(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 April 2006 (20.04.2006)

PCT

(10) International Publication Number
WO 2006/040682 A2

(51) International Patent Classification: Not classified

(21) International Application Number: PCT/IB2005/003371

(22) International Filing Date: 12 October 2005 (12.10.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2004905897 13 October 2004 (13.10.2004) AU

(71) Applicant *(for all designated States except US)*: **SYNAPTIC LABORATORIES LIMITED** [CH/CH]; 6 rue Verdaine, CH-1204 Geneva (CH).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **O'NEIL, Sean** [AU/AU]; 255/421 Brunswick St, Fortitude Valley QLD 4006 (AU).

(74) Agent: **EVANS, Allen, John**; 5/238 The Avenue, Parkville VIC 3052 (AU).

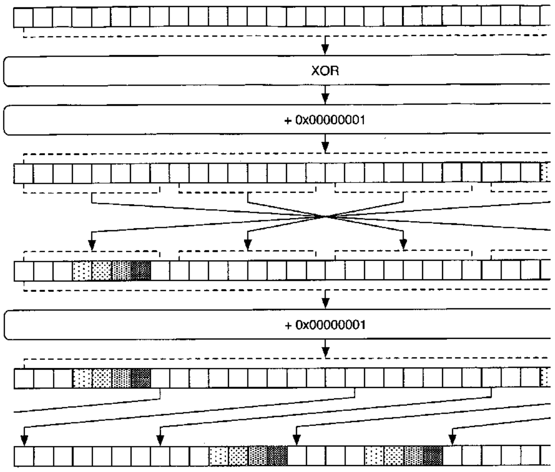
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** PROCESS OF AND APPARATUS FOR ENCODING A DIGITAL INPUT



(57) **Abstract:** A cryptographic process (10) that receives input (11) and produces output (2). The cryptographic process (10) produces each block of output (20) by performing on a block of input (11), in any order, at least one operation (15) of a first type; at least one operation (19) of a second type; at least one operation (13, 17) of a third type; and at least one operation (12) of a fourth type. The operations of the first type (15) are swapping (SWAP) and bit order reversal. The operations of the second type (19) are bitwise rotation to the left (ROTL) and bitwise rotation to the right (ROTR). The operations of the third type (13, 17) are addition (ADD), subtraction (SUB) and negation (NEG). The operations of the fourth type (12) are exclusive-or (XOR), inverse exclusive-or (XNOR), logical AND, inverse logical AND (NAND), logical OR, inverse logical OR (NOR) and logical inverse (NOT). When both the first operation and the last operation in the cryptographic process (10) are swap operations (15), the cryptographic process includes a further swap operation (15).

WO 2006/040682 A2

Title

Process of and apparatus for encoding a digital input

Field of the invention

The present invention relates to cryptographic primitives.

Background of the invention

Throughout this specification, including the claims, we use the terms 'comprises' and 'comprising' to specify the presence of stated features, integers, steps or components but without precluding the presence or addition of one or more other features, integers, steps, components or groups.

In the cryptographic art, crypto-systems can be implemented in dedicated hardware or general-purpose processors. It is desirable that the software-implementation of cryptographic processes on general-purpose processors can efficiently exploit the instruction sets and execution profiles provided on general-purpose hardware.

Summary of the invention

Accordingly, in one aspect we provide a cryptographic process that receives at least one block of input and produces an output block from the at least one block of input, the process comprising:

the performance, in any order, of:

at least one operation of a first type;
at least one operation of a second type;
at least one operation of a third type; and
at least one operation of a fourth type;

each operation of the first type being chosen from the group consisting of: swapping (SWAP) and bit order reversal,

each operation of the second type being chosen from the group consisting of: bitwise rotation to the left (ROTL) and bitwise rotation to the right (ROTR);

each operation of the third type being chosen from the group consisting of: addition (ADD), subtraction (SUB) and negation (NEG); and

each operation of the fourth type being chosen from the group consisting of: exclusive-or (XOR), inverse exclusive-or (XNOR), logical AND, inverse logical AND (NAND), logical OR, inverse logical OR (NOR) and logical inverse (NOT),

and in which, when both the first operation and the last operation in the cryptographic process are swap operations, the cryptographic process further comprises a swap operation.

It is preferred that at least one operation of the second type uses at least one input chosen from the group consisting of: key material, data material and counter material.

It is preferred that at least one operation of the third type uses at least one input chosen from the group consisting of: key material, data material and counter material.

It is preferred that at least one operation of the fourth type uses at least one input chosen from the group consisting of: key material, data material and counter material.

It is preferred that at least one operation of the first type:

is immediately preceded by an operation chosen from the group consisting of:

an operation of the third type; and

an operation of the fourth type,

and is immediately followed by an operation which is also chosen from that group.

In this case, it is preferred that the immediately following operation is of a different type from the type of the immediately preceding operation

It is preferred that at least one operation of the second type:

is immediately preceded by an operation chosen from the group consisting of:

an operation of the third type; and

an operation of the fourth type,

and is immediately followed by an operation which is also chosen from that group.

In this case it is preferred that the immediately following operation is of a different type from the type of the immediately preceding operation

It is preferred that:

at least one fixed N-bit constant is used in at least one operation of the third type or of the fourth type; and

that N-bit constant is chosen as a balanced non-linear Boolean function with log(N) inputs.

It is preferred that all operations of the third type and of the fourth type use an N-bit constant which is chosen as a balanced non-linear Boolean function with log(N) inputs.

In other aspects, we provide apparatus, machine readable substrates, data and signals as summarized in the claims at the end of this specification.

It will be seen that these processes and apparatus provide arithmetic operations, which achieve fast balancing of the distribution of monomials of all possible algebraic degrees in the polynomial relationships between all the bits of input, be it data, key or counter material.

We achieve this while maintaining fast execution on modern high-performance general-purpose processors such as the Pentium and PowerPC architectures.

Due to the unbalanced nature of operations with carry, the polynomial relationship between different bits of output and the input bits to operations with carry will have a different number of monomials and a different algebraic degree for different bit positions.

We correct this imbalance by using two different classes of transposition operations to achieve faster balancing of monomials and algebraic degrees of each of the bits than either class of transposition operations can achieve on their own. Combining two different classes of transposition operations allows widening the range of different bit permutations occurring in the encryption process.

Brief description of the drawings

In the drawings, figures 1, 2 and 3 illustrate preferred embodiments of the invention.

Description of embodiments of the invention

There are three basic transposition operations available in most modern processors that can be used to compensate for the algebraic imbalances: fixed constant rotation, variable rotation and byte order reversal operations.

The byte-swap operation on a 32-bit word is the fastest balancing function as it transposes the order of 4 groups of 8 bits. The byte-swap operation is readily available for 16-bit, 32-bit, 64-bit words such as found on Sparc, MMX and 3DNow! instruction sets and 128-bit words as found on SSE instruction sets.

In figure 1, reference number 10 indicates a process according to a preferred embodiment of the invention.

Reference number 11 indicates a 32-bit wide word. The least significant bit of the 32-bit word 11 is illustrated as the rightmost bit.

The exclusive or (XOR) operation 12 has the word 11 as input and performs a 32-bit wide XOR operation with a second 32-bit value. The second 32-bit value is not illustrated.

The addition operation 13 has the word 12 as input and performs an addition with the constant hexadecimal value 0x00000001 to generate output 13. The cross-hatched boxes in word 13 visually illustrate the probability of each bit generating a carry overflow as a result of the addition operation 13.

During encryption where all variable input values can be usually seen as pseudo-random, the difference between algebraic addition operation (ADD) and a bitwise addition (XOR) has on average more than 75% zeroes and less than 25% ones representing the carry overflow bits: each bit in the addition operation 13 has a 25% probability of generating a carry overflow in the next bit. Once a carry overflow is produced, the probability of its reversal in the following bits is 25% for the immediately following bit decreasing exponentially by 75% with every bit. Thus in order to construct a cryptographically secure cipher, the highly localised small influence of carry overflow bits that also leave less significant bits unaffected needs to be diffused to all other bits with carefully chosen transposition operations.

Byte-swap operation 15 has the word 14 as input and performs a byte-swap operation to produce the output word 16. In the illustration of word 16 in the drawing, it can be seen that the cross-hatching that appeared on the right of the figure in word 14 now appears transposed towards the more significant bits of word 16 as a result of the byte-swap.

The order reversal operation acts as a form of corrective balancing, compensating the dependency bias found in the lowest and the highest bits of the output across the entire word width.

Addition operation 17 has the word 16 as input and performs an addition with the constant hexadecimal value 0x00000001 to generate output 18. The cross-hatched boxes in word 18 visually illustrate the probability of each bit generating a carry overflow as a result of the first addition operation 13 and the second arithmetic operation 17.

It is clear that a byte swap operation of word 18 would result in a redundant transposition.

The rotation operation is a slower compensating construction than is the byte-swap operation, only permuting two contiguous sequences of bits and also not changing their order.

The static rotation operation 19 has the word 18 as input and performs a static rotation left by 17 bits to generate output word 20. Output word 20 visually illustrates the distribution of influence of a carry bit after a byte-swap operation followed by a left 17-bit rotation.

Figure 2 illustrates a portion of the loop of iteratively applied byte-swap and 17-bit rotation operations showing each bit's position after every transposition operation. In figure 2, word 31 illustrates a 32-bit word with a label for each bit position.

In figure 2, byte-swap operation 32 has the word 31 as input and generates word 33 as output.

Rotation operation 34 has the word 33 as input and performs a 32-bit wide rotation by 17 bits left to generate word 35 as output.

Words 36, 38, 40, 42 are the results of a byte-swap operation performed on words 35, 37, 39 and 41.

Words 37, 39, 41, 43 are the results of a 32-bit wide rotation by 17 bits left performed on words 36, 38, 40, 42.

Visually inspecting figure 2 it is clear that not only each of the 32 bits of word 31 is cycled into a unique position, but also in such a way, that the biased influence of all carry bits in arithmetic operations is quickly balanced. It can also be visually seen that such combination has advantages of both rotation and byte swapping operations and does not have their disadvantages: byte-swapping operations may get canceled out in subsequent iterations, and rotation operations offer less balancing of the biased carry overflow influence and maintain the same order of bits throughout the entire cipher operation. Interleaving byte-swapping with any rotation other than by a number of bits divisible by 8 (including 0) results in a transposition that ensures the influence of carry-bits is not cancelled out in a later operation. Interleaving byte-swapping and rotation operations between arithmetic and logical operations also introduces a new effect of continuously changing the order of bits in the word.

The byte-swap operation combined with rotation operations plays a role of a cryptographic transposition operation. This is a fundamentally different from performing byte order conversions to ensure compatibility between big-endian and little-endian architectures, which can be achieved by performing a byte-swap as the first operation when receiving a block of data to encode and as the last operation before returning the encoded block of output.

According to further preferred embodiments of the invention, the reiterated sequence of a static rotation followed by a byte-swap operation over 32-bit, 64-bit, 128-bit or 256-bit word lengths achieves a maximal distance permutation of bits if one or two static rotations by an odd constant are performed between each byte-swap operation. The direction of the static rotation is irrelevant to achieving the desired output distribution; however, single rotations between byte-swaps do not result in maximal length loops for 64-bit or wider words. If such property is desired, multiple rotation operations should be performed before executing the next byte-swap.

Figure 3 illustrates a process according to a further preferred embodiment of the invention.

Word 51 is an input to a cryptographic process 52. Process 52 illustrates a cryptographic process such as a round function. The process 52 comprises at least one arithmetic operation 53 selected from the set of: addition (ADD), subtraction (SUB) and negation (NEG). The process 52 further comprises at least one rotation operation 54 selected from the set of: rotation left (ROL) and rotation right (ROR). Process 52 further comprises at least one byte-swap operation 55. Process 52 further comprises at least one operation 56 selected from the set of Boolean operators: exclusive-or (XOR), inverse exclusive-or (XNOR), logical AND, inverse logical AND (NAND), logical OR, inverse logical OR (NOR) and logical inverse (NOT).

Various further preferred embodiments of the invention (which are not illustrated in the drawings) use plaintext, key material and counter values as input parameters into any of the abovementioned operations, for example allowing use of data-dependent or key-dependent operations and s-boxes implemented either as look-up tables or using bit-slicing techniques.

The output 58 thus depends on at least one operation of each of the four classes of operation 53, 54, 55 and 56. The order of operations 53, 54, 55 and 56 is arbitrary.

Many modern processor architectures such as PowerPC and Pentium families optimize the performance of instructions sequences that match common application execution profiles. The arbitrary execution of operations selected from 53, 54, 55 and 56 achieve high performance on the above processors because they match common application execution profiles. Multiplication operations are not recommended due to the poor performance when executed in close proximity with byte-swap or rotation operations on the above processors.

Process 52 further comprises at least 1 s-box look-up operation 57 from a precomputed table of values stored in memory.

In further preferred embodiment of the invention, arithmetic operations such as illustrated as 53 in figure 3, are interleaved with Boolean logic operations such as represented as 56 in figure 3, to ensure their non-associative and non-commutative behaviour.

In yet further preferred embodiments of the invention, for every three consecutive occurrences of sequences of contiguous transposition operations from the class 54 and 55, the third sequence of transposition operations is not the inverse of the first sequence of transposition operations.

In yet further preferred embodiments of the invention, for every two consecutive occurrences of sequences of contiguous transposition operations from the class 54 and 55, the second sequence of transposition operations is not the inverse of the first sequence of transposition operations.

Claims:

1. A cryptographic process that receives at least one block of input and produces an output block from the at least one block of input, the process comprising:

    the performance, in any order, of:

        at least one operation of a first type;

        at least one operation of a second type;

        at least one operation of a third type; and

        at least one operation of a fourth type;

    each operation of the first type being chosen from the group consisting of: swapping (SWAP) and bit order reversal,

    each operation of the second type being chosen from the group consisting of: bitwise rotation to the left (ROTL) and bitwise rotation to the right (ROTR);

    each operation of the third type being chosen from the group consisting of: addition (ADD), subtraction (SUB) and negation (NEG); and

    each operation of the fourth type being chosen from the group consisting of: exclusive-or (XOR), inverse exclusive-or (XNOR), logical AND, inverse logical AND (NAND), logical OR, inverse logical OR (NOR) and logical inverse (NOT),

    and in which, when both the first operation and the last operation in the cryptographic process are swap operations, the cryptographic process further comprises a swap operation.

2. A cryptographic process as claimed in claim 1, in which at least one operation of the second type uses at least one input chosen from the group consisting of: key material, data material and counter material.

3. A cryptographic process as claimed in claim 1 or claim 2, in which at least one operation of the third type uses at least one input chosen from the group consisting of: key material, data material and counter material.

4. A cryptographic process as claimed in claim any one of the preceding claims, in which at least one operation of the fourth type uses at least one input chosen from

the group consisting of: key material, data material and counter material.

5. A cryptographic process as claimed in any one of the preceding claims, in which at least one operation of the first type:

is immediately preceded by an operation chosen from the group consisting of:

an operation of the third type; and

an operation of the fourth type,

and is immediately followed by an operation which is also chosen from that group.

6. A cryptographic process as claimed in claim 5, in which the immediately following operation is of a different type from the type of the immediately preceding operation

7. A cryptographic process as claimed in any one of the preceding claims, in which at least one operation of the second type:

is immediately preceded by an operation chosen from the group consisting of:

an operation of the third type; and

an operation of the fourth type,

and is immediately followed by an operation which is also chosen from that group.

8. A cryptographic process as claimed in claim 7, in which the immediately following operation is of a different type from the type of the immediately preceding operation

9. A cryptographic process as claimed in any one of the preceding claims, in which:

at least one fixed N-bit constant is used in at least one operation of the third type or of the fourth type; and

that N-bit constant is chosen as a balanced non-linear Boolean function with log(N) inputs.

10. A cryptographic process as claimed in claim 9, in which all operations of the third type and of the fourth type use an N-bit constant which is chosen as a balanced non-linear Boolean function with log(N) inputs.

11. Cryptographic apparatus that receives at least one block of input and produces an output block from the at least one block of input by performing on the block of input the cryptographic process comprising:

the performance, in any order, of:

at least one operation of a first type;

at least one operation of a second type;

at least one operation of a third type; and

at least one operation of a fourth type;

each operation of the first type being chosen from the group consisting of: swapping (SWAP) and bit order reversal,

each operation of the second type being chosen from the group consisting of: bitwise rotation to the left (ROTL) and bitwise rotation to the right (ROTR);

each operation of the third type being chosen from the group consisting of: addition (ADD), subtraction (SUB) and negation (NEG); and

each operation of the fourth type being chosen from the group consisting of: exclusive-or (XOR), inverse exclusive-or (XNOR), logical AND, inverse logical AND (NAND), logical OR, inverse logical OR (NOR) and logical inverse (NOT),

and in which, when both the first operation and the last operation in the cryptographic process are swap operations, the cryptographic process further comprises a swap operation.

12. Cryptographic apparatus as claimed in claim 11, in which at least one operation of the second type uses at least one input chosen from the group consisting of: key material, data material and counter material.

13. Cryptographic apparatus as claimed in claim 11 or claim 12, in which at least one

operation of the third type uses at least one input chosen from the group consisting of: key material, data material and counter material.

14. Cryptographic apparatus as claimed in claim any one of claims 11 to 13, in which at least one operation of the fourth type uses at least one input chosen from the group consisting of: key material, data material and counter material.

15. Cryptographic apparatus as claimed in any one of claims 11 to 14, in which at least one operation of the first type:

is immediately preceded by an operation chosen from the group consisting of:

an operation of the third type; and

an operation of the fourth type,

and is immediately followed by an operation which is also chosen from that group.

16. Cryptographic apparatus as claimed in claim 15, in which the immediately following operation is of a different type from the type of the immediately preceding operation

17. Cryptographic apparatus as claimed in any one of the claims 11 to 16, in which at least one operation of the second type:

is immediately preceded by an operation chosen from the group consisting of:

an operation of the third type; and

an operation of the fourth type,

and is immediately followed by an operation which is also chosen from that group.

18. Cryptographic apparatus as claimed in claim 17, in which the immediately following operation is of a different type from the type of the immediately preceding operation

19. A cryptographic process as claimed in any one of claims 11 to 18, in which:

at least one fixed N-bit constant is used in at least one operation of the third type or of the fourth type; and

that N-bit constant is chosen as a balanced non-linear Boolean function with log(N) inputs.

20. Cryptographic apparatus as claimed in claim 19, in which all operations of the third type and of the fourth type use an N-bit constant which is chosen as a balanced non-linear Boolean function with log(N) inputs.

21. Data which has been generated by the process of any one of claims 1 to 10.

22. A machine readable substrate carrying data which has been generated according to the process of any one of claims 1 to 10.

23. A signal carrying data which has been generated according to the process of any one of claims 1 to 10.

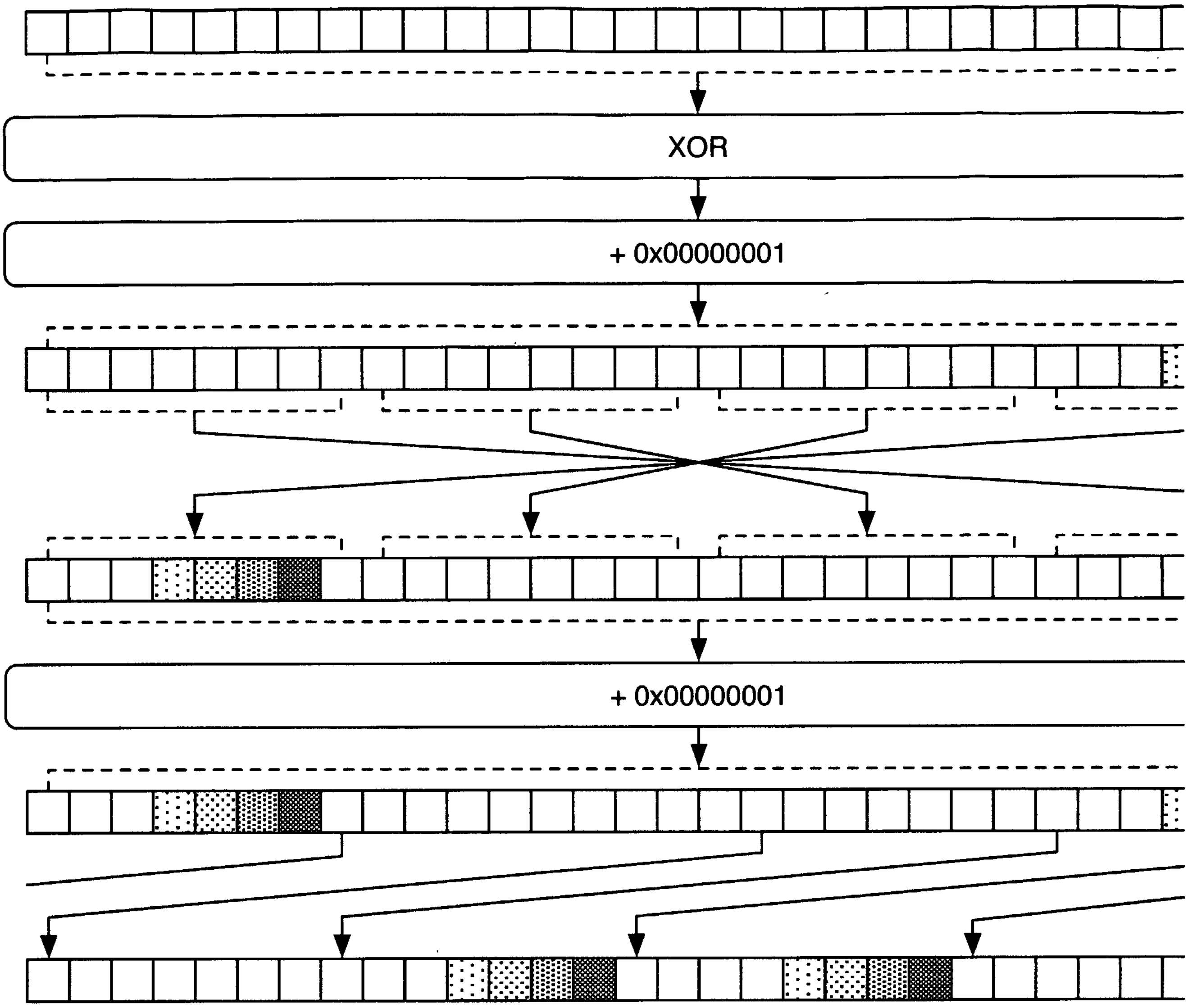


Figure 1

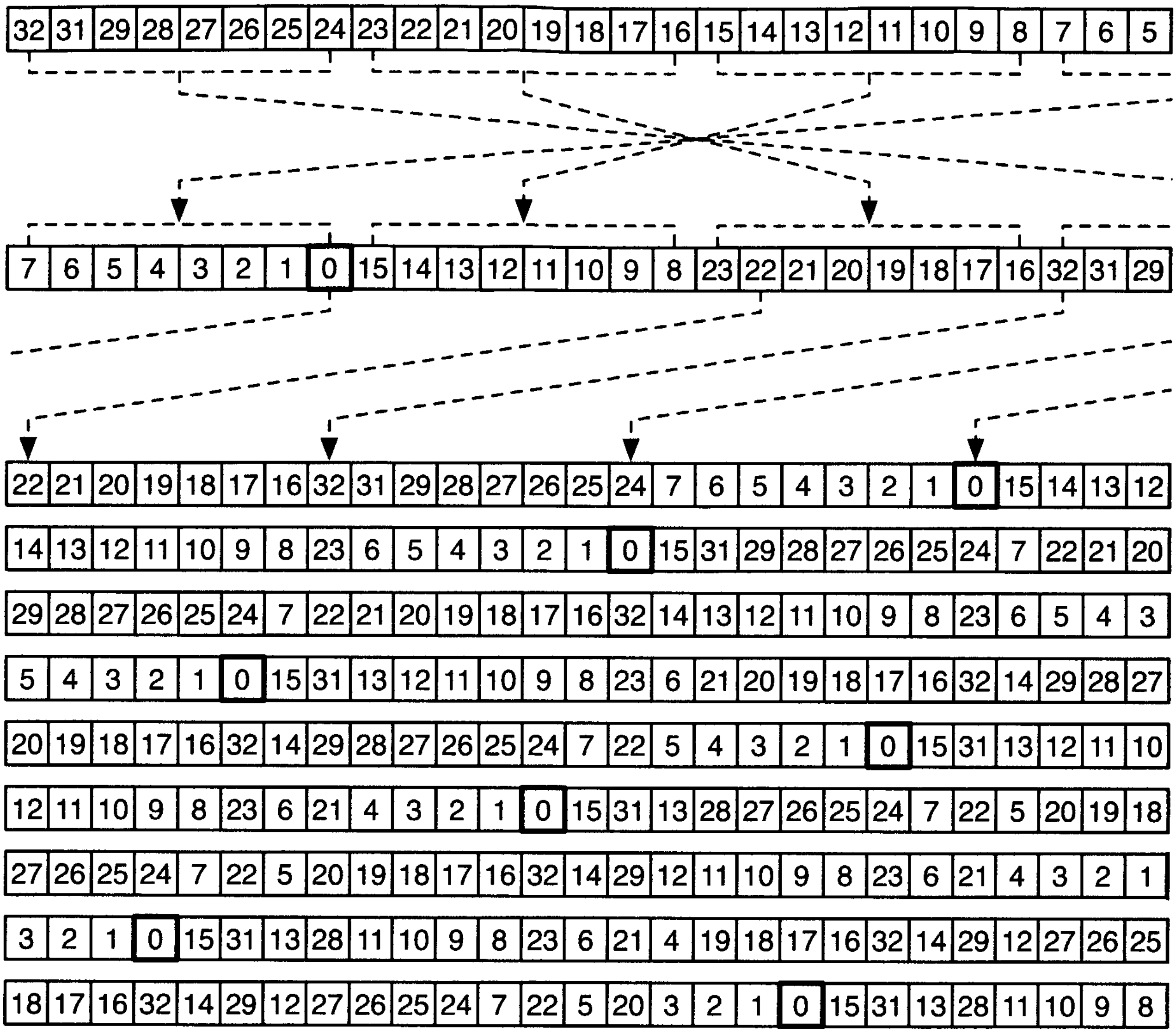

32 31 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 23 22 21 20 19 18 17 16 32 31 29
22 21 20 19 18 17 16 32 31 29 28 27 26 25 24 7 6 5 4 3 2 1 0 15 14 13 12
14 13 12 11 10 9 8 23 6 5 4 3 2 1 0 15 31 29 28 27 26 25 24 7 22 21 20
29 28 27 26 25 24 7 22 21 20 19 18 17 16 32 14 13 12 11 10 9 8 23 6 5 4 3
5 4 3 2 1 0 15 31 13 12 11 10 9 8 23 6 21 20 19 18 17 16 32 14 29 28 27
20 19 18 17 16 32 14 29 28 27 26 25 24 7 22 5 4 3 2 1 0 15 31 13 12 11 10
12 11 10 9 8 23 6 21 4 3 2 1 0 15 31 13 28 27 26 25 24 7 22 5 20 19 18
27 26 25 24 7 22 5 20 19 18 17 16 32 14 29 12 11 10 9 8 23 6 21 4 3 2 1
3 2 1 0 15 31 13 28 11 10 9 8 23 6 21 4 19 18 17 16 32 14 29 12 27 26 25
18 17 16 32 14 29 12 27 26 25 24 7 22 5 20 3 2 1 0 15 31 13 28 11 10 9 8

Figure 2

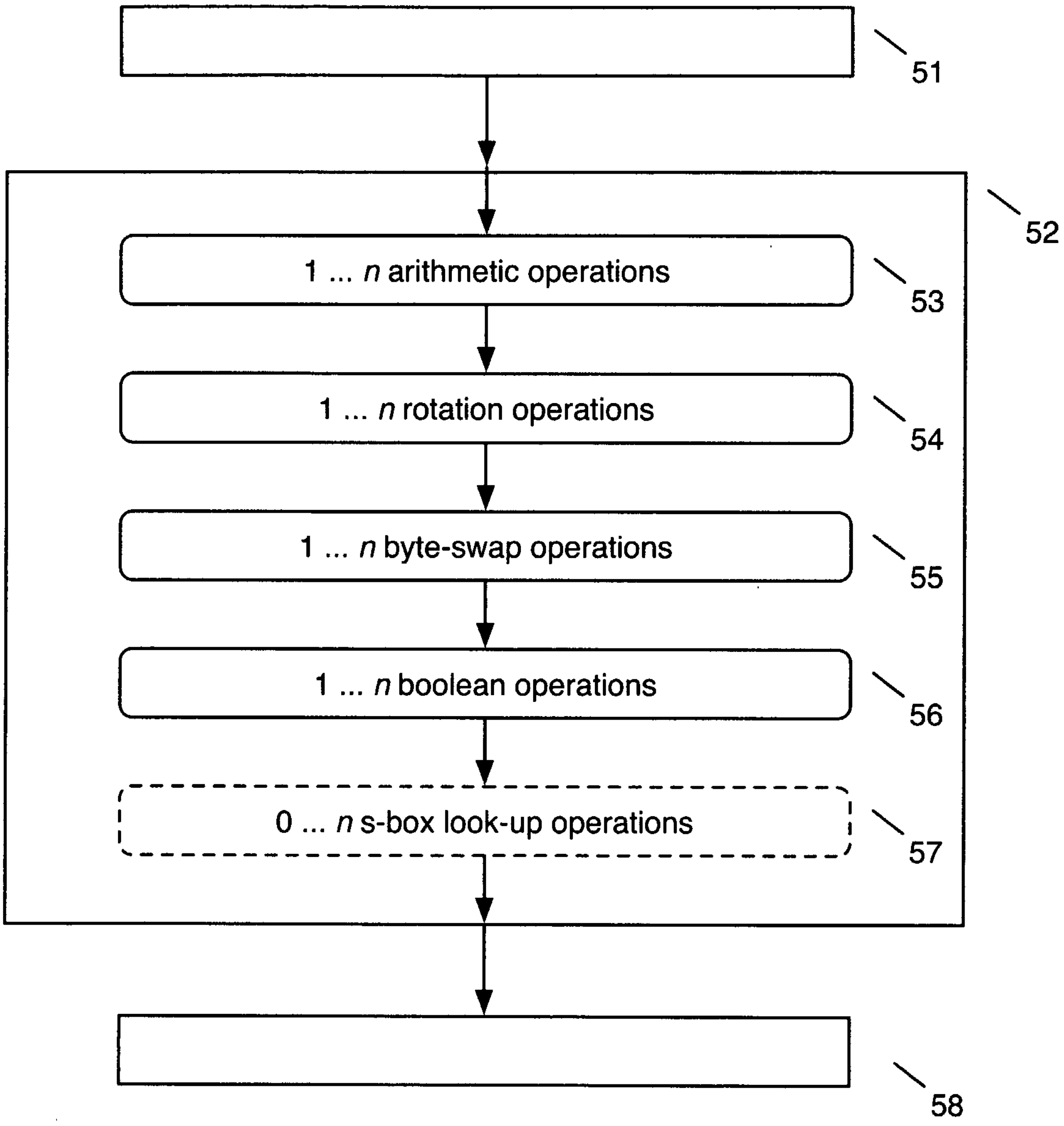

51
52
1 ... *n* arithmetic operations
53
1 ... *n* rotation operations
54
1 ... *n* byte-swap operations
55
1 ... *n* boolean operations
56
0 ... *n* s-box look-up operations
57
58

Figure 3