

Claims

1. A system for storing data on a distributed storage network comprising:
a processor, a memory and instructions store on the memory and executable by

the processor to:

receive a data;

randomise the data;

apply an Information Dispersal Algorithm (IDA) to the randomised data to
obtain slices of data; and

store the slices of data on the distributed storage network, wherein the data is
randomised such that all slices of data are required to reconstruct and re-arrange
to obtain the data.

2. The system according to claim 1 wherein the instruction to randomise the data
comprises instructions to:

divide the data into a plurality of blocks, $m_1, \dots, m_n$, each block being k bit length;

generate an Initialisation Vector (IV) comprising random number of k bit length;

randomise a first block of the plurality of blocks, m_1 , using the IV;

randomise the rest of the plurality of blocks, $m_2, \dots, m_n$, each of the rest of the

plurality of blocks is randomised using the randomised data from the previous block; and
generate an additional block for storing IV.

3. The system according to claim 2 wherein the instruction to randomise the first
block of the plurality of blocks, m_1 , using the IV comprises instruction to:

apply an XOR function to a hash of IV and the first block m_1 , expressed in the
following expression, $m'_1 = h(IV) \oplus m_1$.

4. The system according to anyone of claims 2-3 wherein the instruction to
randomise the rest of the plurality of blocks, $m_2, \dots, m_n$, each of the rest of the plurality of
blocks is randomised using the randomised data from the previous block comprises
instruction to:

apply, for current block, an XOR function to hash of previous randomised block,
 m'_{i-1} , with the current block m_i , expressed in the following expression, $m'_i = h(m'_{i-1}) \oplus m_i$
where $i = 2..n$.

5. The system according to anyone of claims 2-4 wherein the instruction to generate the additional block for storing IV comprises instruction to:

apply an XOR function to a hash of one of the randomised blocks, m'_j , and the IV, expressed with the following expression, $m'_0 = h(m'_j) \oplus IV$ where j is selected from $1 \dots n$.

5

6. The system according to anyone of claims 2-5 further comprising instructions to: retrieve slices of data;

apply the IDA to reconstruct the slices of data to obtain the randomised data; and de-randomise the randomised data to obtain the data.

10

7. The system according to claim 6 wherein the instruction to de-randomise the randomised data to obtain the data comprises instructions to:

divide the randomised data into a plurality of randomised blocks, $\dots m'_n$, each randomised block being k bit length;

15

determine the IV based on the additional randomised block, m'_0 ;

de-randomise a first block of the plurality of randomised blocks, m'_1 , using the IV;

de-randomise the rest of the plurality of randomised blocks, $m'_2, \dots, m'_n$, each of the rest of the plurality of randomised blocks is de-randomised using the previous randomised block; and

20

arrange the de-randomised block to form the data.

8. The system according to claim 7 wherein the instruction to determine the IV based on the additional randomised block, m'_0 , comprises instruction to:

apply an XOR function to a hash of a selected randomised block with the

25

additional randomised block, m'_0 , expressed in the following expression, $IV = h(m'_j) \oplus m'_0$ where j is selected from $1 \dots n$.

9. The system according to anyone of claims 7-8 wherein the instruction to de-randomise the first block of the plurality of randomised blocks, m'_1 , using the IV comprises instruction to:

30

apply an XOR function to the hash of IV and the first randomised block m'_1 , expressed in the following expression, $m_1 = h(IV) \oplus m'_1$.

10. The system according to anyone of claims 7-9 wherein the instruction to de-

35

randomise the rest of the plurality of randomised blocks, $m'_2, \dots, m'_n$, each of the rest of

the plurality of randomised blocks is de-randomised using the previous randomised block comprises instruction to:

apply, for current randomised block, an XOR function to hash of previous randomised block, m'_{i-1} , with the current randomised block m'_i , expressed in the following expression, $m_i = h(m'_{i-1}) \oplus m'_i$ where $i = 2..n$.

11. The system according to anyone of claims 7-10 wherein the de-randomised blocks are arranged with the following expression, $F = m_1, \dots, m_n$.

12. A method for storing data on a distributed storage network comprising:

receiving a data;

randomising the data;

applying an Information Dispersal Algorithm (IDA) to the randomised data to obtain slices of data; and

storing the slices of data on the distributed storage network, wherein the data is randomised such that all slices of data are required to reconstruct and re-arrange to obtain the data.

13. The method according to claim 12 wherein the step of randomising the data comprises:

dividing the data into a plurality of blocks, $m_1, \dots, m_n$, each block being k bit length;

generating an Initialisation Vector (IV) comprising random number of k bit length;

randomising a first block of the plurality of blocks, m_1 , using the IV;

randomising the rest of the plurality of blocks, $m_2, \dots, m_n$, each of the rest of the plurality of blocks is randomised using the randomised data from the previous block; and generating an additional block for storing IV.

14. The method according to claim 13 wherein the step of randomising the first block of the plurality of blocks, m_1 , using the IV comprises:

applying an XOR function to a hash of IV and the first block m_1 , expressed in the following expression, $m'_1 = h(IV) \oplus m_1$.

15. The method according to anyone of claims 13-14 wherein the step of randomising the rest of the plurality of blocks, $m_2, \dots, m_n$, comprises:

applying, for current block, an XOR function to hash of previous randomised block, m'_{i-1} , with the current block m_i , expressed in the following expression, $m'_i = h(m'_{i-1}) \oplus m_i$ where $i = 2..n$.

5 16. The method according to anyone of claims 13-15 wherein the step of generating the additional block for storing IV comprises:

applying an XOR function to a hash of one of the randomised blocks, m'_j , and the IV, expressed with the following expression, $m'_0 = h(m'_j) \oplus IV$ where j is selected from $1...n$.

10

17. The method according to anyone of claims 13-16 further comprising:
retrieving the slices of data from the distributed storage network;
applying the IDA to reconstruct the slices of data to obtain the randomised data;
and

15 de-randomising the randomised data to obtain the data.

18. The method according to claim 17 wherein the step of de-randomising the randomised data to obtain the data comprises:

dividing the randomised data into a plurality of randomised blocks, $\dots m'_n$, each
20 randomised block being k bit length;
determining the IV based on the additional randomised block, m'_0 ;
de-randomising a first block of the plurality of randomised blocks, m'_1 , using the IV;
de-randomising the rest of the plurality of randomised blocks, $m'_2, \dots, m'_n$, each of
the rest of the plurality of randomised blocks is de-randomised using the previous
25 randomised block; and
arranging the de-randomised block to form the data.

19. The method according to claim 18 wherein the step of determining the IV based on the additional randomised block, m'_0 , comprises:

30 applying an XOR function to a hash of a selected randomised block with the additional randomised block, m'_0 , expressed in the following expression, $IV = h(m'_j) \oplus m'_0$ where j is selected from $1...n$.

20. The method according to anyone of claims 18-19 wherein the step of de-randomising the first block of the plurality of randomised blocks, m'_1 , using the IV comprises:

5 applying an XOR function to the hash of IV and the first randomised block m'_1 ,
expressed in the following expression, $m_1 = h(IV) \oplus m'_1$.

21. The method according to anyone of claims 18-20 wherein the step de-randomising the rest of the plurality of randomised blocks, $m'_2, \dots, m'_n$, comprises:

10 applying, for current randomised block, an XOR function to hash of previous
randomised block, m'_{i-1} , with the current randomised block m'_i , expressed in the following
expression, $m_i = h(m'_{i-1}) \oplus m'_i$ where $i = 2..n$.

22. The method according to anyone of claims 18-21 wherein the de-randomised blocks are arranged with the following expression, $F = m_1, \dots, m_n$.

15