CLAIMS

1. A data storage system, comprising:

a processor; and

logic integrated with the processor, executable by the processor, or integrated with and executable by the processor, the logic being configured to:

store a plurality of data blocks, wherein a set of the data blocks are protected by an erasure correcting code, wherein each of the data blocks in the set of data blocks includes block identification information;

verify the block identification information for each of the data blocks in the set of data blocks at a time of read; and

as part of reconstructing a data block, reconstruct the block identification information for the data block~~,~~ and verify the block identification information.

2. The data storage system of claim 1, wherein the block identification information for each of the data blocks is unique thereto.

3. The data storage system of claim 1~~ or claim 2~~, comprising logic configured to declare that one of the data blocks is erased in response to determining that the verification of the block identification information fails.

4. The data storage system of ~~any preceding~~ claim 1, comprising at least two parity blocks.

5. The data storage system of claim 4, wherein the block identification information of the data blocks and of the at least two parity blocks are symbols in a codeword of a Reed-Solomon code such that all the symbols are different, wherein the codeword in the Reed-Solomon code is the vector $(1, \alpha, \alpha^2, \ldots, \alpha^{q-1})$, where $q=2^b$ and α is a primitive element in the finite field GF(q), where *b* is the length of the vector.

6. The data storage system of claim 5, wherein the length of the codeword in the Reed-Solomon code is ~~the vector $(2^b-1, \alpha, \alpha^2, \ldots, \alpha^{q-1})$, where $q = 2^b$ and α is a primitive element in the finite field GF(q), where *b* is the length of the vector.~~

~~7. The data storage system of claim 6, wherein the length of the codeword in the Reed-Solomon code is 2^b-1.~~

~~8.~~7. ~~The data storage system of claim 6~~5, wherein the length of the codeword in the Reed-Solomon code is shorter than 2^b-1.

~~9.~~8. The data storage system of claim ~~5~~6, wherein some entries in the vector are virtual and not assigned to any data block.

~~10.~~9. The data storage system of claim 4, wherein the data blocks and the parity blocks are distributed in a two dimensional array where the XOR of the blocks in rows and columns have even parity, wherein each of the blocks in the array has a unique block identification information symbol, wherein the XOR of the block identification information symbols in each row and in each column are equal to zero.

~~11.~~10. The data storage system of claim ~~10~~9, wherein the array is a p ~~×~~ × p array in an Expanded Blaum-Roth code, wherein p a prime number.

~~12.~~11. The data storage system of claim ~~10~~9, wherein an array comprising the block identification information symbols is also in an Expanded Blaum-Roth code.

12. The data storage system of claim ~~12~~11, wherein the array comprising the block identification information symbols is divided into first and second halves, wherein an array comprising the first half of the block identification information symbols is in an Expanded Blaum-Roth code and an array comprising the second half of the block identification information symbols is also in an Expanded Blaum-Roth code.

13. A data storage system, comprising:

a processor; and

logic integrated with the processor, executable by the processor, or integrated with and executable by the processor, the logic being configured to:

store a plurality of data blocks, wherein a set of the data blocks are protected by an erasure correcting code, wherein each of the data blocks in the set of data blocks includes block identification information;

store at least two parity blocks, wherein each of the at least two parity blocks includes block identification information, wherein the block identification information of the data blocks and of the at least two parity blocks are symbols in a codeword of a Reed-Solomon code such that all the symbols are different;

verify the block identification information for each of the data blocks in the set of data blocks at a time of read;

as part of reconstructing a data block, reconstruct the block identification information for the data block and verify the block identification information; and

return the block identification information and the reconstructed data block for providing end-to-end protection in the storage system.

14. The data storage system of claim 13, wherein the block identification information for each of the data blocks is unique thereto.

15. The data storage system of claim 13, comprising logic configured to declare that one of the data blocks is erased in response to determining that the verification of the block identification information fails.

16. The data storage system of claim 13, wherein the length of the codeword in the Reed-Solomon code is 2^b-1, wherein the codeword in the Reed-Solomon code is the vector $(1, \alpha, \alpha^2, \ldots, \alpha^{q-1})$, where $q=2^b$ and α is a primitive element in the finite field GF(q), where *b* is the length of the vector.

17. The data storage system of claim 13, wherein the length of the codeword in the Reed-Solomon code is shorter than 2^b-1, wherein the codeword in the Reed-Solomon code is the vector $(1, \alpha, \alpha^2, \ldots, \alpha^{q-1})$, where $q=2^b$ and α is a primitive element in the finite field GF(q), where *b* is the length of the vector.

18. The data storage system of claim 13, wherein the data blocks and the parity blocks are distributed in a two dimensional array where the XOR of the blocks in rows and columns have even parity, wherein each of the blocks in the array has a unique block identification information symbol, wherein the XOR of the symbols in each row and in each column are equal to zero.

19. The data storage system of claim 18, wherein the array is a p×p array in an Expanded Blaum-Roth code, wherein p a prime number.

20. The data storage system of claim 18, wherein an array comprising the block identification information symbols is also in an Expanded Blaum-Roth code.

21. The data storage system of claim 20, wherein the array comprising the block identification information symbols is divided into first and second halves, wherein an array comprising the first half of the block identification information symbols is in an Expanded Blaum-Roth code and an array comprising the second half of the block identification information symbols is also in an Expanded Blaum-Roth code.

22. A computer-implemented method, comprising:

for a plurality of data blocks in a data storage system, verifying block identification information for each of the data blocks in a set of data blocks at a time of read,

wherein a set of data blocks are protected by an erasure correcting code, wherein each of the data blocks in the set of data blocks includes the block identification information; and

as part of reconstructing a data block, reconstructing the block identification information for the data block and verifying the block identification information.

23. The method of claim ~~14~~22, wherein the block identification information for each of the data blocks is unique thereto.

24. A computer-implemented method, comprising:

for a plurality of data blocks in a data storage system, verifying block identification information for each of the data blocks in a set of data blocks at a time of read, ~~The method of claim 14 or claim 15,~~

wherein the data storage system comprises at least two parity blocks~~.~~, wherein a set of the data blocks are protected by an erasure correcting code, wherein each of the data blocks in the set of data blocks includes the block identification information,

~~The method of any of claims 14 to 16~~wherein the block identification information of the data blocks and of the at least two parity blocks are symbols in a codeword of a Reed-Solomon code such that all the symbols are different~~.~~;

as part of reconstructing a data block, reconstructing the block identification information for the data block and verifying the block identification information; and

returning the block identification information and the reconstructed data block for providing end-to-end protection in the storage system.

~~13. The method of claim 17, wherein the codeword in the Reed-Solomon code is the vector $(1, \alpha, \alpha^2, \ldots, \alpha^{q-1})$, where $q = 2^b$ and α is a primitive element in the finite field $GF(q)$, where b is the length of the vector.~~

~~14. The method of claim 18, wherein the length of the codeword in the Reed-Solomon code is $2^b - 1$.~~

~~15. The method of claim 18, wherein the length of the codeword in the Reed-Solomon code is shorter than $2^b - 1$.~~

25. A computer program product, the computer program product comprising:

one or more computer readable storage media, and program instructions collectively stored on the one or more computer readable storage media, the program instructions ~~executable on a processor to perform the method of any of claims 14 to 20.~~comprising:

program instructions to, for a plurality of data blocks in a data storage system, verify, by the computer, block identification information for each of the data blocks in a set of the data blocks at a time of read,

wherein a set of data blocks are protected by an erasure correcting code, wherein each of the data blocks in the set of data blocks includes the block identification information; and

program instructions to, as part of reconstructing a data block, reconstruct, by the computer, the block identification information for the data block and verify the block identification information.