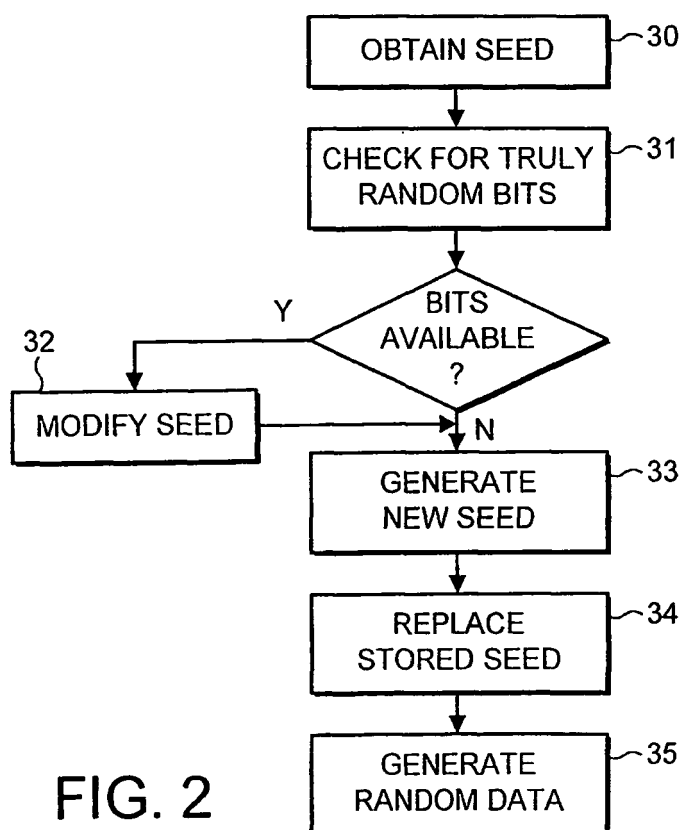


FIG. 2

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 0075761 A [0004]
- US 4694412 A [0004]