numbers, are slow, tie up the microprocessor with the task, clog memory buses, and consume a lot of power/energy (resulting in increased heat dissipation and battery drain).

An additional drawback of prior art pseudorandom number generators is that they are unprotected from side channel attacks, which creates significant risks in IT security applications. During operation of electronic devices that are processing data, some information about secret keys or sensitive data always leaks in side channels (physical sources of unintended information dissemination), including data dependent variation of response times, fluctuation of power use, or ultrasonic or electromagnetic wave radiation. This is why the secrecy and integrity of stored or transmitted data can generally be assured by cryptographic means when no adversary has physical access to the electronic devices.

Thus, it is desirable to have an improved pseudorandom number generator that generates many pseudorandom numbers in one clock cycle and enables improvements in security by reducing side channel leakage without significantly increasing processing time, system complexity, the size of electronic circuits, or energy usage.

BRIEF SUMMARY

In view of the foregoing background, example implementations of the present disclosure provide a system and method for generating pseudorandom numbers. The method includes the steps of initializing a counter value for a call-counter, sending a bit-wise form of the counter value from the call-counter to a mixing function, mixing the bit-wise form of the counter value to generate a pseudorandom number, filling an input register by repeating the counter value in its entirety as many times as possible without exceeding capacity of the input register, and filling any remaining, unfilled bits of the input register with individual bits of the counter value. The system includes a call-counter that is initialized with a counter value, a mixing function that mixes a bit-wise form of the counter value to generate a pseudorandom number, and an input register arranged to be filled by repeating the counter value in its entirety as many times as possible without exceeding capacity of the input register and by filling any unfilled bits of the input register with individual bits of the counter value until the input register is full.

The counter value is incremented by a predetermined value. . Encrypted data may be stored in the memory of a storage device based on the generated pseudorandom number.

In implementations of the present disclosure, the mixing step is performed by a mixing function. The mixing function maps a first segment of the bit-wise form of

CLAIMS:

1. A method for generating pseudorandom numbers comprising the steps of:

initializing a counter value for a call-counter;

sending a bit-wise form of the counter value from the call-counter to a mixing function;

mixing the bit-wise form of the counter value to generate a pseudorandom number;

filling an input register by repeating the counter value in its entirety as many times as possible without exceeding capacity of the input register; and

filling any remaining, unfilled bits of the input register with individual bits of the counter value until the input register is full.

2. The method of claim 1 further comprising at least one of:

the step of incrementing the counter value by a predetermined value; and

the step of using the generated pseudorandom number as an input value for a subsequent cycle of the mixing step.

3. The method of claim 1 or 2,

wherein the mixing step maps a first segment of the bit-wise form of the counter value input with a first segment of the bit-wise form of the counter value output by concatenating the segments, and wherein at least one of:

the mixing step is invertible,

the mixing step is non-linear, and

the mixing step comprises a function selected from the group consisting of an exclusive-OR "XOR" tree mixing function, a substitution-permutation mixing function, and a double-mix Feistel mixing function.

4. The method of any one of claims 1 to 3 wherein the mixing step further comprises steps of mixing the counter value at least twice in parallel mixing functions and exclusive-XORing outputs from the parallel mixing functions.

5. The method of any one of claims 1 to 4 further comprising the steps of:

processing the pseudorandom number by a one-way function or hashing the pseudorandom number with a cryptographic hash function; and

using a result of the one-way processing of the pseudorandom number or hashing of the pseudorandom number as an input value for a subsequent cycle of the mixing function.

6. The method of any one of claims 1 to 5 further comprising the step of using the pseudorandom number to store encrypted data in a memory.

7. A system for generating pseudorandom numbers comprising:

a call-counter that is initialized with a counter value;

a mixing function that mixes a bit-wise form of the counter value to generate a pseudorandom number; and

an input register arranged to be filled by repeating the counter value in its entirety as many times as possible without exceeding capacity of the input register and by filling any unfilled bits of the input register with individual bits of the counter value until the input register is full.

8. The system of claim 7 wherein the counter value is incremented by a predetermined value.

9. The system of claim 7 or 8 wherein the generated pseudorandom number serves as an input value for a subsequent cycle of the mixing function.

10. The system of any one of claims 7 to 9 further comprising a storage device having a memory that stores encrypted data based on the pseudorandom number, the encrypted data being selected from the group consisting of encryption keys, tweak values, nonces, and initial values.

11. The system of any one of claims 7 to 10 wherein the mixing function at least one of:

maps a first segment of the bit-wise form of the counter value input with a first segment of the bit-wise form of the counter value output by concatenating the segments;

is selected from the group consisting of a XOR tree mixing function, substitution-permutation mixing function, and double-mix Feistel mixing function; and

is performed by at least two mixing functions operating in parallel and the outputs from the at least two mixing functions are XORed.

12. The system of any one of claims 7 to 11 wherein at least one of:
the mixing function is invertible, and
the mixing function is non-linear.

13. The system of any one of claims 7 to 12 wherein:

the pseudorandom number is processed by a one-way function or hashed with a cryptographic hash function; and

a result of the one-way processing of the pseudorandom number or hashing of the pseudorandom number is used as an input value for a subsequent cycle of the mixing function.