

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
2 October 2008 (02.10.2008)

PCT

(10) International Publication Number
WO 2008/117142 A1

(51) International Patent Classification:

H04L 9/06 (2006.01)

(21) International Application Number:

PCT/IB2007/054592

(22) International Filing Date:

12 November 2007 (12.11.2007)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

182246 27 March 2007 (27.03.2007) IL

(71) Applicant (for all designated States except US): **NDS LIMITED** [GB/GB]; 1 Heathrow Boulevard, 286 Bath Road, West Drayton Middlesex UB7 0DQ (GB).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **MANTIN, Itsik** [IL/IL]; 6 Hamizpe Street, 73142 Shoham (IL). **WAIS-BARD, Erez** [IL/IL]; 7 Moshe Dayan Street, 60401 Or-yehuda (IL). **KIPNIS, Aviad** [IL/IL]; 24 Beit Ha'arava Street, 93389 Jerusalem (IL).

(74) Agents: **ZVIEL, David** et al.; Director - Intellectual Property, NDS Technologies Israel Limited, 5 Shlomo Halevi Street, 97770 Jerusalem (IL).

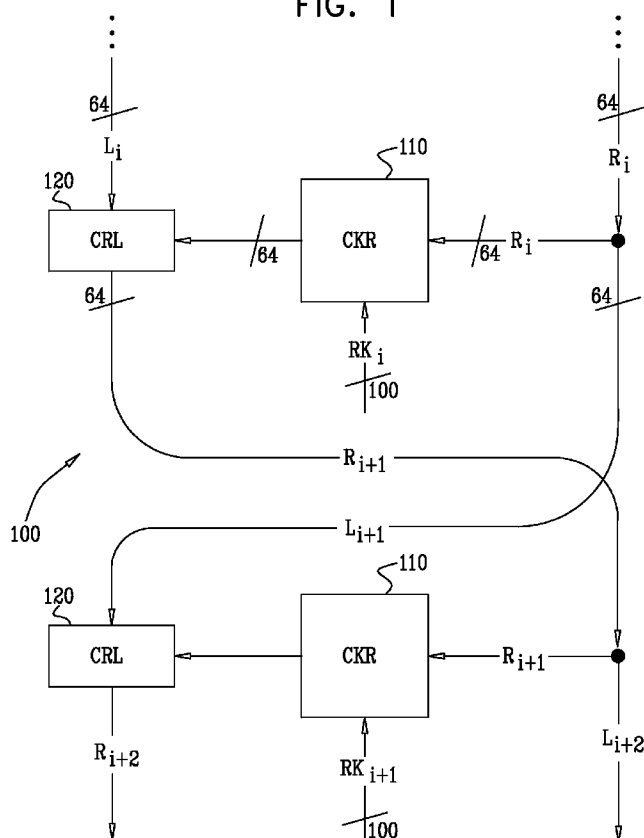
(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN,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

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR BLOCK CIPHER ENCRYPTION

FIG. 1



(57) Abstract: A method and system of encrypting a block of data is described, the method including providing a combining unit operative to combine a key with a block of data, the block of data expressed as a block of bits, providing a mix and condense unit (MAC) operative to mix bits included in the block of bits among themselves, providing a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to provide an output including an input to the mix and condense unit, receiving an input including the block of data expressed as the block of bits, combining, at the combining unit, the block of bits with a key, receiving an output of the combining unit as an input, substituting bits including the input to the plurality of layers of S-boxes with bits including the output of the plurality of layers of S-boxes, outputting the output of the plurality of layers of S-boxes to the mix and condense unit, and mixing, at the mix and condense unit, the output of the plurality of layers of S-boxes, thereby producing an encrypted block of bits.



PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments*

Published:

— *with international search report*

METHOD AND SYSTEM FOR BLOCK CIPHER ENCRYPTION

FIELD OF THE INVENTION

The present invention relates to methods of encryption, and more particularly, to Feistel based block cipher methods of encryption

BACKGROUND OF THE INVENTION

Many encryption methods are known in the art. Of the known methods, many methods are block methods in which a block of plain text is iteratively altered according to a predefined rule; each such iteration is also known as a “round”.

Many block encryption methods can be viewed as specific cases of Feistel networks, also termed herein “Feistel cipher methods”, or “Feistel-like cipher methods”; a single round of a Feistel cipher method is termed herein a “Feistel cipher round”.

Feistel ciphers are described in the Handbook of Applied Cryptography (A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996). The Handbook of Applied Cryptography (HAC) is available on the Internet at www.cacr.math.uwaterloo.ca/hac). The discussion of Feistel ciphers in HAC, on pages 250 - 259, is incorporated herein by reference.

A Feistel cipher is an iterated block cipher mapping a plaintext (comprising two parts, L_0 and R_0), for t -bit blocks L_0 and R_0 , to a ciphertext (R_r and L_r), through an r -round process where $r \geq 1$. For $1 \leq i \leq r$, round i maps (L_{i-1}, R_{i-1}) using key K_i to (L_i, R_i) as follows: $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, where each subkey K_i is derived from the cipher key K (HAC, page 251).

Those skilled in the art will appreciate that although the definition above is for blocks L_0 and R_0 of equal sizes, equality of the sizes is not mandatory.

Decryption of a Feistel cipher is often achieved using the same r -round process but with subkeys used in reverse order, K_r through K_1 .

Throughout the present specification and claims, the terms “first half” and “second half” are used to mean one of either: “right half” or “left half”.

Types of block ciphers which are cases of Feistel networks include the following well-known methods: DES, Lucifer, FEAL, Khufu, Khafre, LOKI,
5 GOST, CAST, and Blowfish.

Feistel ciphers are also discussed in Applied Cryptography, Second Edition (B. Schneier, John Wiley and Sons, Inc., 1996) on pages 347 - 351. The discussion of Feistel ciphers in Applied Cryptography, Second Edition is hereby incorporated herein by reference.

10 DES is specified in FIPS 46-3, available on the Internet at: csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf. FIPS 46-3 is hereby incorporated herein by reference.

FOX: A New Family of Block Ciphers, (Pascal Junod and Serge Vaudenay, Selected Areas in Cryptography 2004: Waterloo, Canada, August 9-10, 2004. Revised papers, Lecture Notes in Computer Science. Springer-Verlag.)
15 describes the design of a new family of block ciphers based on a Lai-Massey scheme, named FOX. The main features of the design, besides a very high security level, are a large implementation flexibility on various platforms as well as high performances. In addition, a new design of strong and efficient key-schedule
20 algorithms is proposed. Evidence is provided that FOX is immune to linear and differential cryptanalysis.

How to Construct Pseudorandom Permutations From Pseudorandom Functions (M. Luby and C. Rackoff., SIAM Journal on Computing, 17:2, pp. 373--386, April 1988), describes a method to efficiently construct a pseudorandom
25 invertible permutation generator from a pseudorandom function generator. A practical result described in Luby-Rackoff is that any pseudorandom bit generator can be used to construct a block private key cryptosystem which is secure against chosen plaintext attacks, which is one of the strongest known attacks against a cryptosystem.

30 The Serpent Cipher, specified at: www.ftp.cl.cam.ac.uk/ftp/users/rja14/serpent.pdf, was an Advanced Encryption Standard (AES) candidate. The design of the serpent cipher design is highly

conservative, yet still allows a very efficient implementation. The serpent cipher uses S-boxes similar to those of DES in a new structure that simultaneously allows a more rapid avalanche, and a more efficient bitslice implementation.

- Unpublished PCT application, PCT/IL2006/001167, filed 5 October
- 5 2006, of NDS Ltd. describes one preferred implementation of a Feistel-like cipher and a method of encrypting a block of data, the method including providing a combining unit operative to combine a key with a block of data, the block of data expressed as a block of bits, providing a mix and condense unit operative to mix bits included in the block of bits among themselves, receiving an input including
- 10 the block of data expressed as the block of bits, combining, at the combining unit, the block of bits with a key, and mixing, at the mixing and condensing unit, the combined block of bits, wherein the mix and condense unit includes a plurality of layers, each layer among the plurality of layers including a plurality of mini-functions. Related apparatus and methods are described. The disclosure of
- 15 Unpublished PCT application, PCT/IL2006/001167 is incorporated herein by reference.

The disclosures of all references mentioned above and throughout the present specification, as well as the disclosures of all references mentioned in those references, are hereby incorporated herein by reference.

SUMMARY OF THE INVENTION

The present invention seeks to provide an improved encryption method, and in particular an improved encryption method related to Feistel encryption methods. A Feistel-like cipher, described herein, is preferably
5 designed to be easily implemented in hardware and difficult to implement in software.

There is thus provided in accordance with a preferred embodiment of the present invention a method of encrypting a block of data, the method including providing a combining unit operative to combine a key with a block of
10 data, the block of data expressed as a block of bits, providing a mix and condense unit (MAC) operative to mix bits included in the block of bits among themselves, providing a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to provide an output including an input to the mix and condense unit,
15 receiving an input including the block of data expressed as the block of bits, combining, at the combining unit, the block of bits with a key, receiving an output of the combining unit as an input, substituting bits including the input to the plurality of layers of S-boxes with bits including the output of the plurality of layers of S-boxes, outputting the output of the plurality of layers of S-boxes to the
20 mix and condense unit, and mixing, at the mix and condense unit, the output of the plurality of layers of S-boxes, thereby producing an encrypted block of bits.

Further in accordance with a preferred embodiment of the present invention the plurality of layers of S-boxes includes two layers of S-boxes.

Still further in accordance with a preferred embodiment of the
25 present invention a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes including an input of the second layer of S-boxes.

Additionally in accordance with a preferred embodiment of the
30 present invention each S-box of the plurality of S-boxes includes one S-box described in the Serpent cipher specification.

Moreover in accordance with a preferred embodiment of the present invention the two layers of S-boxes include a first layer of S-boxes including 25 S-boxes and a second layer of S-boxes including 25 S-boxes.

Further in accordance with a preferred embodiment of the present invention the combining unit is operative to perform a XOR operation.

Still further in accordance with a preferred embodiment of the present invention the method of encrypting cannot be efficiently implemented except on specialized hardware.

Additionally in accordance with a preferred embodiment of the present invention the MAC comprises a plurality of layers of mini-functions.

Moreover in accordance with a preferred embodiment of the present invention the plurality of layers of the MAC includes between 30 layers and 50 layers, inclusive.

Further in accordance with a preferred embodiment of the present invention a mini-function layer includes two micro-functions one balanced micro-function, and one non-linear micro-function.

Still further in accordance with a preferred embodiment of the present invention the mini-function layer is operative to perform the following receiving an input, splitting the input, at a splitter, into a block of balancing bits and a block of remaining input bits, executing the method of the non-linear micro-function on the block of remaining input bits, inputting the result of the non-linear micro-function into the balanced micro-function, executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits, and outputting a result.

Additionally in accordance with a preferred embodiment of the present invention the method further including performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

Moreover in accordance with a preferred embodiment of the present invention the invertible transformation includes an invertible transformation S-box.

Further in accordance with a preferred embodiment of the present invention and wherein the invertible transformation S-box includes a 2bit-to-2bit S-box.

Still further in accordance with a preferred embodiment of the present invention the method further including providing a first function F_i and a second function F_j , providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of the first function F_i , and the second function F_j , providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of the first function F_i , and the second function F_j , utilizing the round key generation function in at least a first round to generate a second round key for use in a second round, and utilizing the round mixing function in at least the first round to mix a first round key with a cipher state, wherein one of the following is performed in the first round the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state, and the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

There is also provided with another preferred embodiment of the present invention a method of encrypting a block of data, the method including providing an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being greater than the first bit size, providing a combining unit operative to combine an expanded block of data with a key, providing a mix and condense unit (MAC) operative to mix the bits of a combined expanded block of data of the second bit size and condense the bit size of the input to a third bit size, the third bit size being less than the second bit size, providing a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input

into the mix and condense unit and to provide an output including an input, expressed as a block of data, to the mix and condense unit, receiving an input including the block of data expressed as the block of bits, inputting the block of bits into the expansion unit, and therein expanding the block of bits to a block of bits of the second bit size, combining, at the combining unit, the block of bits of the second bit size with a key, substituting bits including the input to the plurality of layers of S-boxes with bits including the output of the plurality of layers of S-boxes, outputting the output of the plurality of layers of S-boxes to the mix and condense unit, and mixing, at the mix and condense unit, the output of the plurality of layers of S-boxes, the output of the plurality of layers of S-boxes including a block of bits of the second bit size, and condensing, at the mix and condense unit, the block of bits of the second bit size to a block of bits of the third size, thereby producing an encrypted block of data, the encrypted block of data being expressed as a block of bits of the third bit size, wherein the mix and condense unit includes a plurality of layers, each layer among the plurality of layers including a plurality of mini-functions.

Further in accordance with a preferred embodiment of the present invention the plurality of layers of S-boxes includes two layers of S-boxes.

Still further in accordance with a preferred embodiment of the present invention a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes including an input of the second layer of S-boxes.

Additionally in accordance with a preferred embodiment of the present invention each S-box of the plurality of S-boxes includes one S-box described in the Serpent cipher specification.

Moreover in accordance with a preferred embodiment of the present invention the two layers of S-boxes include a first layer of S-boxes including 25 S-boxes and a second layer of S-boxes including 25 S-boxes.

Further in accordance with a preferred embodiment of the present invention the first bit size is equal to the third bit size.

Still further in accordance with a preferred embodiment of the present invention the first bit size is equal to 64 bits.

Additionally in accordance with a preferred embodiment of the present invention the second bit size is equal to 100 bits.

5 Moreover in accordance with a preferred embodiment of the present invention the third bit size is equal to 64 bits.

Further in accordance with a preferred embodiment of the present invention the combining unit is operative to perform a XOR operation.

10 Still further in accordance with a preferred embodiment of the present invention the expansion unit includes a linear transformation.

Additionally in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein each input bit influences at least two output bits.

15 Moreover in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein each bit of the key influences one output bit.

Further in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein any small set of input bits influences a larger set of output bits.

20 Still further in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein indices are selected so as to be spread equally between input bits and output bits.

25 Additionally in accordance with a preferred embodiment of the present invention the expansion unit includes two layers of gates operative to combine two inputs.

Moreover in accordance with a preferred embodiment of the present invention the gates include XOR operation gates.

30 Further in accordance with a preferred embodiment of the present invention the method further includes a NOT operation gate after the XOR operation gates.

Still further in accordance with a preferred embodiment of the present invention the method of encrypting cannot be implemented except on specialized hardware.

5 Still further in accordance with a preferred embodiment of the present invention the MAC comprises a plurality of layers of mini-functions.

Additionally in accordance with a preferred embodiment of the present invention the plurality of layers of the MAC includes between 30 layers and 50 layers, inclusive.

10 Moreover in accordance with a preferred embodiment of the present invention a mini-function layer includes two micro-functions one balanced micro-function, and one non-linear micro-function.

Further in accordance with a preferred embodiment of the present invention the mini-function layer is operative to perform the following receiving an input, splitting the input, at a splitter, into a block of balancing bits and a block
15 of remaining input bits, executing the method of the non-linear micro-function on the block of remaining input bits, inputting the result of the non-linear micro-function into the balanced micro-function, executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits, and outputting a result.

20 Still further in accordance with a preferred embodiment of the present invention the method further includes performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

25 Additionally in accordance with a preferred embodiment of the present invention the invertible transformation includes an invertible transformation S-box.

Moreover in accordance with a preferred embodiment of the present invention the invertible transformation S-box includes a 2bit-to-2bit S-box.

30 Further in accordance with a preferred embodiment of the present invention the method further includes providing a first function F_i and a second function F_j , providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of the first

function F_i , and the second function F_j , providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of the first function F_i , and the second function F_j , utilizing the round key generation function in at least a first round to generate a second round key for use in a second round, and utilizing the round mixing function in at least the first round to mix a first round key with a cipher state, wherein one of the following is performed in the first round the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state, and the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

There is also provided with still another preferred embodiment of the present invention a method of encrypting a block of data, the method including an emulation resistant combine key method included in a Feistel-like structure.

Further in accordance with a preferred embodiment of the present invention the method is implemented in hardware.

Still further in accordance with a preferred embodiment of the present invention the method further includes mixing and condensing, the mixing and condensing including receiving an input of a block of data expressed as a block of bits, and mixing the bits of the block of data with a round key.

Additionally in accordance with a preferred embodiment of the present invention the method further includes providing an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being greater than the first bit size, providing a combining unit operative to combine an expanded block of data with a key, providing a mix and condense unit (MAC) operative to mix the bits of a combined expanded block of data of the second bit size and condense the bit size of the input to a third bit size, the third bit size being less than the second bit size,

providing a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to provide an output including an input to the mix and condense unit, receiving an input including the block of data expressed as the block of bits, inputting the block of bits into the expansion unit, thereby expanding the block of bits to a block of bits of the second bit size, combining, at the combining unit, the block of bits of the second bit size with a key, substituting bits including the input to the plurality of layers of S-boxes with bits including the output of the plurality of layers of S-boxes, outputting the output of the plurality of layers of S-boxes to the mix and condense unit, and mixing, at the mix and condense unit, the block of bits of the second bit size, and condensing, at the mix and condense unit, the block of bits of the second bit size to a block of bits of the third size, thereby producing an encrypted block of data, the encrypted block of data being expressed as a block of bits of the third bit size.

Moreover in accordance with a preferred embodiment of the present invention the plurality of layers of S-boxes includes two layers of S-boxes.

Further in accordance with a preferred embodiment of the present invention a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes including an input of the second layer of S-boxes.

Still further in accordance with a preferred embodiment of the present invention each S-box of the plurality of S-boxes includes one S-box described in the Serpent cipher specification.

Additionally in accordance with a preferred embodiment of the present invention the two layers of S-boxes include a first layer of S-boxes including 25 S-boxes and a second layer of S-boxes including 25 S-boxes.

Moreover in accordance with a preferred embodiment of the present invention the first bit size is equal to the third bit size.

Further in accordance with a preferred embodiment of the present invention the first bit size is equal to 64 bits.

Still further in accordance with a preferred embodiment of the present invention the second bit size is equal to 100 bits.

Additionally in accordance with a preferred embodiment of the present invention the third bit size is equal to 64 bits.

5 Moreover in accordance with a preferred embodiment of the present invention the combining unit is operative to perform a XOR operation.

Further in accordance with a preferred embodiment of the present invention the expansion unit includes a linear transformation.

10 Still further in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein each input bit influences at least two output bits.

Additionally in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein each bit of the key influences one output bit.

15 Moreover in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein any small set of input bits influences a larger set of output bits.

Further in accordance with a preferred embodiment of the present invention the linear transformation includes an operation wherein indices are selected so as to be spread equally between input bits and output bits.

20 Still further in accordance with a preferred embodiment of the present invention the mix and condense unit includes a plurality of layers, each layer among the plurality of layers including a plurality of mini-functions.

25 Additionally in accordance with a preferred embodiment of the present invention and wherein the plurality of layers of the MAC includes between 30 layers and 50 layers, inclusive.

Moreover in accordance with a preferred embodiment of the present invention a mini-function layer includes two micro-functions one balanced micro-function, and one non-linear micro-function.

30 Further in accordance with a preferred embodiment of the present invention the mini-function layer is operative to perform the following receiving an input, splitting the input, at a splitter, into a block of balancing bits and a block

of remaining input bits, executing the method of the non-linear micro-function on the block of remaining input bits, inputting the result of the non-linear micro-function into the balanced micro-function, executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits, and outputting a result.

Still further in accordance with a preferred embodiment of the present invention the method further includes performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

10 Additionally in accordance with a preferred embodiment of the present invention the invertible transformation includes an invertible transformation S-box.

Moreover in accordance with a preferred embodiment of the present invention the invertible transformation S-box includes a 2bit-to-2bit S-box.

15 Further in accordance with a preferred embodiment of the present invention The method further includes providing a first function F_i and a second function F_j , providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of the first function F_i , and the second function F_j , providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of the first function F_i , and the second function F_j , utilizing the round key generation function in at least a first round to generate a second round key for use in a second round, and utilizing the round mixing function in at least the first round to mix a first round key with a cipher state, wherein one of the following is performed in the first round the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state, and the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round

key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

There is also provided in accordance with still another preferred embodiment of the present invention a method of encrypting a block of data, the method including providing a combining unit operative to combine the block of data with a key, providing a mixing unit operative to mix the bits of a combined key and block of data, providing a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mixing unit and to provide an output including an input to the mixing unit, receiving an input including the block of data expressed as a block of bits, combining, at a combining unit, the block of bits with a key, substituting bits including the input to the plurality of layers of S-boxes with bits including the output of the plurality of layers of S-boxes, outputting the output of the plurality of layers of S-boxes as a second block of bits to the mixing unit, and mixing, at the mixing unit, the second block of bits, thereby producing an encrypted block of data, wherein the mixing unit includes a plurality of layers, each layer including a plurality of mini-functions.

Further in accordance with a preferred embodiment of the present invention the plurality of layers of S-boxes includes two layers of S-boxes.

Still further in accordance with a preferred embodiment of the present invention a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes including an input of the second layer of S-boxes.

Additionally in accordance with a preferred embodiment of the present invention each S-box of the plurality of S-boxes includes one S-box described in the Serpent cipher specification.

Moreover in accordance with a preferred embodiment of the present invention the two layers of S-boxes include a first layer of S-boxes including 25 S-boxes and a second layer of S-boxes including 25 S-boxes.

Further in accordance with a preferred embodiment of the present invention the plurality of layers of the mixing unit includes between 30 and 50 layers, inclusive.

Still further in accordance with a preferred embodiment of the present invention the combining unit is operative to perform a XOR operation.

Additionally in accordance with a preferred embodiment of the present invention a mini-function layer includes two micro-functions one balanced micro-function, and one non-linear micro-function.

Moreover in accordance with a preferred embodiment of the present invention the mini-function layer is operative to perform the following receiving an input, splitting the input, at a splitter, into a block of balancing bits and a block of remaining input bits, executing the method of the non-linear micro-function on the block of remaining input bits, inputting the result of the non-linear micro-function into the balanced micro-function, executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits, and outputting a result.

Further in accordance with a preferred embodiment of the present invention The method further includes performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

Still further in accordance with a preferred embodiment of the present invention the invertible transformation includes an invertible transformation S-box.

Additionally in accordance with a preferred embodiment of the present invention the invertible transformation S-box includes a 2bit-to-2bit S-box.

Moreover in accordance with a preferred embodiment of the present invention the method further includes providing an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being greater than the first bit size, and prior to the combining, inputting the block of bits into the expansion unit, and therein expanding the block of bits to a block of bits of the second bit size.

Further in accordance with a preferred embodiment of the present invention the method further includes after the mixing, condensing, at the mix and condense unit, the block of bits of the second bit size to a block of bits of a third size, thereby producing an encrypted block of data, the encrypted block of data
5 being expressed as a block of bits of the third bit size.

Still further in accordance with a preferred embodiment of the present invention the method further includes providing a first function F_i and a second function F_j , providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of
10 the first function F_i , and the second function F_j , providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of the first function F_i , and the second function F_j , utilizing the round key generation function in at least a first round to generate a second round key for use in a second round, and utilizing the round mixing function in at least the first
15 round to mix a first round key with a cipher state, wherein one of the following is performed in the first round the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state, and the round
20 key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

There is also provided in accordance with still another preferred
25 embodiment of the present invention a method including combining a control input derived from a right part of a Feistel-like structure with a transformation input including a left part of the Feistel-like structure, and producing an output including a combination of bits included in the control input and bits included in the transformation input, wherein no bit of the combination of bits includes a
30 linear combination of bits from the control input and bits from the transformation input.

Further in accordance with a preferred embodiment of the present invention, with respect to a fixed control input, the method includes an invertible method.

5 Still further in accordance with a preferred embodiment of the present invention the inverse of the method is not identical to the method.

Additionally in accordance with a preferred embodiment of the present invention the method includes a non-linear layer including at least one S-box.

10 Moreover in accordance with a preferred embodiment of the present invention the method also including a linear transformation of the control input and the transformation input.

Further in accordance with a preferred embodiment of the present invention the method also including splitting, at a control input splitter, the control input, into a plurality of control input sub-blocks, splitting, at a transformation input splitter, the transformation input, into a plurality of transformation input sub-blocks, linearly combining each one of the plurality of control input sub-blocks with a corresponding one of the plurality of transformation input sub-blocks, and joining the result of the linear combining at a output joiner.

20 Still further in accordance with a preferred embodiment of the present invention each one of the plurality of control input sub-blocks and a corresponding one of the plurality of transformation input sub-blocks include sub-blocks of the same size.

25 Additionally in accordance with a preferred embodiment of the present invention a first sub-block of the plurality of control input sub-blocks includes a sub-block of a different size than a second sub-block of the plurality of control input sub-blocks.

Moreover in accordance with a preferred embodiment of the present invention the transformation input splitter permutes the transformation input prior to the splitting at the transformation input splitter.

30 Further in accordance with a preferred embodiment of the present invention the output joiner permutes an output after the joining operation.

Still further in accordance with a preferred embodiment of the present invention the linearly combining includes $(A(C) \times I) \oplus C$, where C represents the control input sub-block, I represents the transformation input sub-block, and $A(C)$ includes a matrix depending on C , of size $m \times m$, where m is a size of the control input sub-block.

Additionally in accordance with a preferred embodiment of the present invention

$$A(C) = \begin{bmatrix} 1 & C[0] & 0 & C[3] \\ 0 & 1 & C[1] & 0 \\ 0 & 0 & 1 & C[2] \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ C[3] & 1 & 0 & 0 \\ 0 & C[1] & 1 & 0 \\ C[0] & 0 & C[2] & 1 \end{bmatrix}$$

where $C[0...3]$ include bits included in the control input.

Moreover in accordance with a preferred embodiment of the present invention the method also including a non-linear layer including at least one S-box.

Further in accordance with a preferred embodiment of the present invention an output from the linear transformation includes an input for the non-linear layer.

Still further in accordance with a preferred embodiment of the present invention an output from the non-linear layer includes a transformation input for the linear transformation.

Additionally in accordance with a preferred embodiment of the present invention at least one of the S-boxes includes an S-box according to the Serpent Cipher specification.

Moreover in accordance with a preferred embodiment of the present invention the S-box layer includes S-boxes which are simple to implement in hardware.

Further in accordance with a preferred embodiment of the present invention the method is cryptographically secure and non-involutable.

There is also provided in accordance with still another preferred embodiment of the present invention an encryptor for encrypting a block of data, the encryptor including a combining unit operative to combine a key with a block

of data, the block of data being expressed as a block of bits, a mix and condense unit operative to mix bits included in the block of bits among themselves, and a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to
5 provide an output including an input to the mix and condense unit, wherein a received input including the block of data expressed as the block of bits is combined, at the combining unit, with a key, and bits included in the combined block of bits are, in each layer of the plurality of layers of S-boxes, substituted for other bits, thereby providing output bits, bits in the output bits are mixed among
10 themselves at the mix and condense unit, and the mix and condense unit includes a plurality of layers, each layer among the plurality of layers including a plurality of mini-functions.

Further in accordance with a preferred embodiment of the present invention the encrypting cannot be efficiently implemented except on specialized
15 hardware.

There is also provided in accordance with still another preferred embodiment of the present invention an encryptor for encrypting a block of data, the encryptor including an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit
20 size being greater than the first bit size, thereby producing an expanded block of data, a combining unit operative to receive the expanded block of data from the expansion unit and combine the expanded block of data with a key thereby producing a combined expanded block of data of the second bit size, a mix and condense unit operative to mix the bits of the combined expanded block of data of
25 the second bit size and condense the bit size of the combined expanded block of data of the second bit size to a third bit size, the third bit size being less than the second bit size, and a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to provide an output including an input to the mix and condense
30 unit, wherein the mix and condense unit includes a plurality of layers, each layer among the plurality of layers including a plurality of mini-functions.

Further in accordance with a preferred embodiment of the present invention the encryptor cannot be implemented except on specialized hardware.

There is also provided in accordance with still another preferred embodiment of the present invention an encryptor operative to encrypt a block of data, the encryptor including a combining unit operative to combine the block of data with a key and produce a combined key and block of data, a mixing unit operative to mix the bits of the combined key and block of data, and a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to provide an output including an input to the mix and condense unit, wherein the mixing unit includes a plurality of layers, each layer including a plurality of mini-functions.

There is also provided in accordance with still another preferred embodiment of the present invention an apparatus including a combiner operative to combine a control input derived from a right part of a Feistel-like structure with a transformation input including a left part of the Feistel-like structure, an outputter operative to producing an output including a combination of bits included in the control input and bits included in the transformation input, and a plurality of layers of S-boxes, the S-boxes operative to receive an input including an input which has not yet been input into the mix and condense unit and to provide an output including an input to the mix and condense unit, wherein no bit of the combination of bits includes a linear combination of bits from the control input and bits from the transformation input.

There is also provided in accordance with still another preferred embodiment of the present invention a cipher device including a combine key right (CKR) unit, operative to receive a first portion of a message and to combine the first portion of the message with a key, the CKR including an expansion unit operative to expand the first portion of the message to a number of bits appropriate to a key size, thereby producing an expanded first portion of the message, a combining function unit operative to receive the expanded first portion of the message and combine the expanded first portion of the message with the key, thereby producing a combined output, a mixing and condensing function unit operative to receive the combined output and condensing the combined output,

and a plurality of layers of S-boxes disposed at least one of before the mixing and condensing function unit, each S-box among the plurality of S-boxes included in each of the plurality of layers of the S-boxes being operative to receive an input string of bits and substitute the input string of bits with an output string of bits, and
5 a combine right - left (CRL) unit, operative to combine an output of the CKR with a second portion of the message.

Further in accordance with a preferred embodiment of the present invention the first portion of the message includes a right portion of the message, and the second portion of the message includes a left portion of the message.

10 Still further in accordance with a preferred embodiment of the present invention the first portion of the message includes a left portion of the message, and the second portion of the message includes a right portion of the message.

Additionally in accordance with a preferred embodiment of the present invention the plurality of layers of S-boxes includes two layers of S-boxes.
15

Moreover in accordance with a preferred embodiment of the present invention a diffusion layer, disposed between each layer of the plurality of S-boxes routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes including an input of the second layer of S-boxes.

20 There is also provided in accordance with still another preferred embodiment of the present invention in a cipher device including a combine key right (CKR) unit, operative to receive a first portion of a message and to combine the first portion of the message with a key, the CKR including an expansion unit operative to expand the first portion of the message to a number of bits appropriate
25 to a key size, thereby producing an expanded first portion of the message, a combining function unit operative to receive the expanded first portion of the message and combine the expanded first portion of the message with the key, thereby producing a combined output, a mixing and condensing function unit operative to receive the combined output and condensing the combined output,
30 and a combine right - left (CRL) unit, operative to combine an output of the CKR with a second portion of the message, an improvement including adding a plurality of layers of S-boxes disposed before the mixing and condensing function unit,

thereby increasing a level of confusion of an output ciphertext and making the ciphertext more resistant to a differential cryptographic attack.

Further in accordance with a preferred embodiment of the present invention the first portion of the message includes a right portion of the message,
5 and the second portion of the message includes a left portion of the message.

Still further in accordance with a preferred embodiment of the present invention the first portion of the message includes a left portion of the message, and the second portion of the message includes a right portion of the message.

10 Additionally in accordance with a preferred embodiment of the present invention the plurality of layers of S-boxes includes two layers of S-boxes.

Moreover in accordance with a preferred embodiment of the present invention a diffusion layer disposed between each layer of the plurality of S-boxes routes the output of a first layer of S-boxes to a second layer of S-boxes, the output
15 of the first layer of S-boxes including an input of the second layer of S-boxes.

BRIEF DESCRIPTION OF THE DRAWINGS AND APPENDICES

The present invention will be understood and appreciated more fully from the following detailed description, taken in conjunction with the drawings in which:

5 Fig. 1 is an illustration of a hardened Feistel-like structure constructed and operative in accordance with a preferred embodiment of the present invention;

 Fig. 2A is an illustration of a Combine Key RightPart function comprised in the hardened Feistel-like structure of Fig. 1;

10 Fig. 2B is an illustration of an alternative preferred embodiment of the Combine Key RightPart function comprised in the hardened Feistel-like structure of Fig. 1;

 Fig. 3 is an illustration of a preferred implementation of hardware for a RightPart Expansion Function comprised in the Combine Key RightPart
15 function of Figs. 2A and 2B;

 Fig. 4 is an illustration of a preferred embodiment of a mini-function, the mini-function serving as a building block for a Mix and Condense function comprised in the Combine Key RightPart function of Figs. 2A and 2B;

 Fig. 5 is an illustration of a Combine RightPart Combine LeftPart
20 function comprised in the hardened Feistel-like structure of Fig. 1;

 Fig. 6 is an illustration of one preferred implementation of a linear layer in the Combine RightPart Combine LeftPart function of Fig. 5;

 Fig. 7 is an illustration of one preferred implementation of an S-boxes layer in the Combine RightPart Combine LeftPart function of Fig. 5;

25 Fig. 8 is an illustration of one preferred implementation of a key expansion function comprised in the hardened Feistel-like structure of Fig. 1;

 Fig. 9 is an illustration of one preferred implementation of round key generation utilizing the Mix and Condense function in the key expansion function of Fig. 8;

30 Figs. 10 - 13 are simplified flowchart illustrations of preferred alternative methods of operation of the hardened Feistel-like structure of Fig. 1, in accordance with preferred embodiments thereof;

Fig. 14 is a simplified block diagram illustration of a system for robust cipher design constructed and operative in accordance with a preferred embodiment of the invention described in Appendix B;

5 Fig. 15 is a time line showing one preferred implementation of the relationship between key expansion and encryption rounds in a cipher designed according to the method of Fig. 14;

Fig. 16 is a simplified block diagram illustration depicting the use of MUX and DEMUX modules in a preferred implementation of the method of Fig. 14;

10 Fig. 17 is a simplified block diagram illustration of a preferred implementation of a round key generation function operative to generate round keys in a cipher designed according to the method of Fig. 14;

15 Fig. 18 is a simplified block diagram illustration of four rounds of a typical Feistel block cipher constructed and operative in accordance with the system of Fig. 14;

Fig. 19 is a simplified block diagram illustration of four rounds of a typical AES-like block cipher constructed and operative in accordance with the system of Fig. 14;

20 Fig. 20 is a simplified block diagram illustration of eight rounds of a typical Feistel block cipher constructed and operative in accordance with an alternative preferred embodiment of the system of Fig. 14;

Fig. 21 is a simplified block diagram illustration of eight rounds of a typical AES-like block cipher constructed and operative in accordance with an alternative preferred embodiment of the system of Fig. 14;

25 Fig. 22 is a simplified block diagram illustration of eight rounds of a typical Feistel block cipher constructed and operative in accordance with yet another alternative preferred embodiment of the system of Fig. 14;

30 Fig. 23 is a simplified block diagram illustration of eight rounds of a typical AES-like block cipher constructed and operative in accordance with yet another alternative preferred embodiment of the system of Fig. 14;

Fig. 24 is an illustration of a hardened Feistel-like structure constructed and operative in accordance with a preferred embodiment of the present invention;

5 Fig. 25 is an illustration of an alternative preferred embodiment of the hardened Feistel-like structure of Fig. 24;

Fig. 26 is a simplified block diagram of a preferred implementation of a MixKey function of the system of Fig. 24; and

Fig. 27 is a simplified block diagram of a CombParts function of the system of Fig. 24.

10

The following Appendices may be helpful in understanding certain preferred embodiments of the present invention:

15 Appendix A comprises table describes one implementation of the relationship between bits output from the first layer of S-boxes and input into the second layer of S-boxes in the system of Fig. 2B.

Appendix B is a description of a method for robust cipher design, comprising a preferred method of key expansion and set up and a preferred implementation of a round key encryption function, the method of Appendix B comprising a preferred implementation of the Feistel-like structure of Fig. 1;

20 Appendix C is a copy of Appendix A.5 of the Serpent Cipher specification, describing S-boxes S_0 through S_7 of the Serpent Cipher; and

Appendix D comprises a description of certain alternative preferred embodiments of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Reference is now made to Fig. 1, which is an illustration of a hardened Feistel-like structure 100 constructed and operative in accordance with a preferred embodiment of the present invention. It is appreciated that Fig. 1 provides an illustration of data structures and methods for implementing an encryption network, the illustration being drawn in a format which is well known in the art. Fig. 1 depicts two rounds of the hardened Feistel-like structure 100, it being appreciated that a plurality of rounds comprising more than two rounds is preferred, similarly to the plurality of rounds known in the prior art in the case of Feistel-like networks.

The Feistel-like structure 100 of Fig. 1 comprises a Combine Key RightPart (CKR) function 110, a preferred implementation of which is described below with reference to Figs. 2A and 2B, and a Combine RightPart Combine LeftPart (CRL) function 120, a preferred implementation of which is described below with reference to Fig. 5. A preferred implementation of a key expansion function (not depicted in Fig. 1), operative to provide a round key (RK_j, RK_{j+1}) for each round of the Feistel-like structure 100 is described below with reference to Fig. 8.

In each round of the hardened Feistel-like structure 100, two halves of a plaintext, left and right, depicted as L and R, are operated on by the CKR function 110 and the CRL function 120. It is appreciated that in each round, L and R preferably have an identical size of 64 bits. It is nevertheless appreciated that L and R may be any equal size, and 64 bits is used herein as an example. It is further appreciated that the size of the round key, RK_j , is described herein as 100 bits by way of example, only. RK_j may be any appropriate size.

It is appreciated that the plurality of rounds may preferably be preceded by preprocessing of L and R. For example, L and R may preferably be permuted according to a pre-defined permutation in the same manner the DES block cipher permutes the input before the first round (refer to FIPS 46-3). It is further appreciated that after the plurality of rounds are completed, an encrypted output of the hardened Feistel-like structure 100 may be post-processed. For example, output may preferably be further permuted according to a pre-defined

permutation in the same manner the DES block cipher permutes the state after the 16th round (refer to FIPS 46-3).

For any given n rounds of the hardened Feistel-like structure 100, a particular round (first round, last round, or any other round) may preferably differ
5 from the other n-1 rounds.

The Feistel-like structure 100 preferably uses a 128-bit key to encrypt and decrypt 128-bit blocks. The number of rounds (RN) is preferably RN between 40 and 50, inclusive.

It is appreciated that the Feistel-like structure 100 is preferably less
10 efficient if implemented in software.

The Feistel-like structure 100 preferably utilizes CKR 110 to integrate a round key with a right half of a state and the function CRL 120 to combine the result of the key integration with a left half of the state. The left and right halves of the state are referred below as L and R, respectively.

Reference is now made to Fig. 2A, which is an illustration of a
15 Combine Key RightPart (CKR) function 110 comprised in the hardened Feistel-like structure of Fig. 1.

The CKR function 110 preferably comprises the following operations:

- 20 1. RExp (Right Part Expansion) 210 preferably expands the right half R from 64 to 100 bits;
2. Using a XOR operation 220, a 100 bit round key, RK_i , is preferably combined with the expanded 100 bit right half;
3. MCF (Mix and Condense Function) 230 preferably mixes the
25 100 bit result of RExp 210 and, preferably in a pseudorandom fashion, preferably condenses the mixed 100 bits to 64 bits.

Reference is now made to Fig. 2B, which is an illustration of an alternative preferred embodiment of the Combine Key RightPart function comprised in the hardened Feistel-like structure of Fig. 1. In the preferred
30 embodiment of the CKR depicted in Fig. 2B, a plurality of layers of S-boxes 310, 330 is added, before the MCF 230. The plurality of layers of S-boxes 310, 330 is operative to receive the output of the 100 bit result of the XOR operation 220. As

is well known in the art, S-boxes substitute bits comprising a set of input bits with a set of output bits, the substitution preferably increasing a level of confusion of an output ciphertext.

In one preferred embodiment of the present invention, depicted in
5 Fig. 2B, the plurality of layers of S-boxes 310, 330 comprises a first layer of S-boxes 310, a diffusion layer 320, and a second layer of S-boxes 330. Each of the first layer of S-boxes 310 and the second layer of S-boxes 330 comprises 25 S-boxes, each S-box operative to receive a 4 bit input, and produce a 4 bit output.

The diffusion layer 320 connects the first layer of S-boxes 310 and
10 the second layer of S-boxes 330, such that a large number of S-boxes in the second layer of S-boxes 330 are affected by substitutions in the first layer of S-boxes 310.

Those skilled in the art will appreciate that, in principle, as many layers of S-boxes as desired may be added. However, each additional layer slows down processing time. In one preferred embodiment of the present invention, the
15 25 S-boxes are S-boxes from the Serpent Cipher, as discussed below. The Serpent Cipher and the S-boxes thereof are discussed at greater length below, with reference to Fig. 7.

Those skilled in the will appreciate that any appropriate S-boxes may be used in the preferred embodiment depicted in Fig. 2B. The above
20 discussion wherein the S-boxes in the preferred embodiment depicted in Fig. 2B are described as the Serpent Cipher S-boxes is by way of example, and not meant to be limiting.

Reference is now made to Appendix A, which comprises a table describing one implementation of the relationship between bits output from the
25 first layer of S-boxes 310 and input into the second layer of S-boxes 330 in the system of Fig. 2B

Reference is now made to Fig. 3, which is an illustration of a preferred implementation of hardware for a RightPart Expansion Function comprised in the Combine Key RightPart function of Figs. 2A and 2B. It is
30 appreciated that Fig. 3 provides an illustration of a preferred implementation of hardware structures and methods for implementing an expansion function, the illustration being drawn in a format which is well known in the art. RExp 210

(Figs. 2A and 2B) preferably uses a linear transformation to expand the 64 bit R into a 100 bit expanded RightPart, where each of the 100 bit output bits is the result of a XORing of 2 or 3 input bits.

Indices implemented in the proposed hardware of Fig. 3 are preferably selected pseudo-randomly with the following constraints:

1. Each one of the 64 input bits of the R preferably influences at least two output bits;
2. Each bit of the 100 bit round key preferably influences exactly one output bit;
- 10 3. Indices are preferably selected so as to be spread equally between the input and output bits, thereby avoiding a situation where a small number of input bits influence only a small number of output bits; and
4. Any small set of input bits preferably influences a larger set of output bits.

Those skilled in the art will appreciate that error correcting codes, such as the well known Hamming error correcting code, share similar design criteria with the indices implemented in the proposed hardware and thus, error correcting codes may be well suited for use as the indices implemented in the proposed hardware.

It is preferable that the RExp function 210 (Figs. 2A and 2B) and the subsequent XOR 220 operation (with the round key) balance between a proper mixing of the round key with the right part and a time-efficient implementation of the mixing, thereby allowing a hardware implementation of both the RExp function 210 (Figs. 2A and 2B) and the XOR 220 operation that preferably comprises only two layers of XOR operations (and, in some preferred embodiments, an additional layer of NOT gates).

Returning to the discussion of Fig. 2A, the MCF function 230 is now discussed. The 100 bit expanded right half, after XORing with the 100 bit round key RK_j , is preferably input into the MCF function 230. A 100 bit result of the XORing is preferably reduced and condensed into a 64-bit temporary result, which is used later as a control input of the CRL function (described with

reference to Fig. 5). The MCF function 230 is preferably critical in making the Feistel-like structure 100 (Fig. 1) emulation resistant.

Reference is now made to Fig. 4, which is an illustration of a preferred embodiment of the mini-function, the mini-function serving as a building
5 block for the MCF function 230 (Figs. 2A and 2B) comprised in the CKR function 110 of Figs. 2A and 2B.

The MCF function preferably uses between round key generation function and 50, inclusive, layers of mini-functions 400, where each of the mini-functions 400 preferably comprises two micro-functions, a balanced micro-
10 function BF 410 and a non-linear micro-function NLF 420.

A balanced micro-function BF 410 is defined as follows: a set of the input bits for the balanced function are denoted as the balancing set and for every selection of the other input bits, a uniform distribution on the balancing set guarantees uniform distribution on the output (i.e., a uniform distribution of zeros
15 and ones input guarantees a uniform distribution of zeros and ones output). For example and without limiting the generality of the foregoing, a XOR operation is a balanced function for which each of the input bits is a balancing set.

The mini-functions 400 are preferably designed as follows:

the input bits are preferably input into a splitter 415, which
20 splits the balancing set of bits from the other input bits;

NLF 420 is preferably executed on the other input bits; and

afterwards BF 410 is preferably executed on the output of NLF 420 and on the balancing set of bits, received from the splitter 415.

In some preferred embodiments of the present invention, the
25 balancing set of bits goes through a third type of micro-functions, comprising an invertible transformation, such as a 2bit-to-2bit S-box, where the balancing set comprises 2 bits. Putting the balancing set through the invertible transformation is preferably performed simultaneously with the NLF, and thus, employing the third micro-function can be performed preferably without cost in execution time.

30 For example and without limiting the generality of the foregoing, the following functions process 3-bit inputs (according to the design criteria stated immediately above):

(input1 $\vee$ input2) $\oplus$ input3;

NOT ((input1 $\wedge$ input2) $\oplus$ input3);

The Majority function; and

MUX, where a single bit selects which of the two other input

5 bits to output.

The mini-functions 400 in layer i preferably receive inputs from the outputs of the mini-functions 400 in layer i-1. Selection of which output of layer i-1 goes to which input of layer i is preferably performed in a manner that preferably maximizes the mixing between layers and thus preferably avoids
10 localization effects.

It is preferable that the exact MCF 230 (Figs. 2A and 2B) utilized is automatically generated during design. However, the MCF utilized preferably passes several statistical tests measuring correlation between output bits (in particular, linear correlations). The statistical tests are preferably not restricted to
15 input and output, but preferably also measure correlations in internal layers between inputs and outputs. In addition, it is preferable that it is not possible to express any small set of output bits of MCF 230 (Figs. 2A and 2B) as a short expression of input bits of MCF 230 (Figs. 2A and 2B).

Reference is now made to Appendix B, which is a description of a
20 method for robust cipher design, comprising a preferred method of key expansion and set up and a preferred implementation of a round key encryption function, the method of Appendix B comprising a preferred implementation of the Feistel-like structure of Fig. 1. In order to harden the Feistel-like structure 100 (Fig. 1) and prevent single points of failure, MCF 230 (Figs. 2A and 2B) preferably is
25 implemented in two versions. The two versions are preferably used in an alternating manner throughout the rounds of the Feistel-like structure 100 (Fig. 1). It is appreciated that even if one of the two versions is found to be “faulty”, the Feistel-like structure 100 (Fig. 1) as a whole preferably remains strong. A “faulty” function in the present context is either a cryptographically weak function (e.g.,
30 having strong linear or differential properties) or a function that is easy to emulate in software.

Reference is now made to Fig. 5, which is an illustration of a Combine RightPart Combine LeftPart (CRL) function 120 comprised in the hardened Feistel-like structure 100 of Fig. 1. The CRL 120 function combines the 64-bit result of the MCF 230 as the last stage of the CKR 110 with the unchanged
 5 64-bit left half L_i to get a new 64-bit pseudo-random right half, R_{i+1} .

The CRL function 120 preferably complies with the following design criteria:

1. CRL 120 is preferably invertible in a second parameter when fixing a first parameter. That is, there shall be ICRL, such that, for every X, Y,
 10 ICRL(X, CRL(X, Y))=Y, where the CKR 110 result is used as the first parameter X (also denoted hereinafter as the “control input”) and the left half, L_i , is used as the second parameter Y (also denoted hereinafter as the “transform input”).

2. CRL 120 is preferably not an involution. That is, ICRL preferably differs significantly from CRL 120 (as opposed, for example, to the
 15 XOR function that is used in DES).

The CRL function 120 preferably comprises two stages, each stage working on small sub-blocks. In preferred embodiment of the present invention, each sub-block comprises 4 bits. After each of the stages, a permutation is preferably applied to the result, breaking any locality effect of working on small
 20 sub-blocks.

The first stage comprises a linear layer LL 510 that mixes the control input with the transform input.

After LL 510, a bit-permutation PL 520 is preferably applied to the result of the LL 510.

25 Afterwards, the output of PL 520 is preferably input into an S-boxes layer SL 530, comprised of sixteen 4-bit to 4-bit S-boxes.

Finally, a bit-permutation (not depicted) is preferably applied to the output of SL 530.

Reference is now made to Fig. 6, which an illustration of one
 30 preferred implementation of the linear layer 510 in the Combine RightPart Combine LeftPart (CRL) function 120 of Fig. 5. LL 510 comprises a first splitter 610 which splits transform input, L_i , into 4-bit micro-blocks. Similarly, a second

splitter splits control input into 4-bit micro-blocks. The 4-bit micro-blocks resulting from the control input are preferably used to determine a linear transformation (LT). The determined transformation is preferably applied to the input 4-bit micro-blocks, thereby producing a 4-bit output micro-block. Linear transform operations of the control data 4-bit micro-blocks and the transform data 4-bit micro-blocks are depicted in Fig. 6 as "LT".

For the control bits $C[0..3]$ and the input bits $I[0..3]$ the linear transformation preferably $O = (A(C) \times I) \oplus C$ where $A(C)$ is a linear transformation depending on control input C :

$$10 \quad A(C) = \begin{bmatrix} A_{11}(C) & A_{12}(C) & A_{13}(C) & A_{14}(C) \\ A_{21}(C) & A_{22}(C) & A_{23}(C) & A_{24}(C) \\ A_{31}(C) & A_{32}(C) & A_{33}(C) & A_{34}(C) \\ A_{41}(C) & A_{42}(C) & A_{43}(C) & A_{44}(C) \end{bmatrix}$$

for A_{ij} s which are 4bit-to-1bit functions which are applied to the control input, and O is the resulting output.

$A(C)$ is invertible; that is there exists $B(C)$, such that:

$$B(C) = \begin{bmatrix} B_{11}(C) & B_{12}(C) & B_{13}(C) & B_{14}(C) \\ B_{21}(C) & B_{22}(C) & B_{23}(C) & B_{24}(C) \\ B_{31}(C) & B_{32}(C) & B_{33}(C) & B_{34}(C) \\ B_{41}(C) & B_{42}(C) & B_{43}(C) & B_{44}(C) \end{bmatrix}$$

$$15 \quad \text{such that for every control input } C: A(C) \times B(C) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \text{ that is } A(C) \text{ is the}$$

inverse of $B(C)$.

In preferred embodiments of the present invention $A(C)$ comprises:

$$\begin{aligned}
& \begin{bmatrix} A_{11}(C) & A_{12}(C) & A_{13}(C) & A_{14}(C) \\ A_{21}(C) & A_{22}(C) & A_{23}(C) & A_{24}(C) \\ A_{31}(C) & A_{32}(C) & A_{33}(C) & A_{34}(C) \\ A_{41}(C) & A_{42}(C) & A_{43}(C) & A_{44}(C) \end{bmatrix} = \\
& \begin{bmatrix} 1 & C[0] & 0 & C[3] \\ 0 & 1 & C[1] & 0 \\ 0 & 0 & 1 & C[2] \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ C[3] & 1 & 0 & 0 \\ 0 & C[1] & 1 & 0 \\ C[0] & 0 & C[2] & 1 \end{bmatrix} \\
& = \begin{bmatrix} C[0] \oplus C[3] & C[0] & C[2]C[3] & C[3] \\ C[3] & C[1] & C[1] & 0 \\ C[0]C[2] & C[1] & C[2] & C[2] \\ C[0] & 0 & C[2] & 1 \end{bmatrix}
\end{aligned}$$

(equation 1)

It is appreciated that if the transformation $A(C)$ is used during decryption, then during encryption the inverse transformation of $A(C)$ is used. In particular, if $A(C)$ is as described in equation 1, then, since both matrices comprising control bits used in equation 1 are involutions, the inverse transformation $B(C)$ is the composition of the transformations in reversed order. The results of all linear transformations are preferably input into join function 630. Join function 630 preferably joins the results of all 16 linear transformations into one 64 bit value.

10 The 64 bit output of join function 630 is preferably input into bit-permutation PL 520, thereby producing a 64 bit permuted output. Bit-permutations are well known cryptographic structures.

Reference is now made to Fig. 7, which is an illustration of one preferred implementation of an S-boxes layer in the Combine RightPart Combine LeftPart (CRL) function 120 of Fig. 5. The layer of S-boxes SL 530 (Fig. 5) preferably comprises 4-bit to 4-bit S-boxes, which are preferably simple to implement in hardware and still comprise a significant contribution to non-linearity of the hardened Feistel-like structure 100 (Fig. 1). The 64-bit input is input into an S-box splitter 710. The S-box splitter 710 preferably divides the 64-bit input into 16 4-bit micro-blocks. The 16 4-bit micro-blocks go through sixteen

S-boxes 720. Output from the sixteen S-boxes 720 is all mixed in a bit permutation join function 730.

The specification of the Serpent cipher (refer to www.ftp.cl.cam.ac.uk/ftp/users/rja14/serpent.pdf) describes eight 4bit-to-4bit S-boxes, which were optimized against linear and differential attacks. It is the opinion of the inventors of the present invention that the S-boxes described in the specification of the Serpent cipher should preferably be used in the hardened Feistel structure 100 (Fig. 1) described herein. Reference is now made to Appendix C which is a copy of Appendix A.5 of the Serpent Cipher specification, describing S-boxes S_0 through S_7 of the Serpent Cipher.

Reference is now made to Fig. 8, which is an illustration of one preferred implementation of a key expansion function 800 comprised in the hardened Feistel-like structure 100 of Fig. 1. The key setup function 800 preferably extends a 128-bit key to RN 100-bit round keys (RN is the number of rounds). The key expansion function is preferably designed according to the following principles:

1. Preferably reuse available hardware functions.
2. Preferably enhance robustness of the hardened Feistel-like structure 100 (Fig. 1), as discussed above, with reference to the discussion of Appendix B.
3. Preferably allow both forward and backward generation of the round keys.

As discussed above, with reference to the discussion of Appendix B, the key expansion function 800 takes advantage of the fact that the MCF preferably comprises two variations; one variation is preferably active during any round in the MCF function for the CKR 110 (Figs. 2A and 2B), while the other variation is preferably available for use. The key expansion function 800 therefore preferably uses the available MCF function in order to generate the round keys in a cryptographically secure manner.

Imitating a typical design for stream ciphers, the key setup function 800 preferably employs two functions; a first function, state update 810, is preferably operative to update a state. The second function, round key generation

830, preferably derives a new round key 840 from the new state. The state update 810 and round key generation 830 functions are executed in an alternating order generating round keys 840 which are preferably cryptographically decoupled from the key itself, as well as from each other.

5 The state of the key setup is preferably a 128-bit shift register. The 128-bit shift register is initialized 850 with the 128-bit key. The state update function 810 preferably comprises a circular rotation of the 128-bit register. It is appreciated that the number of rounds (RN) is preferably smaller than the size of the 128-bit register, and thus the state update function preferably does not loop
10 during a round.

 During decryption, in order to get the round keys in the proper order (reverse order from the order used during encryption), a decrypter preferably receives the state in reverse order used during encryption. In some preferred embodiments of the present invention, decryption preferably begins with shifting
15 the shift register as many times as needed in order to get the state appropriate for the last round key. Each subsequent round then preferably shifts the state in the opposite direction to the direction used to circularly shift the state during encryption.

 It is appreciated that replacement of a short LFSR (left shift register)
20 with 2-3 smaller LFSRs may be preferable. If 2-3 smaller LFSRs are utilized, the decryption key is the result of applying a linear transformation (calculated in advance and hard-wired) on the encryption key, and then the LFSRs are preferably rolled back to get the round keys in the reverse order.

 In order to avoid weak keys and slide attacks, an additional XOR
25 with a predefined round string may preferably be applied after the state update function 810.

 Reference is now made to Fig. 9, which is an illustration of one preferred implementation of round key generation 830 utilizing the Mix and Condense function (MCF) 230 (Figs. 2A and 2B) in the key expansion function
30 800 of Fig. 8. The round key generation 830 function inputs the 128-bit state into the MCF 230 (Figs. 2A and 2B) and takes the 100-bit output as the next round key, as discussed above with reference to Appendix B.

The following are design principles for selecting the order of using the MCF variations in the key setup and the round operation:

1. Preferably allow a smooth pipeline between the round operation and the key setup. Specifically, have both functions active together where one generates the key for the next round and the other is used for the round operation itself.

2. Preferably use as many different combinations as possible, maximizing the distribution of the “responsibility” for both security and emulation resistance.

As discussed in greater detail in Appendix B, for two MCF functions A and B, the round operation preferably uses A and B in the following order: A A B B A A B B A A B B ...

The key setup operation uses the function that is left available, i.e., B on rounds 1, 2 (preparing the keys for round 2, 3), A on round 3, 4 (preparing the key for round 4, 5) etc.

Thus the rounds of the hardened Feistel-like structure 100 (Fig. 1) have the following combinations as round key derivation and round operation:

Round $4t+1$: AA;

Round $4t+2$: BA;

Round $4t+3$: BB; and

Round $4t+4$: AB.

Alternative preferred implementations are discussed at length in Appendix B.

The implementation of MCF 230 (Figs. 2A and 2B) that is preferably used in the round operation and the MCF that is used in the key expansion have different sizes of inputs and outputs. Specifically, a 128 bit value is preferably input in order to produce a 100 bit output for key setup, and a 100 bit value is preferably input in order to produce a 64 bit output for a round operation.

In order to use the same hardware for both operations, the implemented MCFs are preferably implantations of 100 bits going to 128 bits going to 100 bits going to 64 bits, where most of the layers are in the 128 bits going to 100 bits part. Thus, the round operation uses the whole function and the key expansion uses only the middle part of the function. The blowing effect herein

described also contributes to preferably making the function hard to emulate in software.

Reference is now made to Figs. 10 - 13, which are simplified flowchart illustrations of preferred alternative methods of operation of the hardened Feistel-like structure of Fig. 1, in accordance with preferred
5 embodiments thereof. The methods of Figs. 10 - 13 are believed to be self explanatory with reference to the above discussion.

Reference is now made to Appendix D, which comprises a description of certain alternative preferred embodiments of the present invention.

10 It is appreciated that software components of the present invention may, if desired, be implemented in ROM (read only memory) form. The software components may, generally, be implemented in hardware, if desired, using conventional techniques.

15 It is appreciated that various features of the invention which are, for clarity, described in the contexts of separate embodiments may also be provided in combination in a single embodiment. Conversely, various features of the invention which are, for brevity, described in the context of a single embodiment may also be provided separately or in any suitable subcombination.

20 It will be appreciated by persons skilled in the art that the present invention is not limited by what has been particularly shown and described hereinabove. Rather the scope of the invention is defined only by the claims which follow:

What is claimed is:

CLAIMS

1. A method of encrypting a block of data, the method comprising:
5 providing a combining unit operative to combine a key with a block of data, the block of data expressed as a block of bits;
providing a mix and condense unit (MAC) operative to mix bits comprised in the block of bits among themselves;
providing a plurality of layers of S-boxes, the S-boxes operative to
10 receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input to the mix and condense unit;
receiving an input comprising the block of data expressed as the block of bits;
15 combining, at the combining unit, the block of bits with a key;
receiving an output of the combining unit as an input;
substituting bits comprising the input to the plurality of layers of S-boxes with bits comprising the output of the plurality of layers of S-boxes;
outputting the output of the plurality of layers of S-boxes to the mix
20 and condense unit; and
mixing, at the mix and condense unit, the output of the plurality of layers of S-boxes, thereby producing an encrypted block of bits.
2. The method according to claim 1 and wherein the plurality of layers
25 of S-boxes comprises two layers of S-boxes.
3. The method according to either claim 1 or claim 2 and wherein a
diffusion layer, disposed between each layer of the plurality of S-boxes, routes the
output of a first layer of S-boxes to a second layer of S-boxes, the output of the
30 first layer of S-boxes comprising an input of the second layer of S-boxes.

4. The method according to any of claims 1 - 3 and wherein each S-box of the plurality of S-boxes comprises one S-box described in the Serpent cipher specification.
- 5 5. The method according to any of claims 2 - 4 and wherein the two layers of S-boxes comprise a first layer of S-boxes comprising 25 S-boxes and a second layer of S-boxes comprising 25 S-boxes.
6. The method according to any of claims 1 - 5 and wherein the
10 combining unit is operative to perform a XOR operation.
7. The method according to any of claims 1 - 6 and wherein the method of encrypting cannot be efficiently implemented except on specialized hardware.
- 15 8. The method according to any of claims 1 - 7 and wherein the MAC comprises a plurality of layers of mini-functions.
9. The method according to claim 8 and wherein the plurality of layers
20 of the MAC comprises between 30 layers and 50 layers, inclusive.
10. The method according to either of claim 8 or claim 9 and wherein a mini-function layer comprises two micro-functions:
one balanced micro-function; and
25 one non-linear micro-function.
11. The method according to claim 10 and wherein the mini-function layer is operative to perform the following:
receiving an input;
30 splitting the input, at a splitter, into a block of balancing bits and a block of remaining input bits;

executing the method of the non-linear micro-function on the block of remaining input bits;

inputting the result of the non-linear micro-function into the balanced micro-function;

- 5 executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits; and
 outputting a result.

12. The method according to claim 11 and further comprising
10 performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

13. The method according to claim 12 and wherein the invertible transformation comprises an invertible transformation S-box.

14. The method according to claim 13 and wherein the invertible transformation S-box comprises a 2bit-to-2bit S-box.

15. The method according to any of claims 1 - 14 and further
20 comprising:

 providing a first function F_i and a second function F_j ;

 providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of:

 the first function F_i ; and

- 25 the second function F_j ;

 providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of:

 the first function F_i ; and

 the second function F_j ;

- 30 utilizing the round key generation function in at least a first round to generate a second round key for use in a second round; and

utilizing the round mixing function in at least the first round to mix a first round key with a cipher state,

wherein one of the following is performed in the first round:

the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state; and

the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

16. A method of encrypting a block of data, the method comprising:

providing an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being greater than the first bit size;

providing a combining unit operative to combine an expanded block of data with a key;

providing a mix and condense unit (MAC) operative to mix the bits of a combined expanded block of data of the second bit size and condense the bit size of the input to a third bit size, the third bit size being less than the second bit size;

providing a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input, expressed as a block of data, to the mix and condense unit;

receiving an input comprising the block of data expressed as the block of bits;

inputting the block of bits into the expansion unit, and therein expanding the block of bits to a block of bits of the second bit size;

combining, at the combining unit, the block of bits of the second bit size with a key;

substituting bits comprising the input to the plurality of layers of S-boxes with bits comprising the output of the plurality of layers of S-boxes;

outputting the output of the plurality of layers of S-boxes to the mix and condense unit; and

5 mixing, at the mix and condense unit, the output of the plurality of layers of S-boxes, the output of the plurality of layers of S-boxes comprising a block of bits of the second bit size; and

condensing, at the mix and condense unit, the block of bits of the second bit size to a block of bits of the third size, thereby producing an encrypted
10 block of data, the encrypted block of data being expressed as a block of bits of the third bit size,

wherein the mix and condense unit comprises a plurality of layers, each layer among the plurality of layers comprising a plurality of mini-functions.

15 17. The method according to claim 16 and wherein the plurality of layers of S-boxes comprises two layers of S-boxes.

18. The method according to either claim 16 or claim 17 and wherein a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the
20 output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes comprising an input of the second layer of S-boxes.

19. The method according to any of claims 16 - 18 and wherein each S-box of the plurality of S-boxes comprises one S-box described in the Serpent
25 cipher specification.

20. The method according to any of claims 17 - 19 and wherein the two layers of S-boxes comprise a first layer of S-boxes comprising 25 S-boxes and a second layer of S-boxes comprising 25 S-boxes.

30

21. The method according to any of claims 16 - 20 and wherein the first bit size is equal to the third bit size.

22. The method according to any of claims 16 - 21 and wherein the first bit size is equal to 64 bits.

5 23. The method according to any of claims 16 - 21 and wherein the second bit size is equal to 100 bits.

24. The method according to any of claims 16 - 21 and wherein the third bit size is equal to 64 bits.

10

25. The method according to any of claims 16 - 24 and wherein the combining unit is operative to perform a XOR operation.

15 26. The method according to any of claims 16 - 25 and wherein the expansion unit comprises a linear transformation.

27. The method according to claim 26 and wherein the linear transformation comprises an operation wherein each input bit influences at least two output bits.

20

28. The method according to claim 26 and wherein the linear transformation comprises an operation wherein each bit of the key influences one output bit.

25 29. The method according to claim 26 and wherein the linear transformation comprises an operation wherein any small set of input bits influences a larger set of output bits.

30 30. The method according to claim 26 and wherein the linear transformation comprises an operation wherein indices are selected so as to be spread equally between input bits and output bits.

31. The method according to any of claims 16 - 30 and wherein the expansion unit comprises two layers of gates operative to combine two inputs.
32. The method according to claim 31 and wherein the gates comprise
5 XOR operation gates.
33. The method according to claim 32 and further comprising a NOT operation gate after the XOR operation gates.
- 10 34. The method according to any of claims 16 - 33 and wherein the method of encrypting cannot be implemented except on specialized hardware.
35. The method according to any of claims 16 - 34 and wherein the MAC comprises a plurality of layers of mini-functions.
15
36. The method according to claim 35 and wherein the plurality of layers of the MAC comprises between 30 layers and 50 layers, inclusive.
37. The method according to either of claim 35 or claim 36 and wherein
20 a mini-function layer comprises two micro-functions:
one balanced micro-function; and
one non-linear micro-function.
38. The method according to claim 37 and wherein the mini-function
25 layer is operative to perform the following:
receiving an input;
splitting the input, at a splitter, into a block of balancing bits and a block of remaining input bits;
executing the method of the non-linear micro-function on the block
30 of remaining input bits;
inputting the result of the non-linear micro-function into the balanced micro-function;

executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits; and
outputting a result.

5 39. The method according to claim 38 and further comprising performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

40. The method according to claim 39 and wherein the invertible
10 transformation comprises an invertible transformation S-box.

41. The method according to claim 40 and wherein the invertible transformation S-box comprises a 2bit-to-2bit S-box.

15 42. The method according to any of claims 16 - 41 and further comprising:

providing a first function F_i and a second function F_j ;

providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of:

20 the first function F_i ; and

the second function F_j ;

providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of:

the first function F_i ; and

25 the second function F_j ;

utilizing the round key generation function in at least a first round to generate a second round key for use in a second round; and

utilizing the round mixing function in at least the first round to mix a first round key with a cipher state,

30 wherein one of the following is performed in the first round:

the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state; and

5 the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

10 43. A method of encrypting a block of data, the method comprising an emulation resistant combine key method comprised in a Feistel-like structure.

44. The method according to claim 43 and wherein the method is implemented in hardware.

15 45. The method according to either of claim 43 or claim 44 and further comprising mixing and condensing, the mixing and condensing comprising:

receiving an input of a block of data expressed as a block of bits;

and

20 mixing the bits of the block of data with a round key.

46. The method according to either of claim 43 or claim 44 and further comprising:

25 providing an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being greater than the first bit size;

providing a combining unit operative to combine an expanded block of data with a key;

30 providing a mix and condense unit (MAC) operative to mix the bits of a combined expanded block of data of the second bit size and condense the bit size of the input to a third bit size, the third bit size being less than the second bit size;

providing a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input to the mix and condense unit;

5 receiving an input comprising the block of data expressed as the block of bits;

inputting the block of bits into the expansion unit, thereby expanding the block of bits to a block of bits of the second bit size;

combining, at the combining unit, the block of bits of the second bit
10 size with a key;

substituting bits comprising the input to the plurality of layers of S-boxes with bits comprising the output of the plurality of layers of S-boxes;

outputting the output of the plurality of layers of S-boxes to the mix and condense unit;

15 mixing, at the mix and condense unit, the block of bits of the second bit size; and

condensing, at the mix and condense unit, the block of bits of the second bit size to a block of bits of the third size, thereby producing an encrypted block of data, the encrypted block of data being expressed as a block of bits of the
20 third bit size.

47. The method according to claim 46 and wherein the plurality of layers of S-boxes comprises two layers of S-boxes.

25 48. The method according to either claim 46 or claim 47 and wherein a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes comprising an input of the second layer of S-boxes.

30 49. The method according to any of claims 46 - 48 and wherein each S-box of the plurality of S-boxes comprises one S-box described in the Serpent cipher specification.

50. The method according to any of claims 47 - 49 and wherein the two layers of S-boxes comprise a first layer of S-boxes comprising 25 S-boxes and a second layer of S-boxes comprising 25 S-boxes.

5

51. The method according to any of claims 46 - 50 and wherein the first bit size is equal to the third bit size.

52. The method according to any of claims 46 - 51 and wherein the first
10 bit size is equal to 64 bits.

53. The method according to any of claims 46 - 51 and wherein the second bit size is equal to 100 bits.

15 54. The method according to any of claims 46 - 51 and wherein the third bit size is equal to 64 bits.

55. The method according to any of claims 46 - 54 and wherein the combining unit is operative to perform a XOR operation.

20

56. The method according to any of claims 46 - 55 and wherein the expansion unit comprises a linear transformation.

57. The method according to claim 56 and wherein the linear
25 transformation comprises an operation wherein each input bit influences at least two output bits.

58. The method according to claim 56 and wherein the linear transformation comprises an operation wherein each bit of the key influences one
30 output bit.

59. The method according to claim 56 and wherein the linear transformation comprises an operation wherein any small set of input bits influences a larger set of output bits.

5 60. The method according to claim 56 and wherein the linear transformation comprises an operation wherein indices are selected so as to be spread equally between input bits and output bits.

61. The method according to any of claims 46 - 60 and wherein the mix
10 and condense unit comprises a plurality of layers, each layer among the plurality of layers comprising a plurality of mini-functions.

62. The method according to claim 61 and wherein the plurality of layers of the MAC comprises between 30 layers and 50 layers, inclusive.

15

63. The method according to claim 61 or claim 62 and wherein a mini-function layer comprises two micro-functions:

one balanced micro-function; and
one non-linear micro-function.

20

64. The method according to claim 63 and wherein the mini-function layer is operative to perform the following:

receiving an input;

splitting the input, at a splitter, into a block of balancing bits and a

25 block of remaining input bits;

executing the method of the non-linear micro-function on the block of remaining input bits;

inputting the result of the non-linear micro-function into the balanced micro-function;

30 executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits; and

outputting a result.

65. The method according to claim 64 and further comprising performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

5

66. The method according to claim 65 and wherein the invertible transformation comprises an invertible transformation S-box.

67. The method according to claim 66 and wherein the invertible transformation S-box comprises a 2bit-to-2bit S-box.

10

68. The method according to any of claims 43 - 67 and further comprising:

providing a first function F_i and a second function F_j ;

15 providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of:

the first function F_i ; and

the second function F_j ;

20 providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of:

the first function F_i ; and

the second function F_j ;

utilizing the round key generation function in at least a first round to generate a second round key for use in a second round; and

25 utilizing the round mixing function in at least the first round to mix a first round key with a cipher state,

wherein one of the following is performed in the first round:

the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially

30 simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state; and

the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

5

69. A method of encrypting a block of data, the method comprising:
providing a combining unit operative to combine the block of data with a key;

10

providing a mixing unit operative to mix the bits of a combined key and block of data;

providing a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mixing unit and to provide an output comprising an input to the mixing unit;

15

receiving an input comprising the block of data expressed as a block of bits;

combining, at a combining unit, the block of bits with a key;

substituting bits comprising the input to the plurality of layers of S-boxes with bits comprising the output of the plurality of layers of S-boxes;

20

outputting the output of the plurality of layers of S-boxes as a second block of bits to the mixing unit; and

mixing, at the mixing unit, the second block of bits, thereby producing an encrypted block of data,

wherein the mixing unit comprises a plurality of layers, each layer comprising a plurality of mini-functions.

25

70. The method according to claim 69 and wherein the plurality of layers of S-boxes comprises two layers of S-boxes.

30

71. The method according to either claim 69 or claim 70 and wherein a diffusion layer, disposed between each layer of the plurality of S-boxes, routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes comprising an input of the second layer of S-boxes.

72. The method according to any of claims 69 - 71 and wherein each S-box of the plurality of S-boxes comprises one S-box described in the Serpent cipher specification.

5

73. The method according to any of claims 70 - 72 and wherein the two layers of S-boxes comprise a first layer of S-boxes comprising 25 S-boxes and a second layer of S-boxes comprising 25 S-boxes.

10 74. The method according to any of claims 69 - 73 and wherein the plurality of layers of the mixing unit comprises between 30 and 50 layers, inclusive.

15 75. The method according to any of claims 69 - 74 and wherein the combining unit is operative to perform a XOR operation.

76. The method according to any of claims 69 - 75 and wherein a mini-function layer comprises two micro-functions:

20 one balanced micro-function; and
 one non-linear micro-function.

77. The method according to claim 76 and wherein the mini-function layer is operative to perform the following:

25 receiving an input;
 splitting the input, at a splitter, into a block of balancing bits and a block of remaining input bits;
 executing the method of the non-linear micro-function on the block of remaining input bits;
 inputting the result of the non-linear micro-function into the
30 balanced micro-function;
 executing the method of the balanced micro-function on the result of the non-linear micro-function and the balancing bits; and
 outputting a result.

78. The method according to claim 77 and further comprising performing an invertible transformation on the block of balancing bits prior to the executing the method of the balanced micro-function.

5

79. The method according to claim 78 and wherein the invertible transformation comprises an invertible transformation S-box.

80. The method according to claim 79 and wherein the invertible transformation S-box comprises a 2bit-to-2bit S-box.

10

81. The method according to any of claims 69 - 80 and further comprising:

providing an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being greater than the first bit size; and

15

prior to the combining, inputting the block of bits into the expansion unit, and therein expanding the block of bits to a block of bits of the second bit size.

20

82. The method according to claim 81 and further comprising:

after the mixing, condensing, at the mix and condense unit, the block of bits of the second bit size to a block of bits of a third size, thereby producing an encrypted block of data, the encrypted block of data being expressed as a block of bits of the third bit size.

25

83. The method according to any of claims 69 - 82 and further comprising:

providing a first function F_i and a second function F_j ;

30

providing a round key generation function, the round key generation function being operative to utilize, in any given round, exactly one of:

the first function F_i ; and

the second function F_j ;

providing a round mixing function, the round mixing function being operative to utilize, in any given round, exactly one of:

the first function F_i ; and

5 the second function F_j ;

utilizing the round key generation function in at least a first round to generate a second round key for use in a second round; and

utilizing the round mixing function in at least the first round to mix a first round key with a cipher state,

10 wherein one of the following is performed in the first round:

the round key generation function utilizes the first function F_i to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the second function F_j to mix the first round key with the cipher state; and

15 the round key generation function utilizes the second function F_j to generate the second round key for use in the second round, substantially simultaneously with the round key mixing function utilizing the first function F_i to mix the first round key with the cipher state.

20 84. A method comprising:

combining a control input derived from a right part of a Feistel-like structure with a transformation input comprising a left part of the Feistel-like structure; and

25 producing an output comprising a combination of bits comprised in the control input and bits comprised in the transformation input,

wherein no bit of the combination of bits comprises a linear combination of bits from the control input and bits from the transformation input.

85. The method according to claim 84 and wherein, with respect to a
30 fixed control input, the method comprises an invertible method.

86. The method according to claim 85 and wherein the inverse of the method is not identical to the method.

87. The method according to any of claims 84 - 86 and wherein the method comprises a non-linear layer comprising at least one S-box.

88. The method according to of claims 84 - 86 and also comprising a linear transformation of the control input and the transformation input.

89. The method according to claim 88 and also comprising:
splitting, at a control input splitter, the control input, into a plurality of control input sub-blocks;
splitting, at a transformation input splitter, the transformation input, into a plurality of transformation input sub-blocks;
linearly combining each one of the plurality of control input sub-blocks with a corresponding one of the plurality of transformation input sub-blocks; and
joining the result of the linear combining at a output joiner.

90. The method according to claim 89 wherein each one of the plurality of control input sub-blocks and a corresponding one of the plurality of transformation input sub-blocks comprise sub-blocks of the same size.

91. The method according to claim 90 wherein a first sub-block of the plurality of control input sub-blocks comprises a sub-block of a different size than a second sub-block of the plurality of control input sub-blocks.

92. The method according to any of claims 89 - 91 and wherein the transformation input splitter permutes the transformation input prior to the splitting at the transformation input splitter.

93. The method according to any of claims 89 - 92 wherein the output joiner permutes an output after the joining operation.

94. The method according to any of claims 89 - 93 and wherein the linearly combining comprises $(A(C) \times I) \oplus C$, where C represents the control input sub-block, I represents the transformation input sub-block, and $A(C)$ comprises a matrix depending on C , of size $m \times m$, where m is a size of the control input sub-block.

95. The method according to claim 94 and wherein $A(C) =$

$$\begin{bmatrix} 1 & C[0] & 0 & C[3] \\ 0 & 1 & C[1] & 0 \\ 0 & 0 & 1 & C[2] \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ C[3] & 1 & 0 & 0 \\ 0 & C[1] & 1 & 0 \\ C[0] & 0 & C[2] & 1 \end{bmatrix}$$

where $C[0...3]$ comprise bits comprised in the control input.

96. The method according to any of claims 88 - 95 and also comprising a non-linear layer comprising at least one S-box.

97. The method according to claim 96 and wherein an output from the linear transformation comprises an input for the non-linear layer.

98. The method according to claim 96 and wherein an output from the non-linear layer comprises a transformation input for the linear transformation.

99. The method according to any of claims 87 - 98 and wherein at least one of the S-boxes comprises an S-box according to the Serpent Cipher specification.

100. The method according to any of claims 87 - 99 and wherein the S-box layer comprises S-boxes which are simple to implement in hardware.

101. The method according to any of claims 84 - 100 and wherein the method is cryptographically secure and non-involutable.

102. An encryptor for encrypting a block of data, the encryptor
5 comprising:

a combining unit operative to combine a key with a block of data, the block of data being expressed as a block of bits;

a mix and condense unit operative to mix bits comprised in the block of bits among themselves; and

10 a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input to the mix and condense unit,

wherein a received input comprising the block of data expressed as the block of bits is combined, at the combining unit, with a key, and

15 bits comprised in the combined block of bits are, in each layer of the plurality of layers of S-boxes, substituted for other bits, thereby providing output bits,

bits in the output bits are mixed among themselves at the mix and condense unit, and

20 the mix and condense unit comprises a plurality of layers, each layer among the plurality of layers comprising a plurality of mini-functions.

103. The encryptor according to claim 102 and wherein the encrypting cannot be efficiently implemented except on specialized hardware.

25

104. An encryptor for encrypting a block of data, the encryptor comprising:

an expansion unit operative to expand the block of data, expressed as a block of bits, from a first bit size to a second bit size, the second bit size being

30 greater than the first bit size, thereby producing an expanded block of data;

a combining unit operative to receive the expanded block of data from the expansion unit and combine the expanded block of data with a key thereby producing a combined expanded block of data of the second bit size;

5 a mix and condense unit operative to mix the bits of the combined expanded block of data of the second bit size and condense the bit size of the combined expanded block of data of the second bit size to a third bit size, the third bit size being less than the second bit size; and

10 a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input to the mix and condense unit,

wherein the mix and condense unit comprises a plurality of layers, each layer among the plurality of layers comprising a plurality of mini-functions.

105. The encryptor according to claim 104 and wherein the encryptor cannot be implemented except on specialized hardware.

106. An encryptor operative to encrypt a block of data, the encryptor comprising:

20 a combining unit operative to combine the block of data with a key and produce a combined key and block of data;

a mixing unit operative to mix the bits of the combined key and block of data; and

25 a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input to the mix and condense unit,

wherein the mixing unit comprises a plurality of layers, each layer comprising a plurality of mini-functions.

107. An apparatus comprising:

30 a combiner operative to combine a control input derived from a right part of a Feistel-like structure with a transformation input comprising a left part of the Feistel-like structure;

an outputter operative to producing an output comprising a combination of bits comprised in the control input and bits comprised in the transformation input; and

5 a plurality of layers of S-boxes, the S-boxes operative to receive an input comprising an input which has not yet been input into the mix and condense unit and to provide an output comprising an input to the mix and condense unit,

wherein no bit of the combination of bits comprises a linear combination of bits from the control input and bits from the transformation input.

10 108. A cipher device comprising:

a combine key right (CKR) unit, operative to receive a first portion of a message and to combine the first portion of the message with a key, the CKR comprising:

15 an expansion unit operative to expand the first portion of the message to a number of bits appropriate to a key size, thereby producing an expanded first portion of the message;

a combining function unit operative to receive the expanded first portion of the message and combine the expanded first portion of the message with the key, thereby producing a combined output;

20 a mixing and condensing function unit operative to receive the combined output and condensing the combined output; and

a plurality of layers of S-boxes disposed at least one of before the mixing and condensing function unit, each S-box among the plurality of S-boxes comprised in each of the plurality of layers of the S-boxes being operative
25 to receive an input string of bits and substitute the input string of bits with an output string of bits; and

a combine right - left (CRL) unit, operative to combine an output of the CKR with a second portion of the message.

30 109. The device according to claim 108 and wherein the first portion of the message comprises a right portion of the message, and the second portion of the message comprises a left portion of the message.

110. The device according to claim 108 and wherein the first portion of the message comprises a left portion of the message, and the second portion of the message comprises a right portion of the message.

5

111. The device according to any of claims 108 - 110 and wherein the plurality of layers of S-boxes comprises two layers of S-boxes.

112. The device according to either any of claims 108 - 111 and wherein
10 a diffusion layer, disposed between each layer of the plurality of S-boxes routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes comprising an input of the second layer of S-boxes.

113 The method according to any of claims 108 - 112 and wherein the
15 S-boxes comprise S-boxes described in the Serpent Cipher specification.

114. In a cipher device comprising:
a combine key right (CKR) unit, operative to receive a first portion
of a message and to combine the first portion of the message with a key, the CKR
20 comprising:

an expansion unit operative to expand the first portion of the message to a number of bits appropriate to a key size, thereby producing an expanded first portion of the message;

a combining function unit operative to receive the expanded
25 first portion of the message and combine the expanded first portion of the message with the key, thereby producing a combined output;

a mixing and condensing function unit operative to receive the combined output and condensing the combined output; and

a combine right - left (CRL) unit, operative to combine an output of
30 the CKR with a second portion of the message,

an improvement comprising adding a plurality of layers of S-boxes disposed before the mixing and condensing function unit, thereby increasing a

level of confusion of an output ciphertext and making the ciphertext more resistant to a differential cryptographic attack.

115. The device according to claim 114 and wherein the first portion of
5 the message comprises a right portion of the message, and the second portion of the message comprises a left portion of the message.

116. The device according to claim 114 and wherein the first portion of
the message comprises a left portion of the message, and the second portion of the
10 message comprises a right portion of the message.

117. The device according to any of claims 114 - 116 and wherein the plurality of layers of S-boxes comprises two layers of S-boxes.

118. The device according to either any of claims 114 - 117 and wherein
15 a diffusion layer, disposed between each layer of the plurality of S-boxes routes the output of a first layer of S-boxes to a second layer of S-boxes, the output of the first layer of S-boxes comprising an input of the second layer of S-boxes.

119. The method according to any of claims 114 - 118 and wherein the
20 S-boxes comprise S-boxes described in the Serpent Cipher specification.

FIG. 1

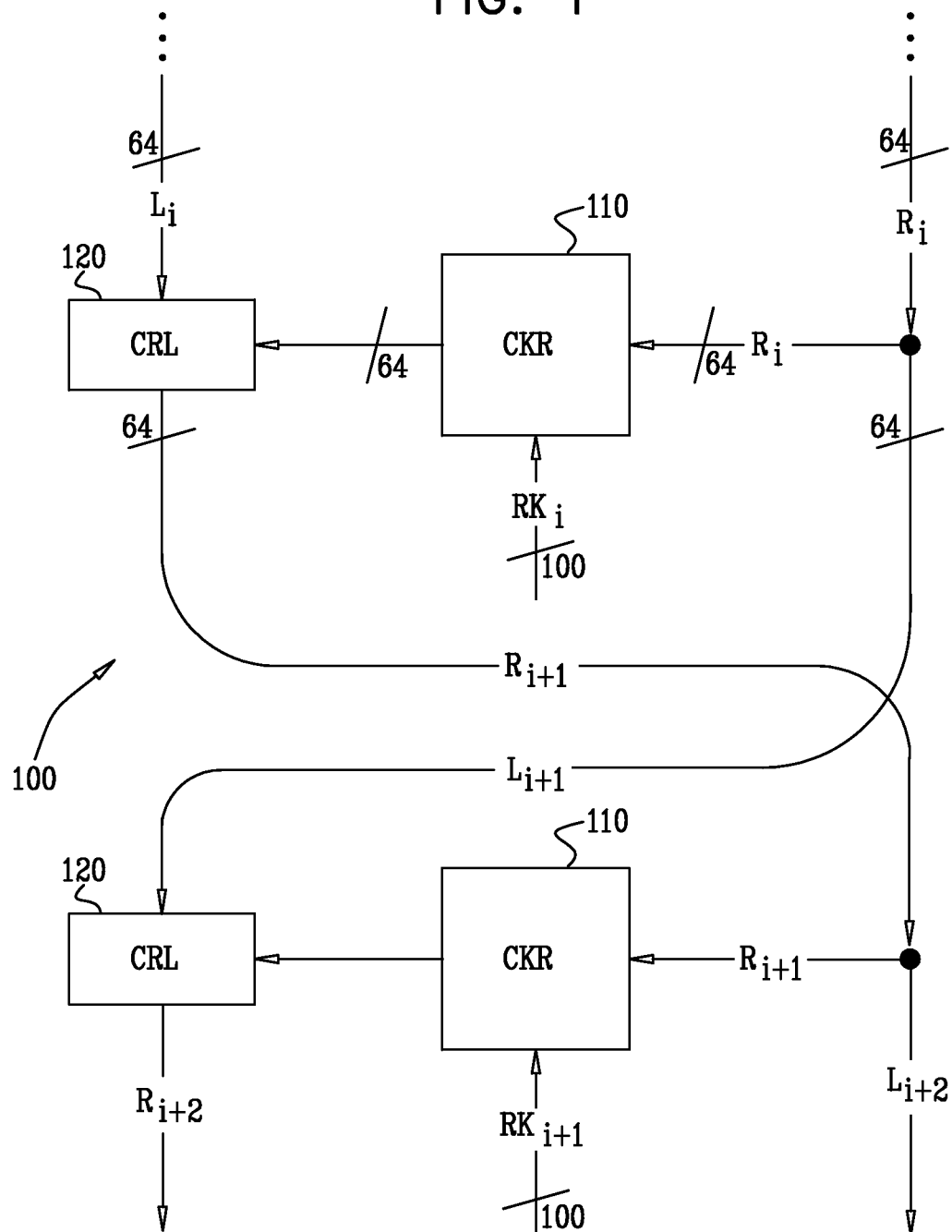


FIG. 2A

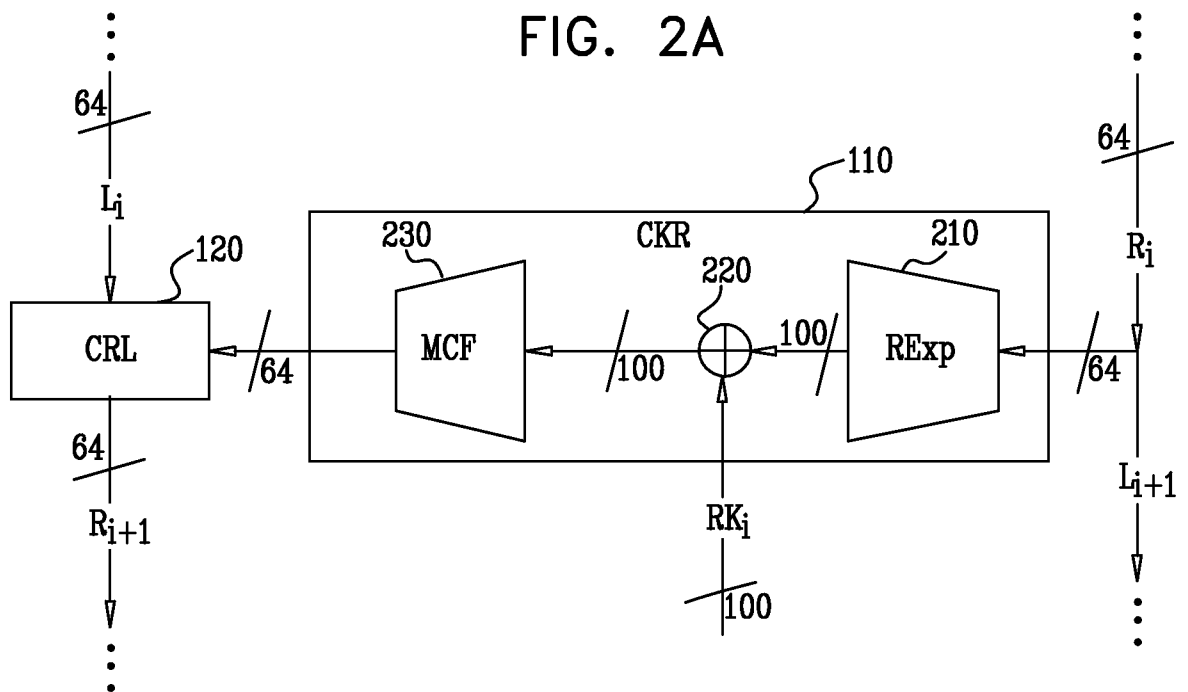


FIG. 2B

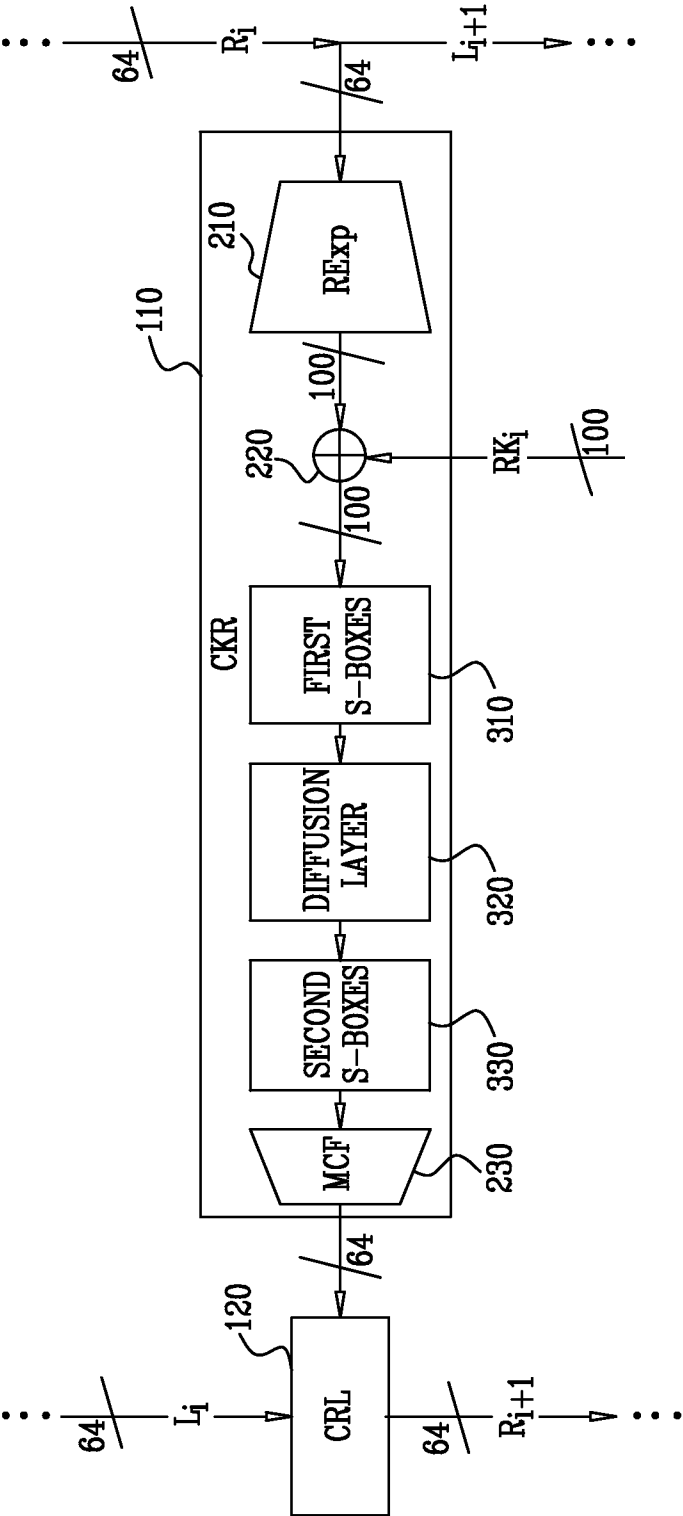


FIG. 3

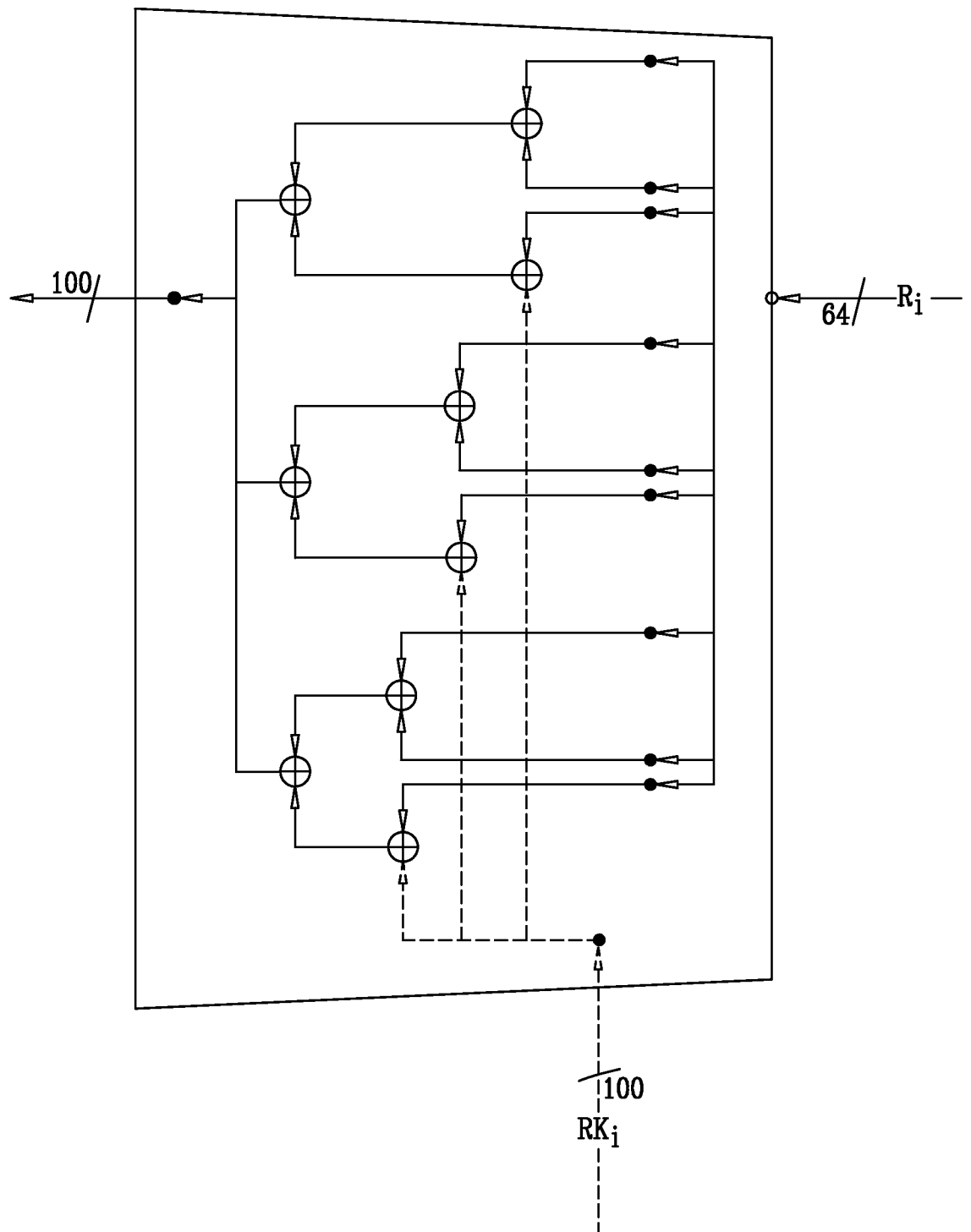


FIG. 4

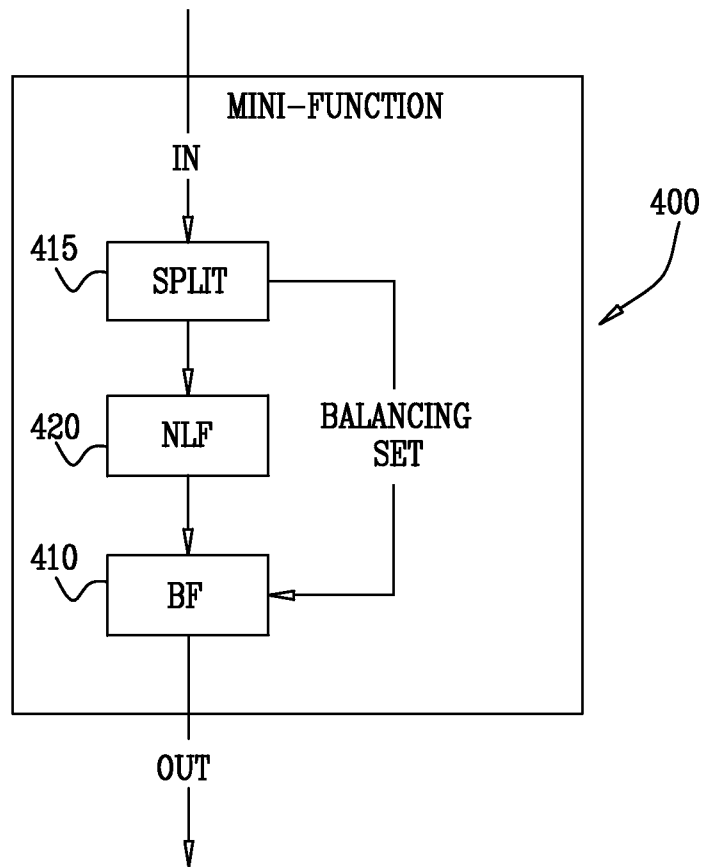
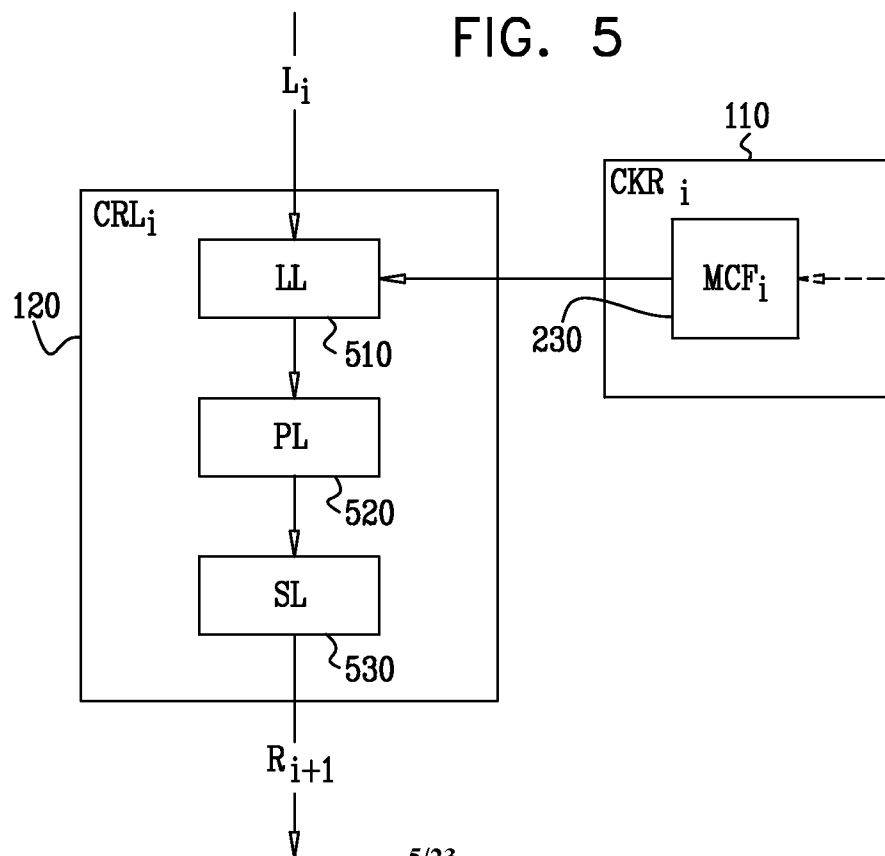
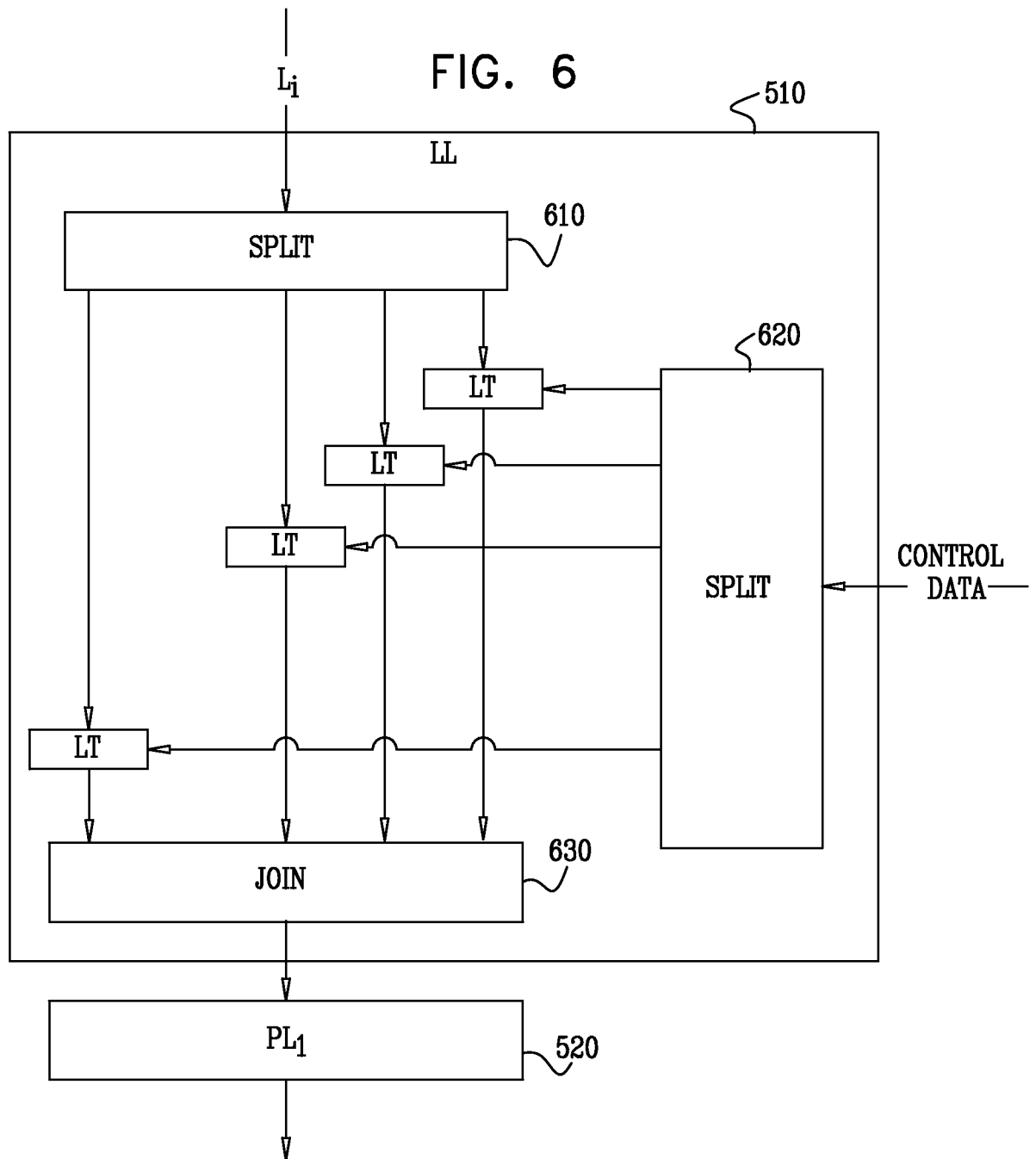


FIG. 5





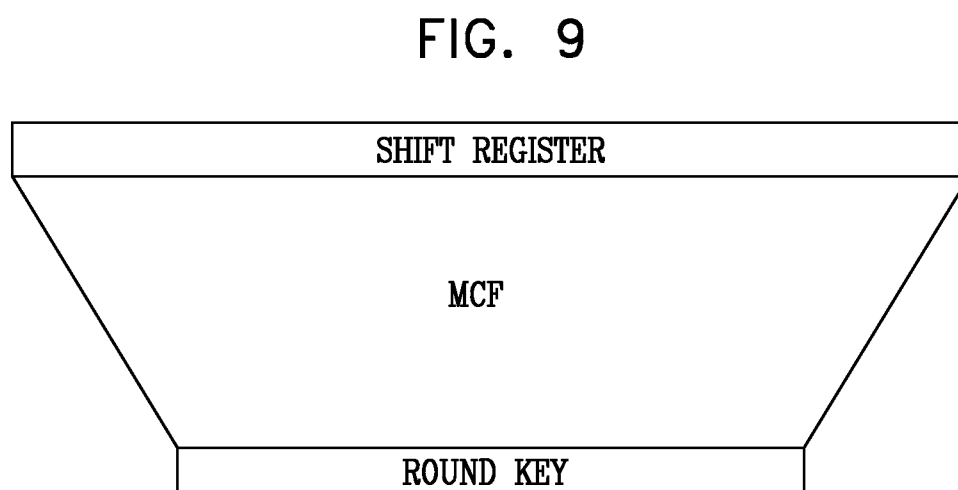
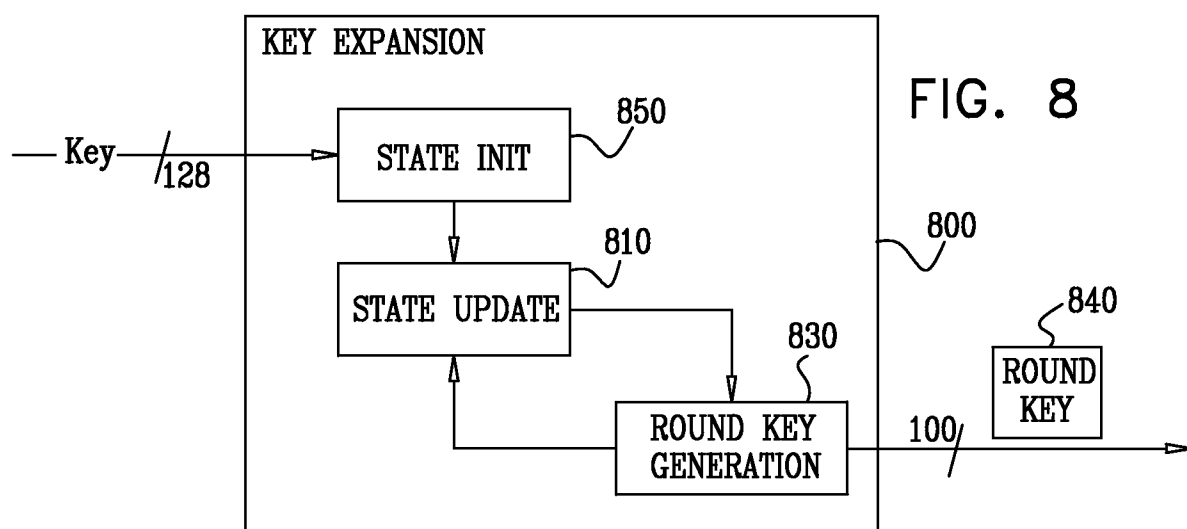
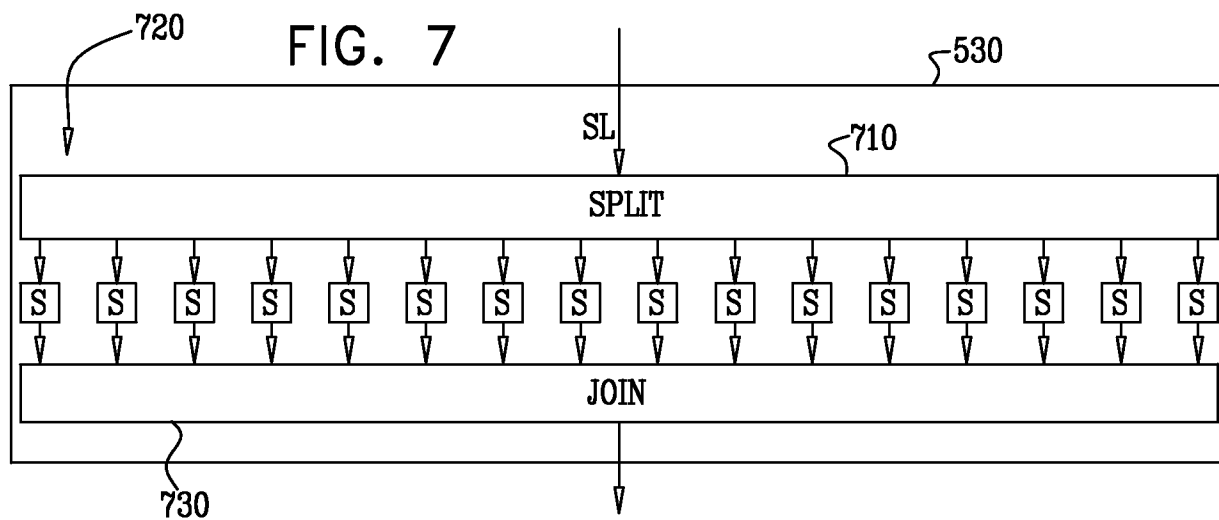


FIG. 10

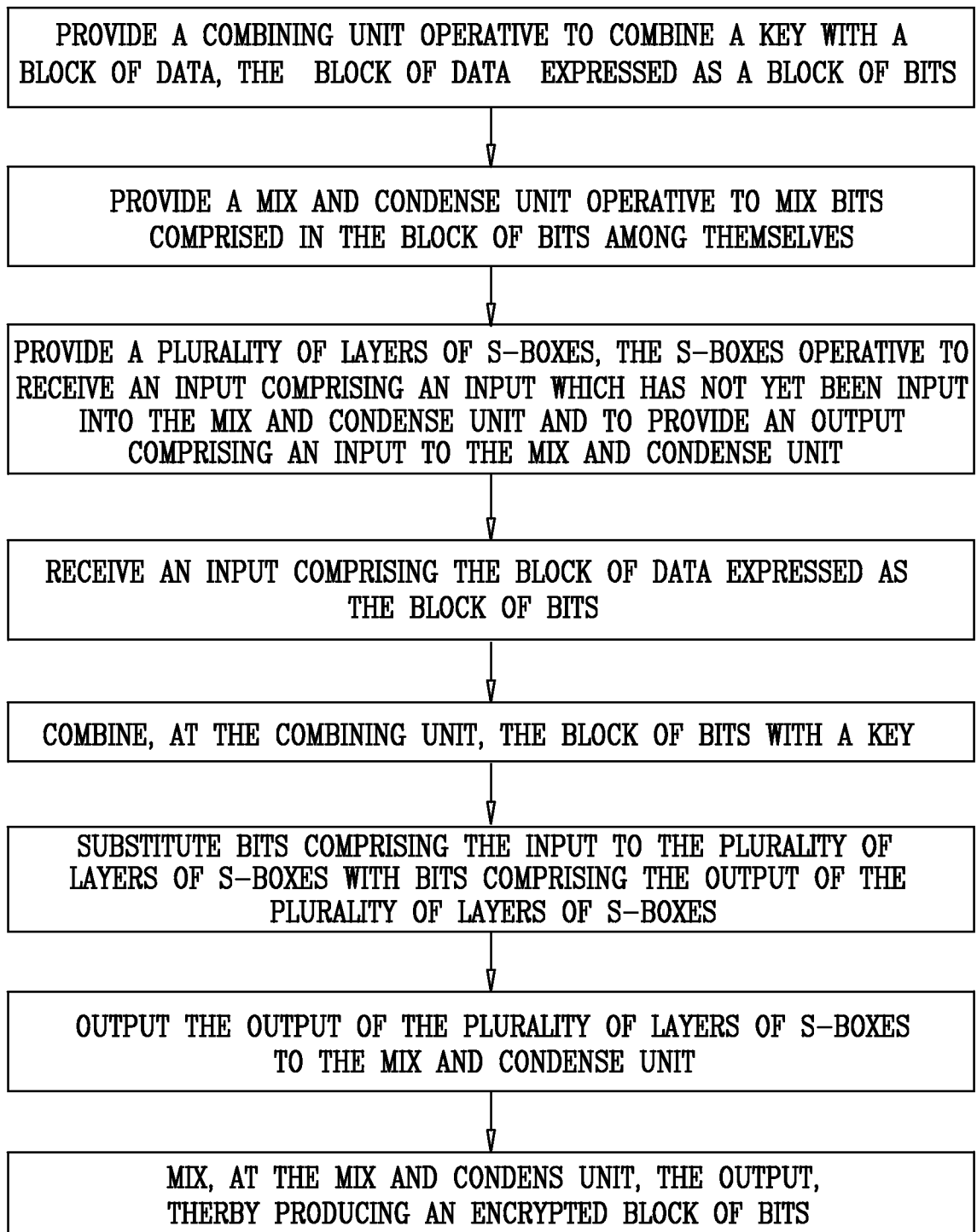


FIG. 11A

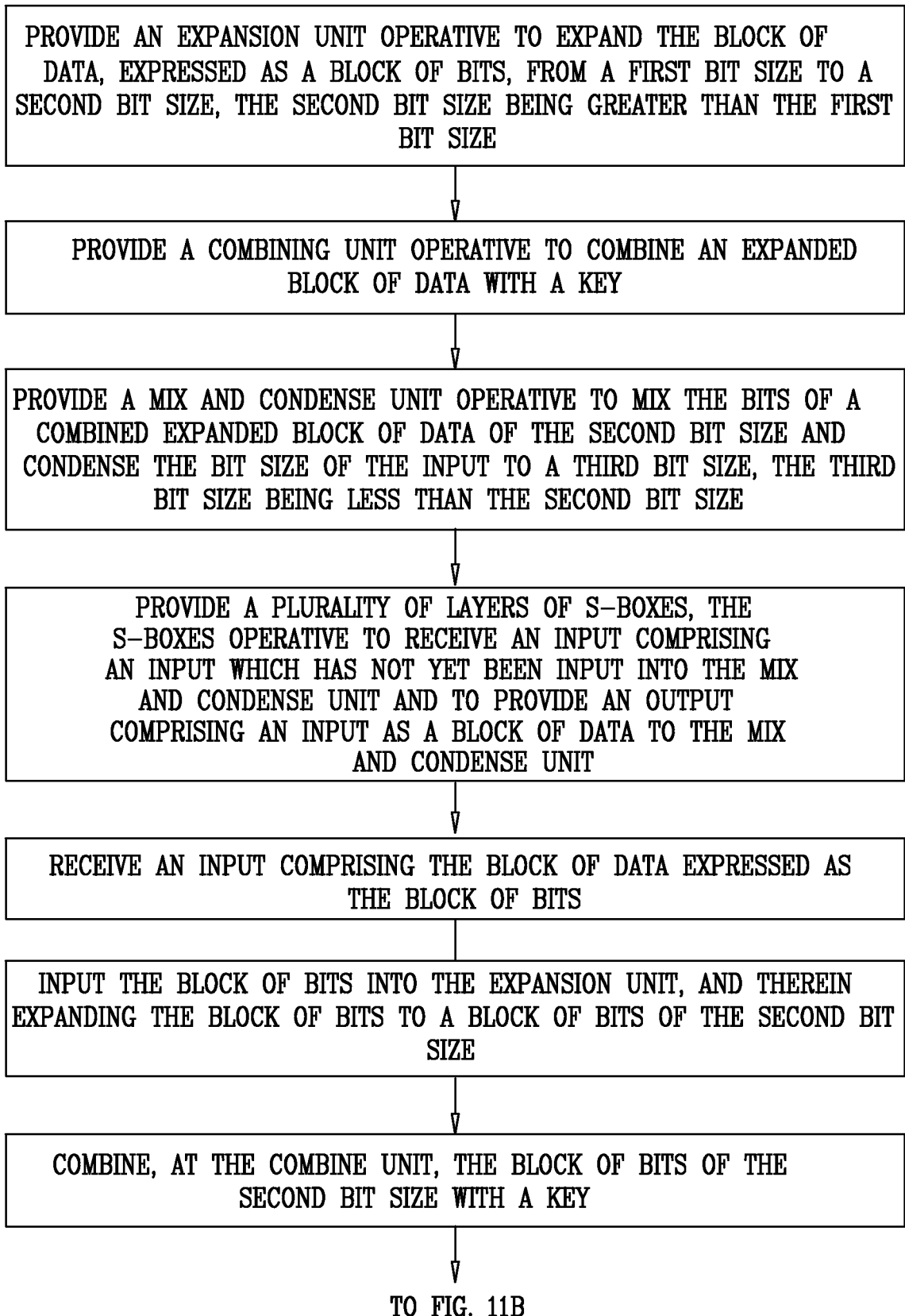


FIG. 11B

FROM FIG. 11A

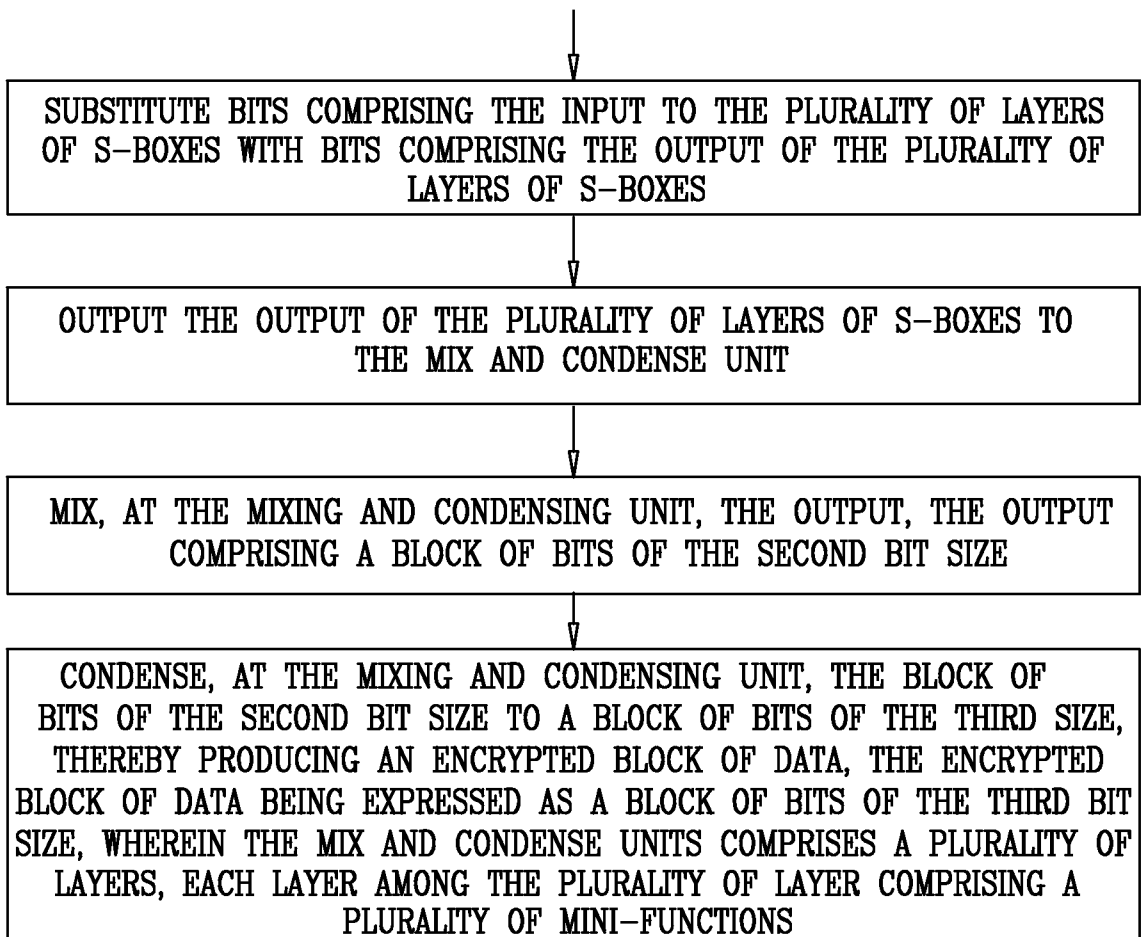


FIG. 12

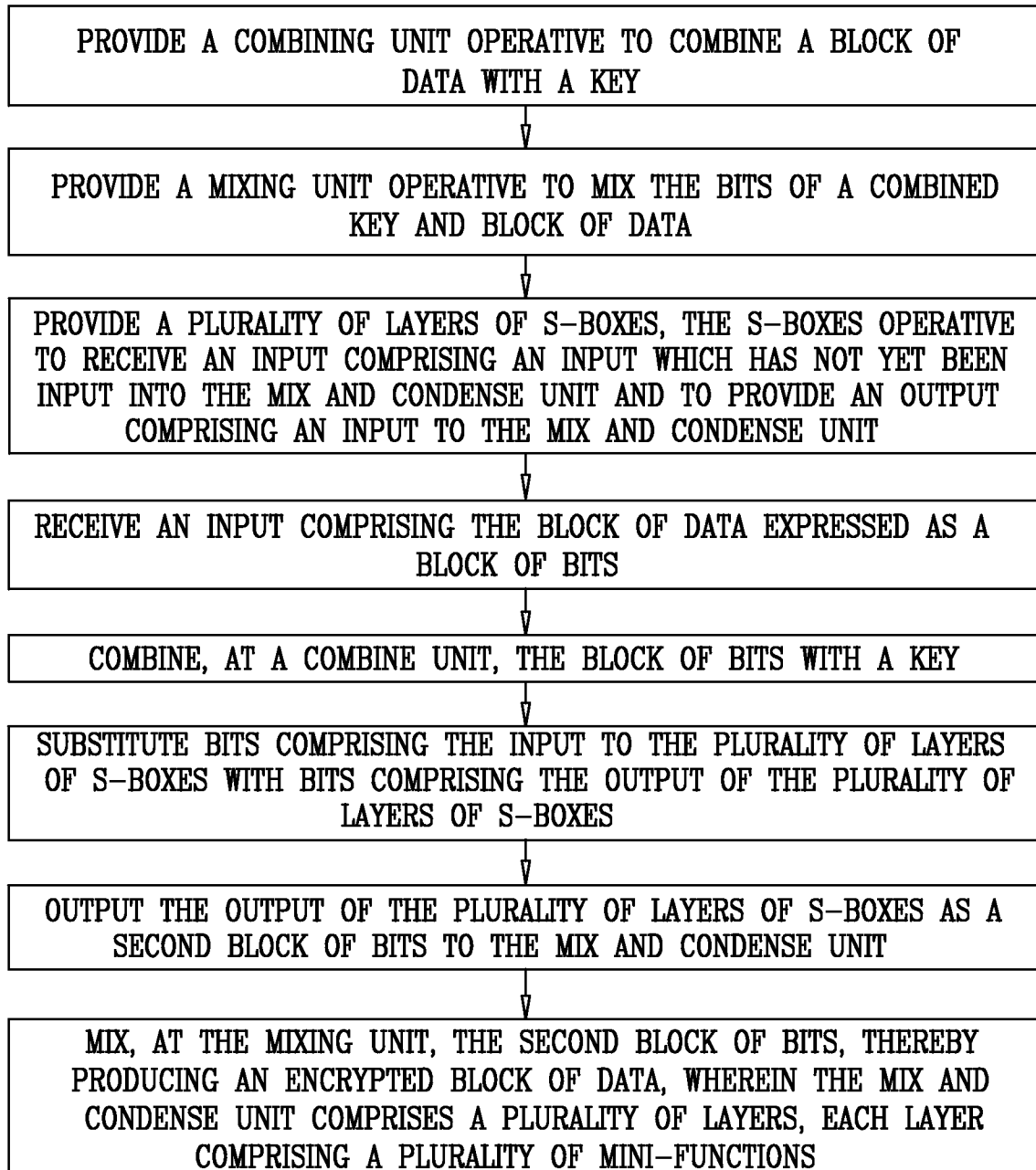


FIG. 13

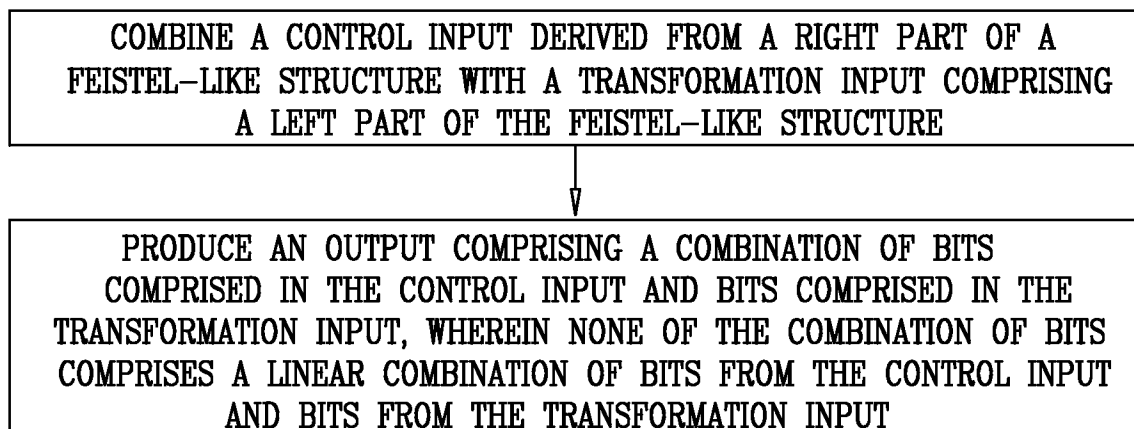


FIG. 14

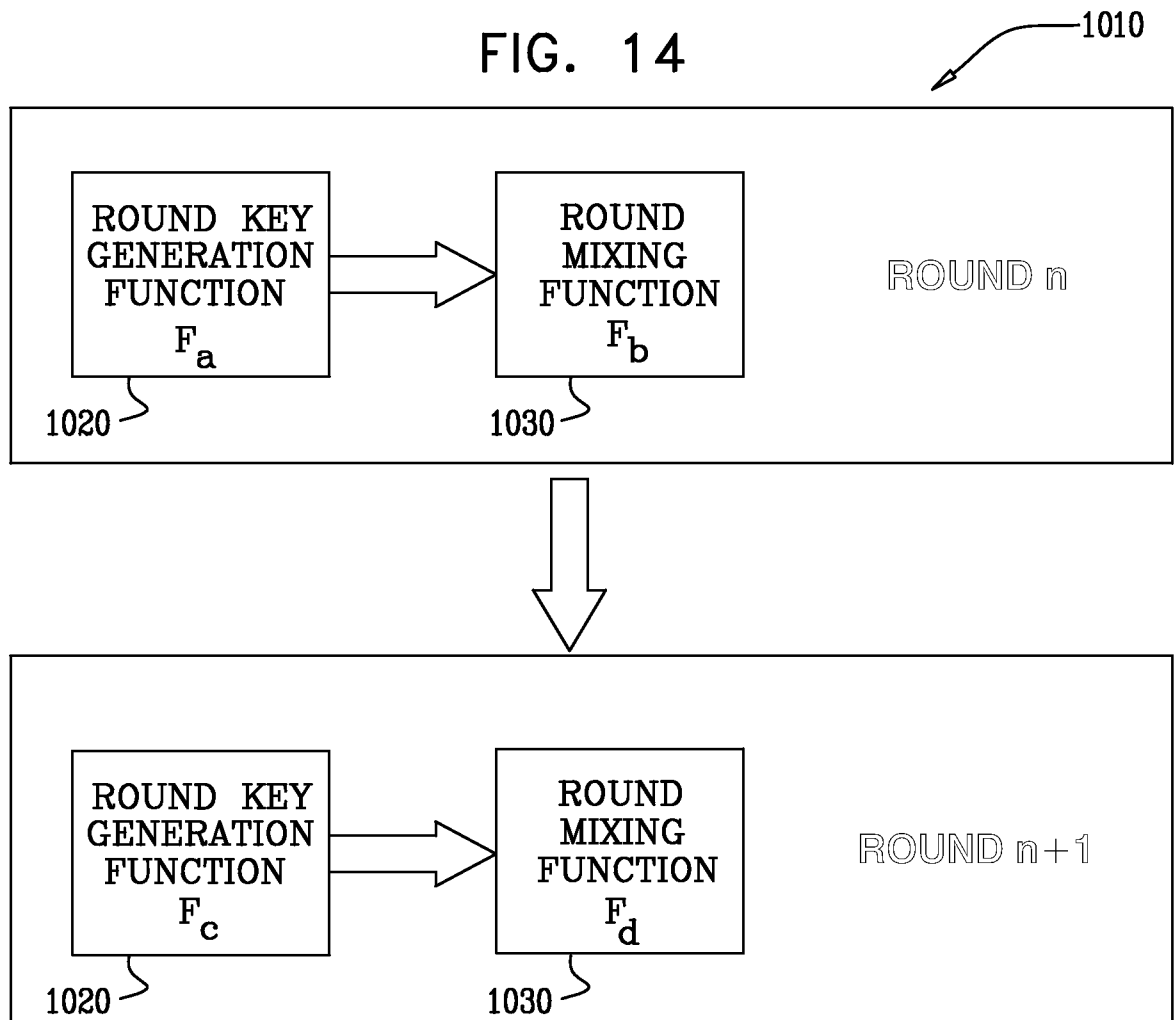


FIG. 15

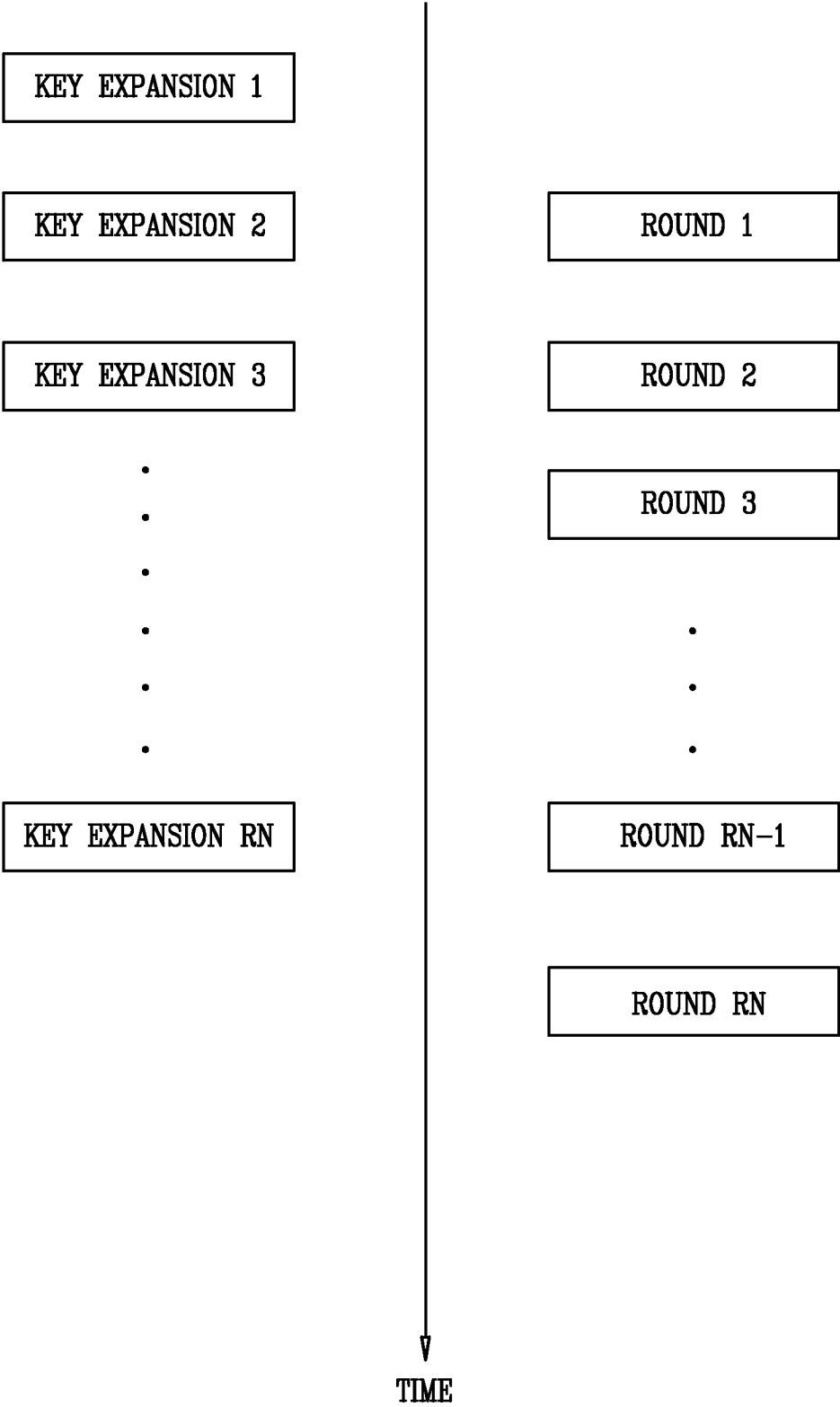


FIG. 16

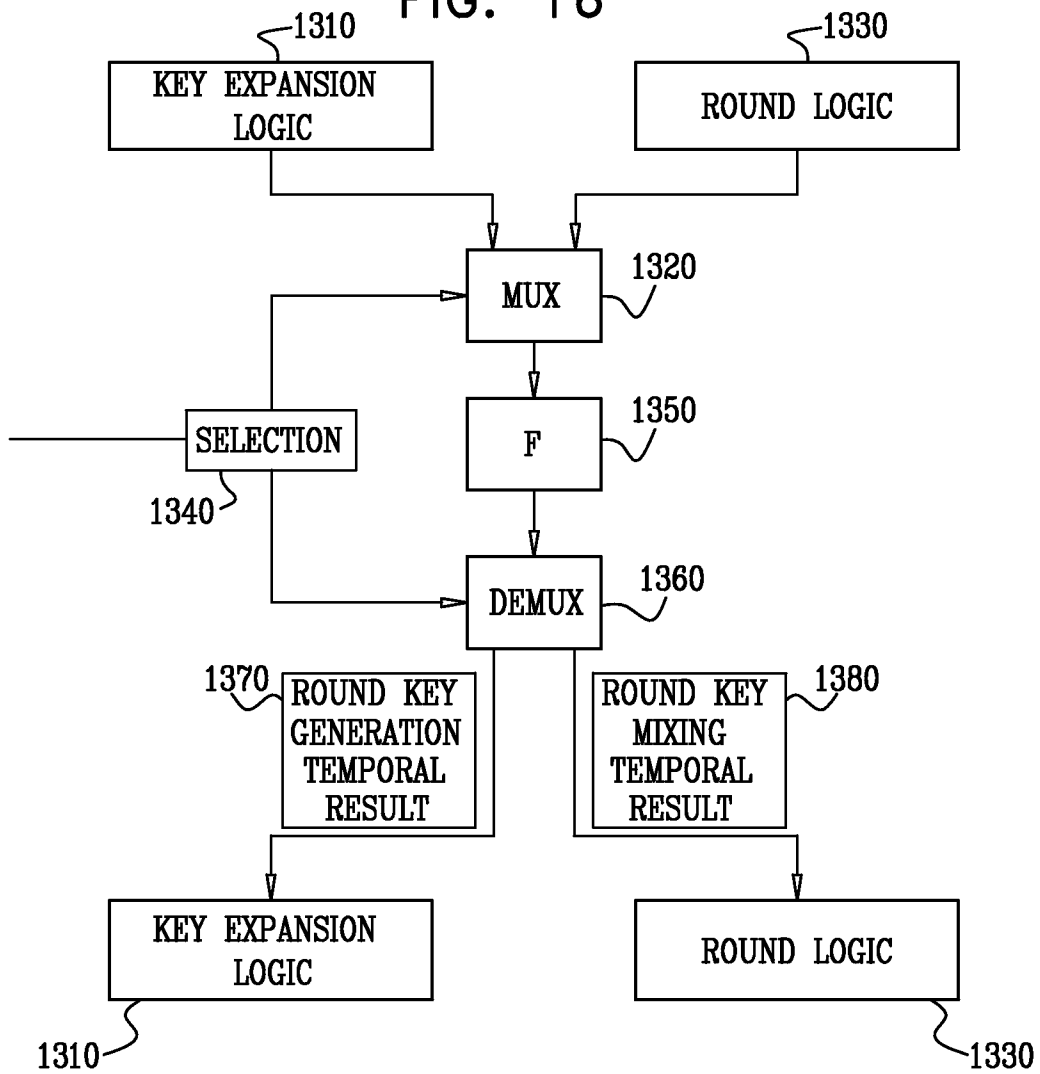
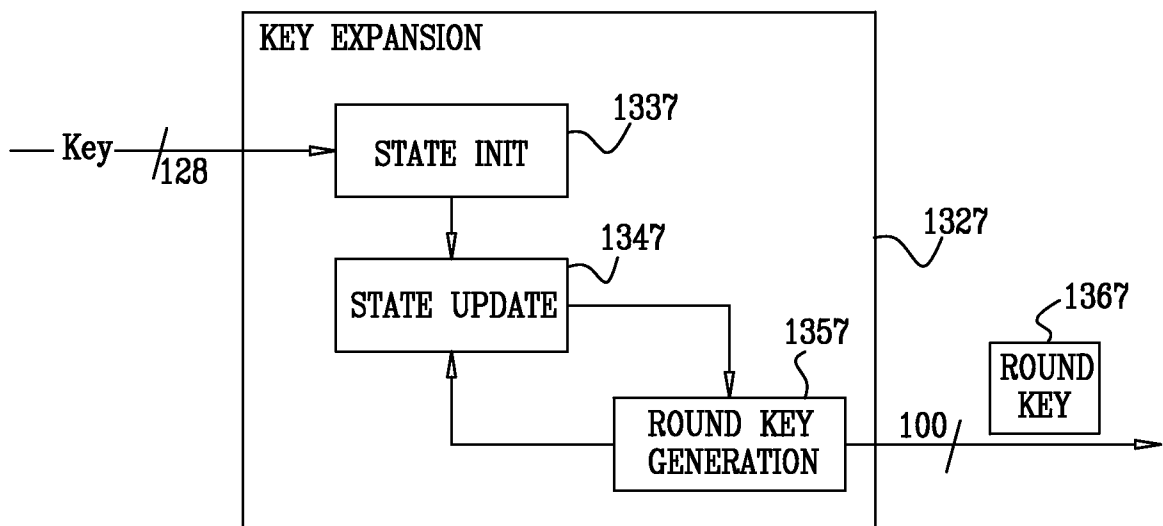


FIG. 17



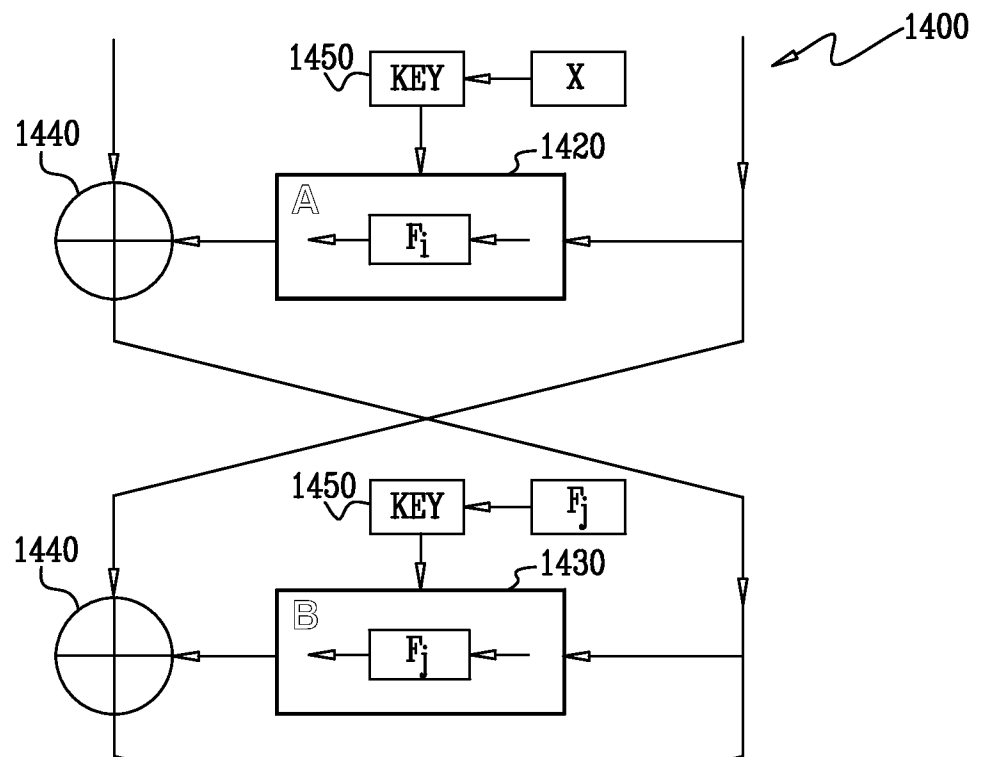


FIG. 18

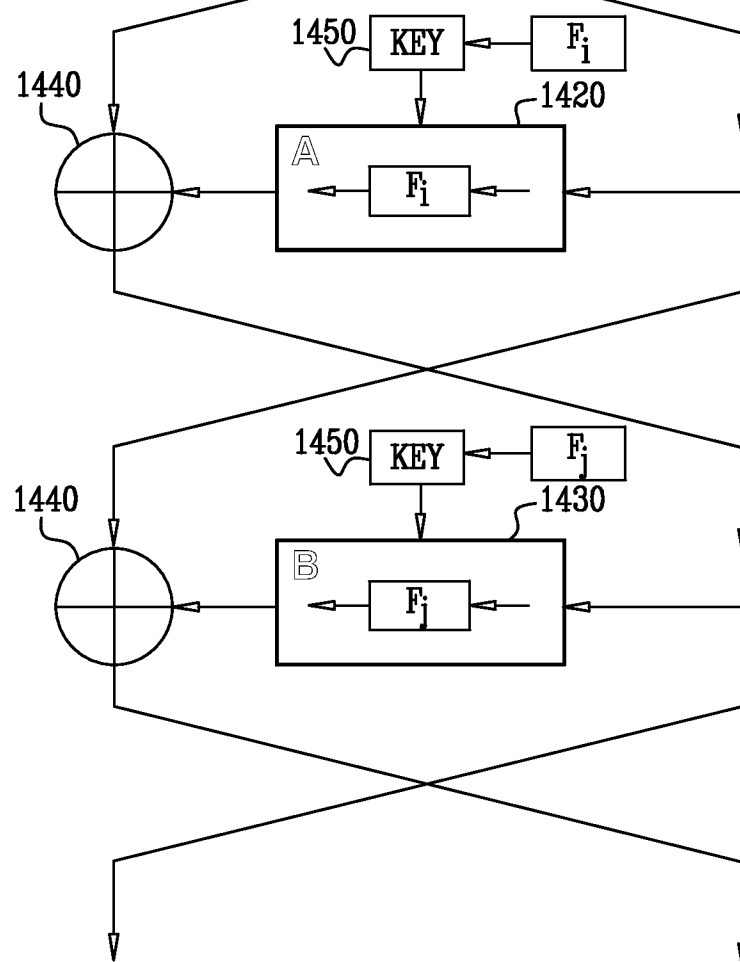


FIG. 19

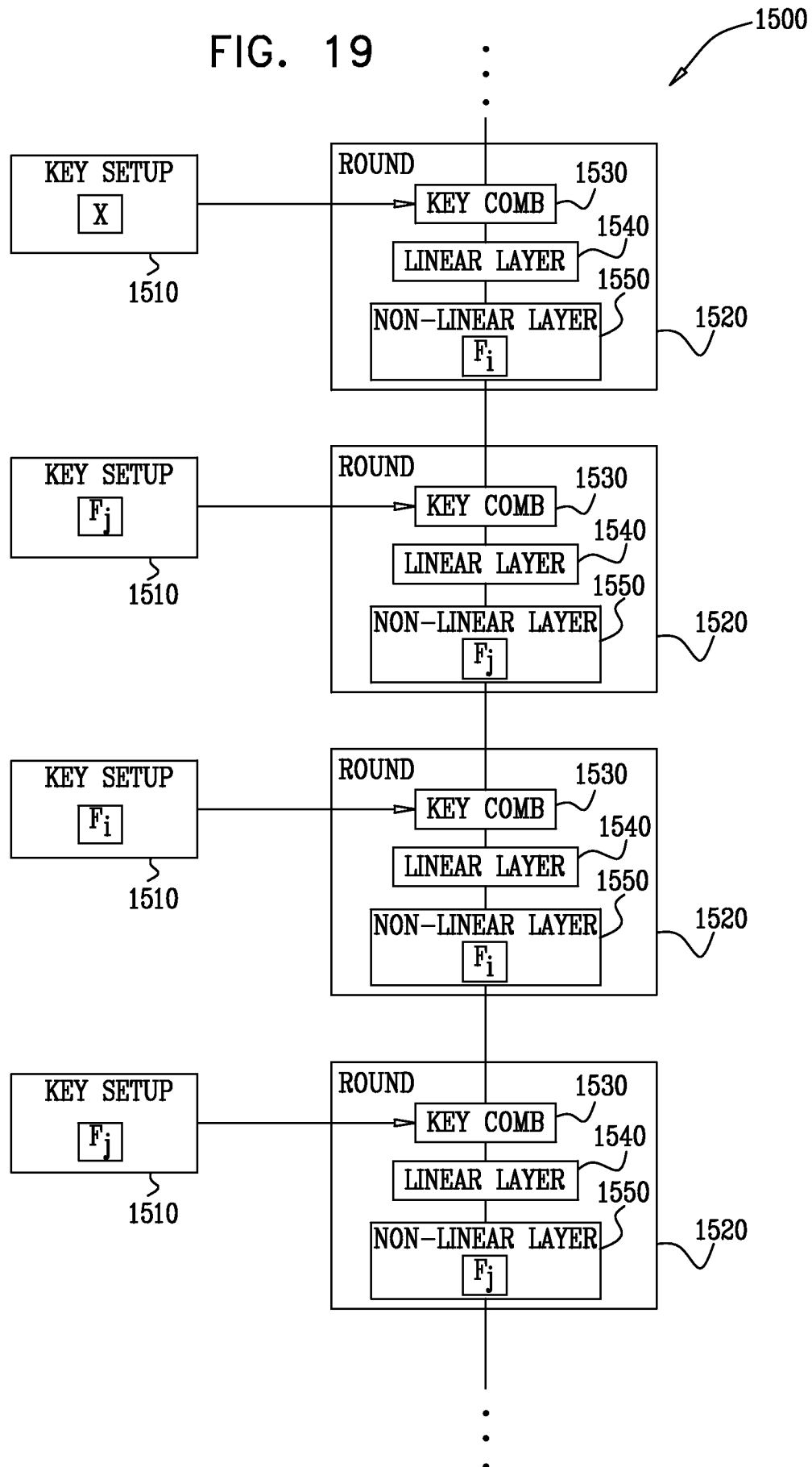


FIG. 20

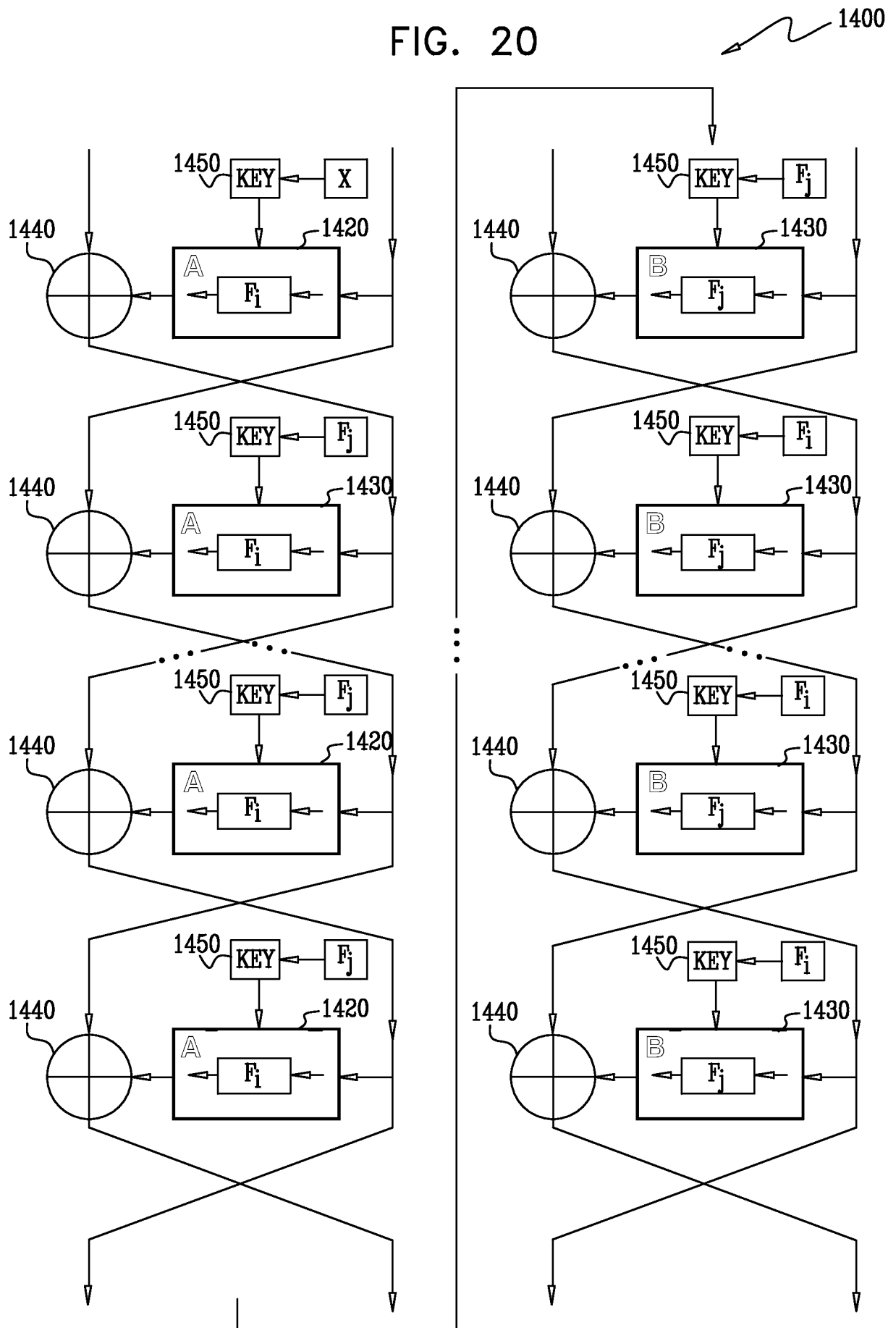


FIG. 21

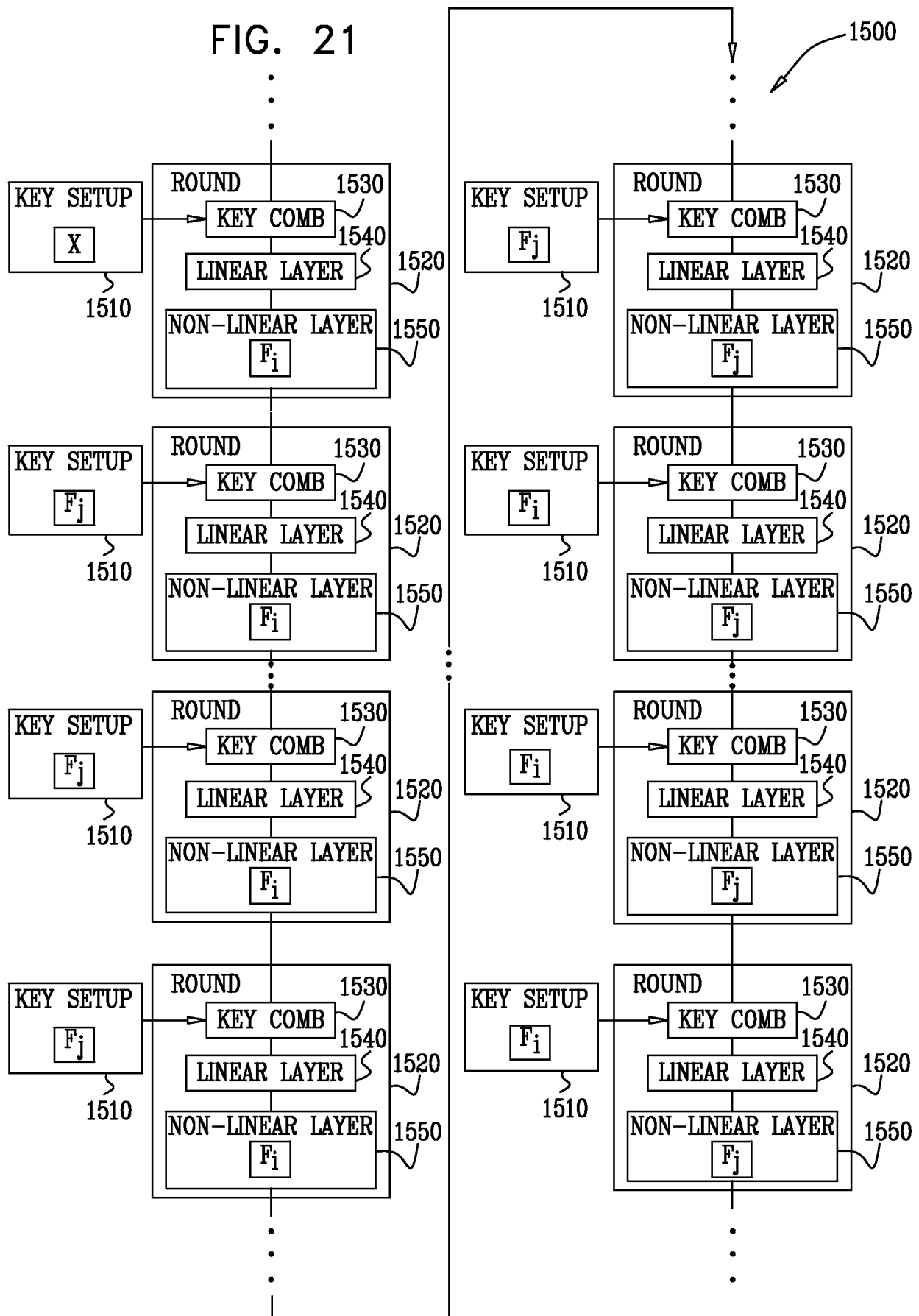


FIG. 22

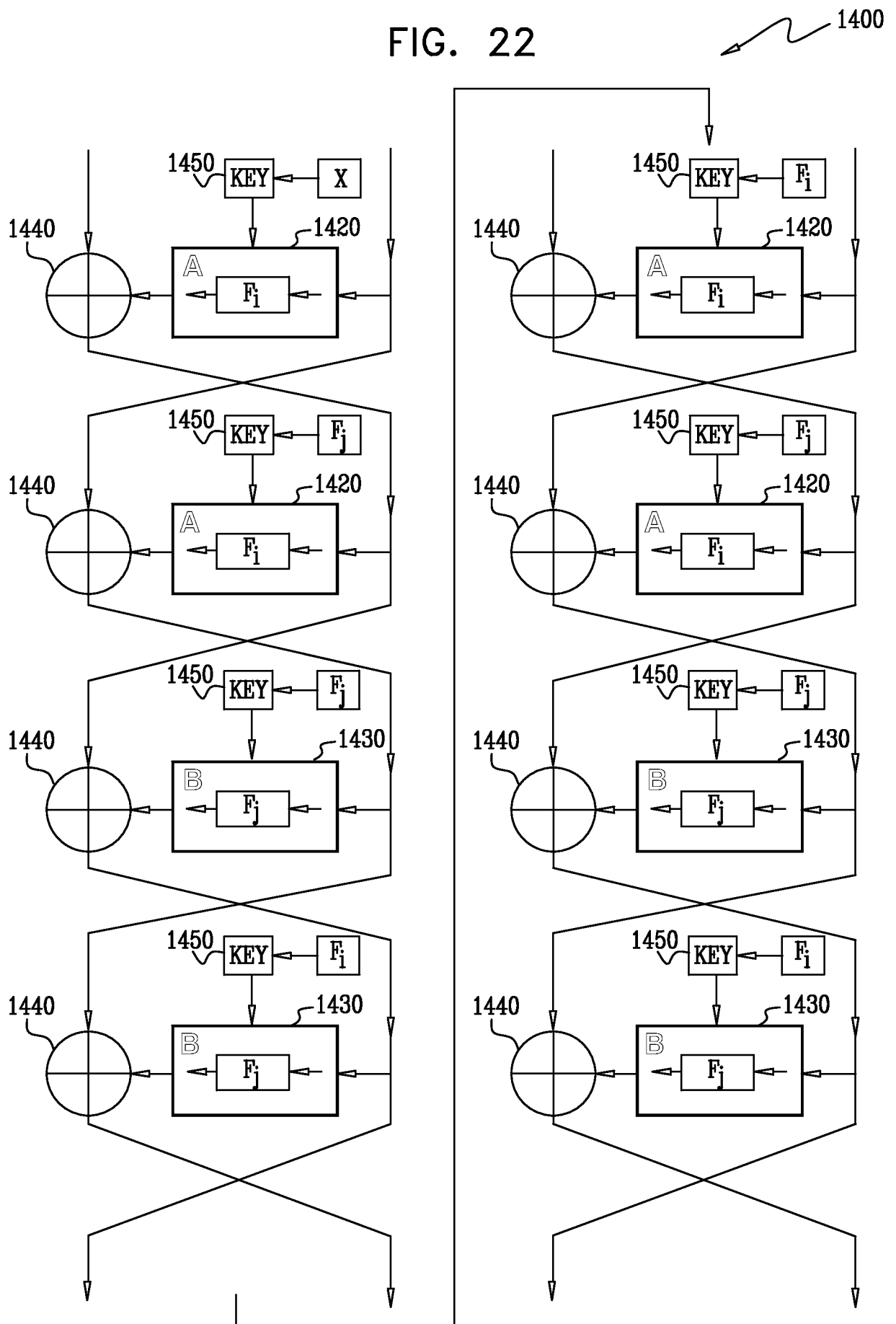


FIG. 23

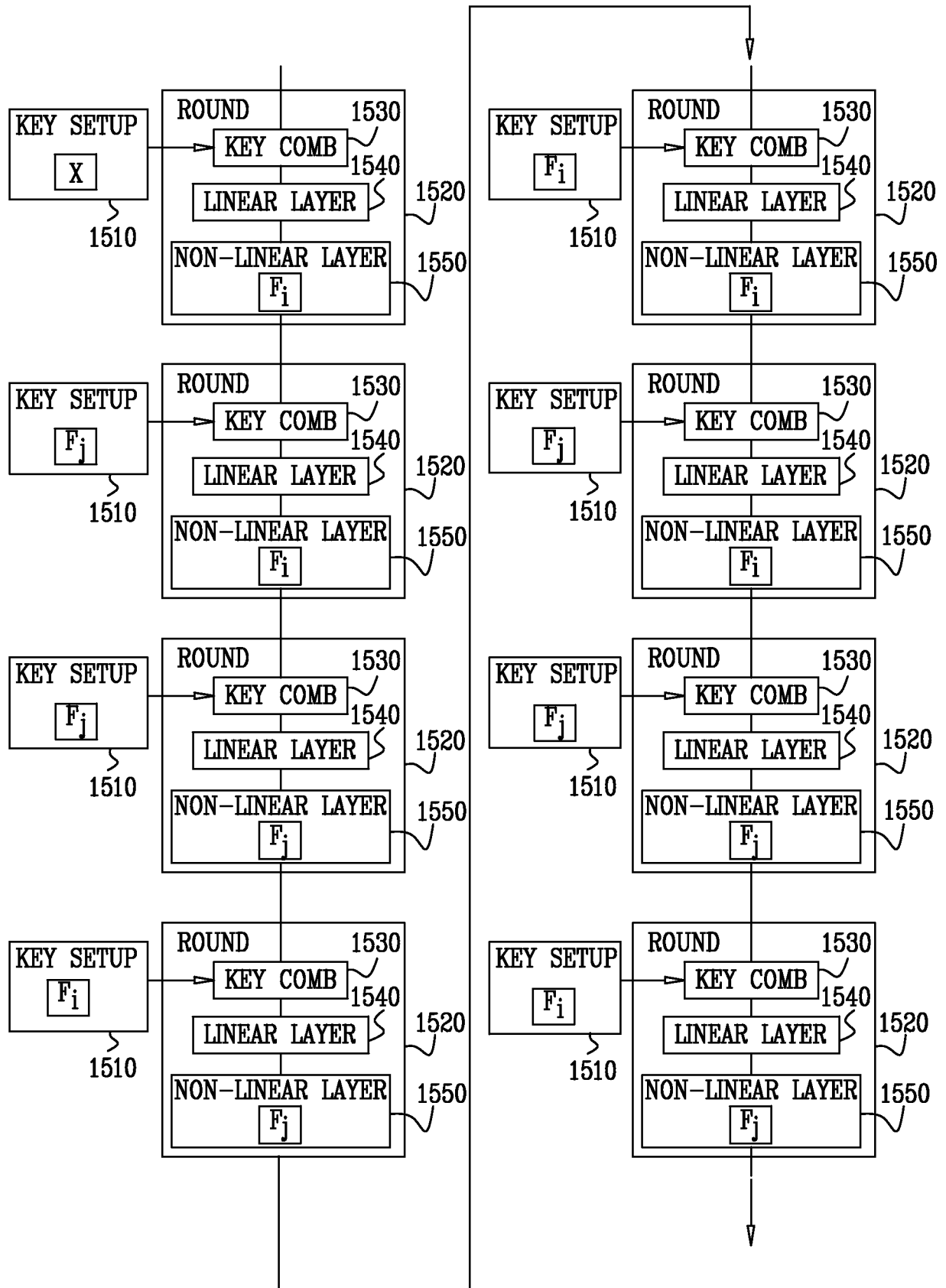


FIG. 24

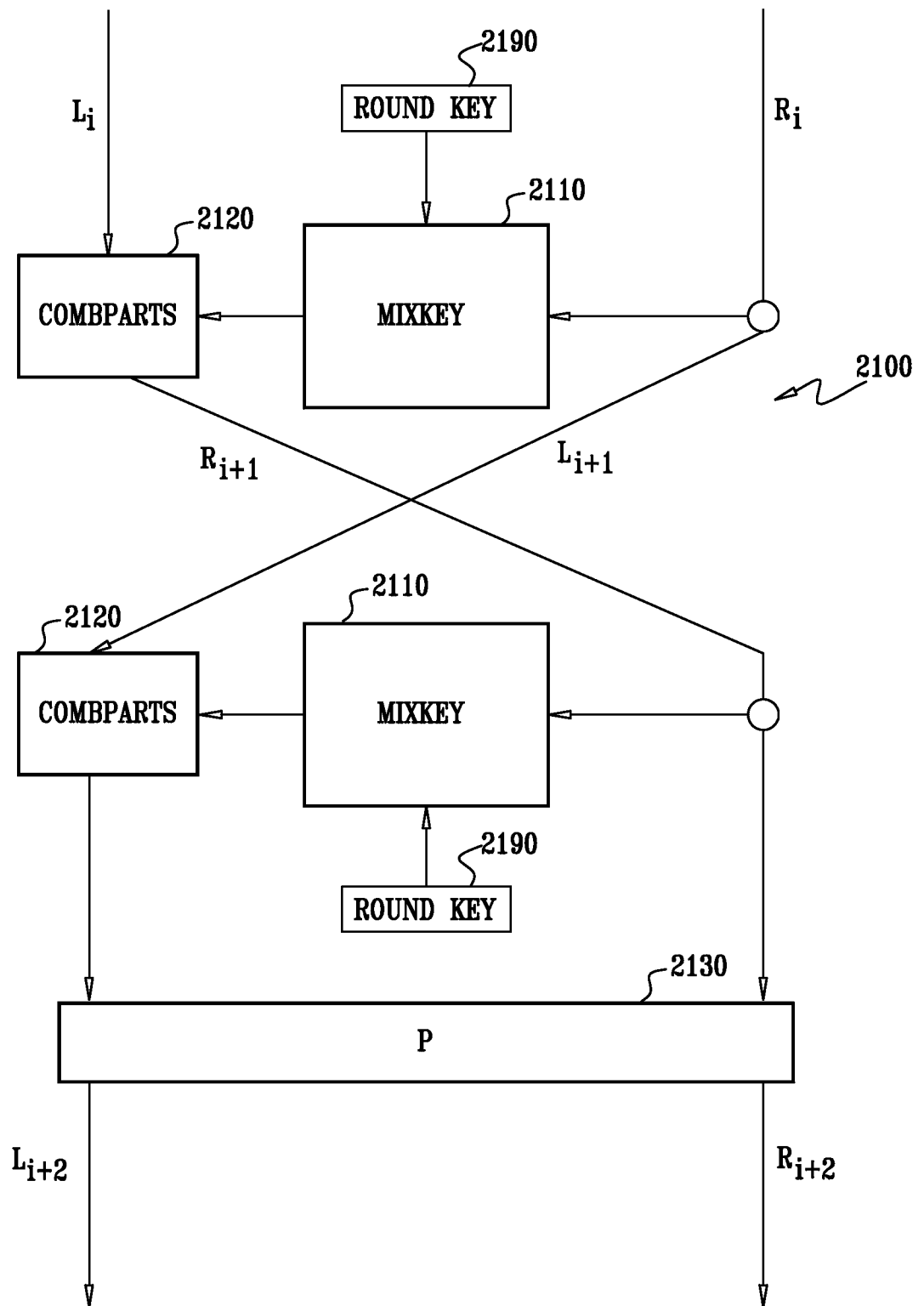


FIG. 25

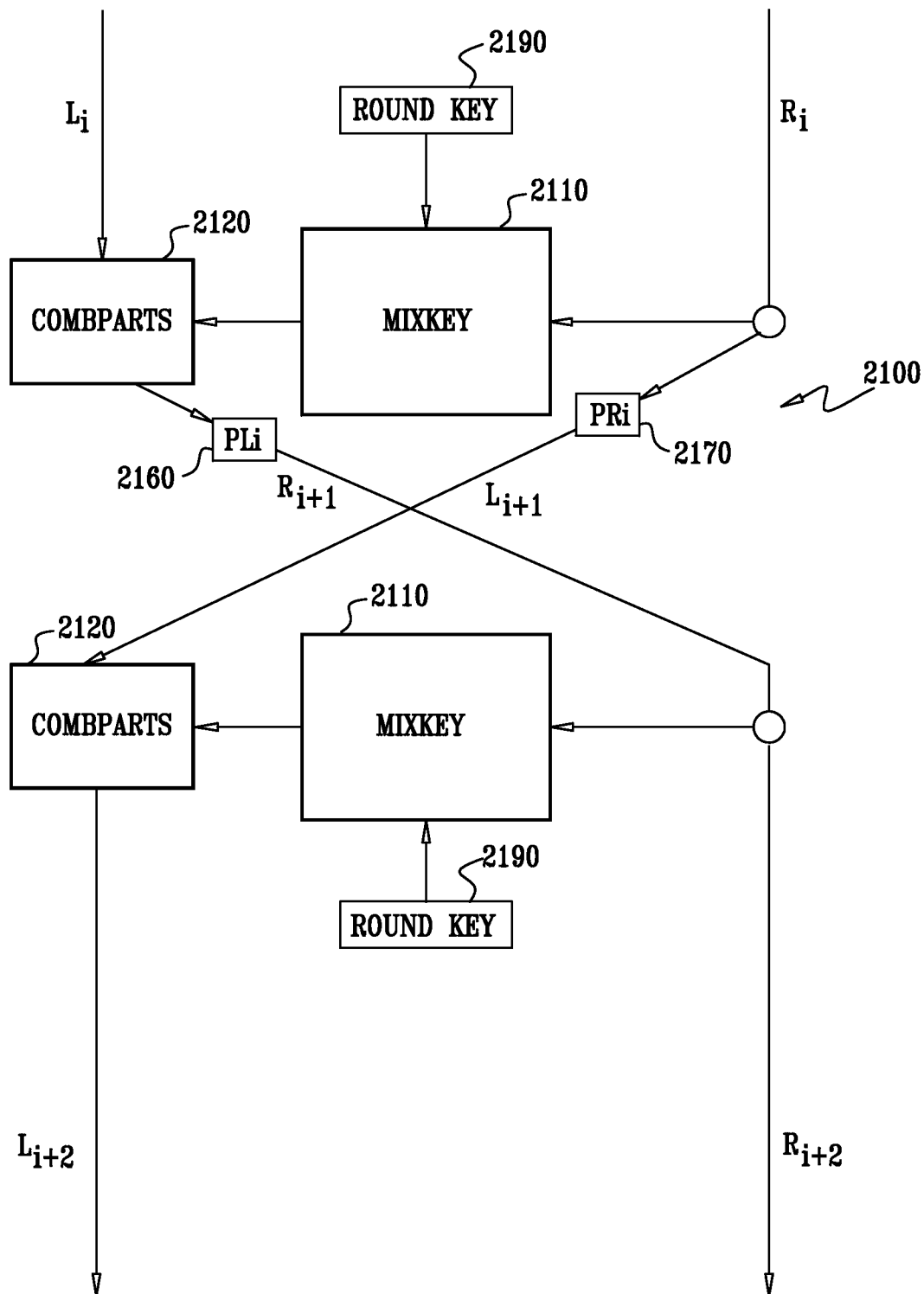


FIG. 26

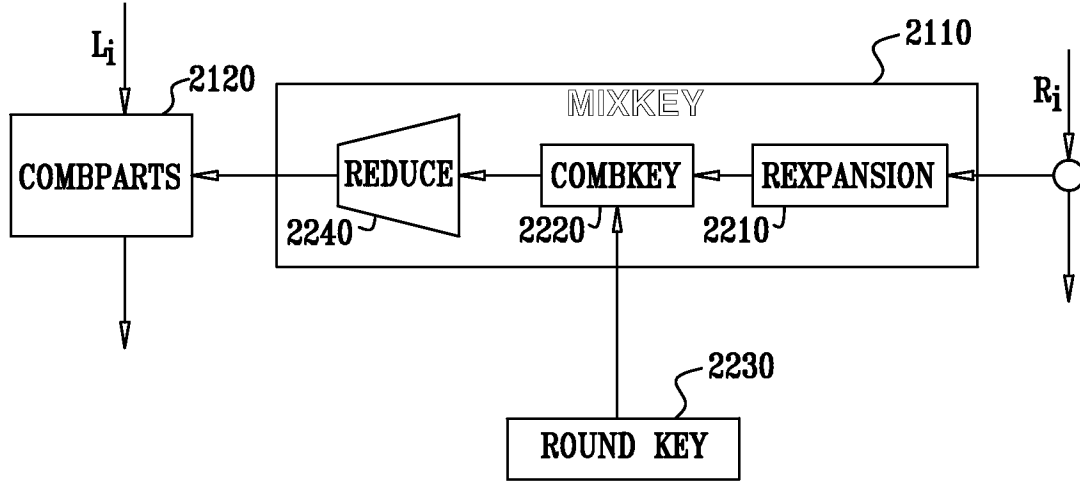
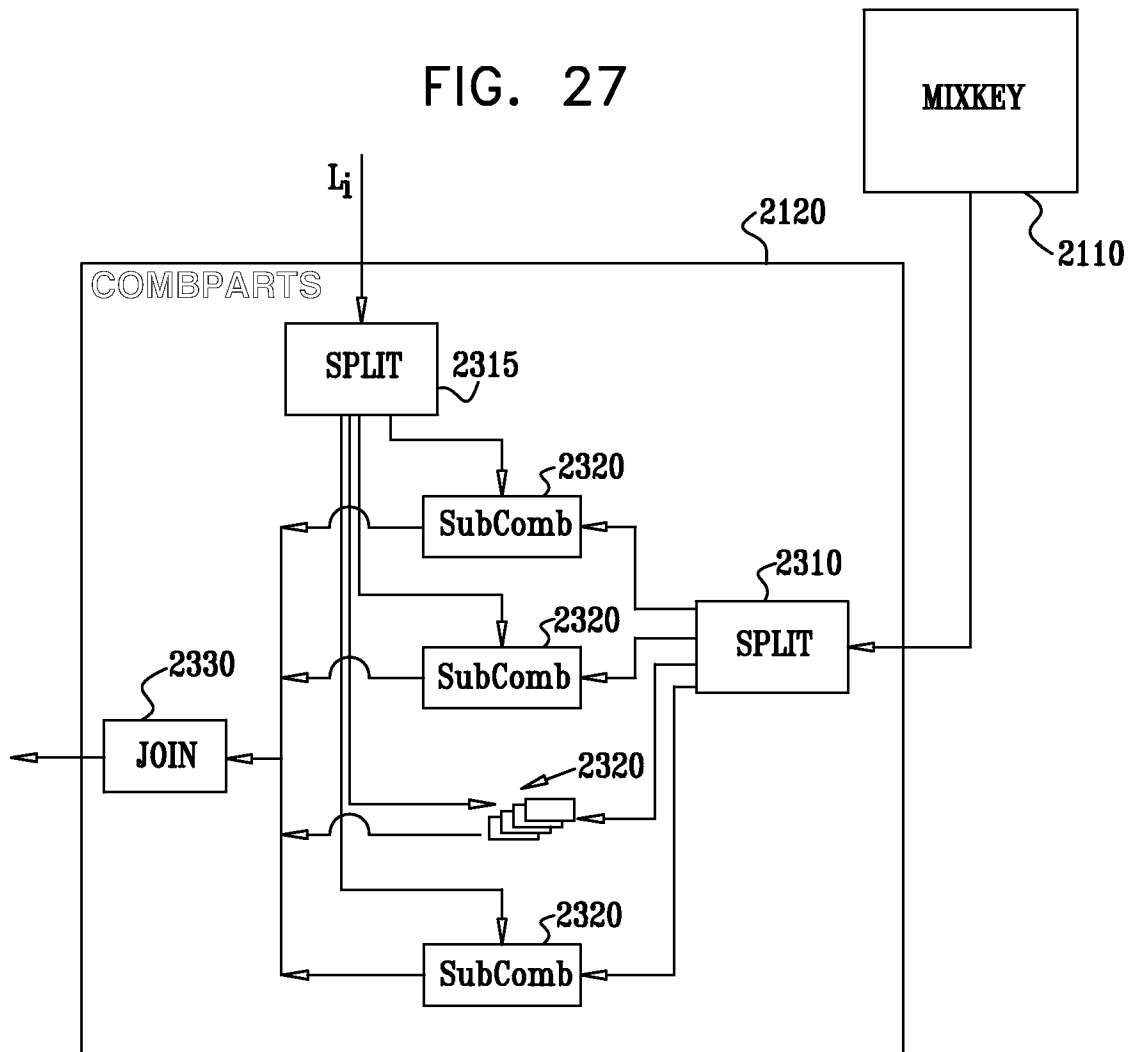


FIG. 27



INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2007/054592

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, IBM-TDB, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	MENEZES A ET AL: "Handbook of Applied Cryptography" HANDBOOK OF APPLIED CRYPTOGRAPHY, CRC PRESS SERIES ON DISCRETE MATHEMATICS AND ITS APPLICATIONS, BOCA RATON, FL, CRC PRESS, US, 1997, pages 223-282, XP002209334 ISBN: 0-8493-8523-7 cited in the application page 250, line 18 - page 256, line 16	1-83, 102-106, 108-119
Y	EP 1 133 100 A (TOKYO SHIBAURA ELECTRIC CO [JP]) 12 September 2001 (2001-09-12) paragraphs [0003], [0009], [0020], [0022], [0036], [0037]; figures 1,6 ----- -/--	1-83, 102-106, 108-119

☒ Further documents are listed in the continuation of Box C:☒ See patent family annex.

* Special categories of cited documents:

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

8 document member of the same patent family

Date of the actual completion of the international search

29 April 2008

Date of mailing of the international search report

23/07/2008

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Cretaine, Philippe

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2007/054592

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ANDERSON R ET AL: "SERPENT: A PROPOSAL FOR THE ADVANCED ENCRYPTION STANDARD" SERPENT: A PROPOSAL FOR THE ADVANCED ENCRYPTION STANDARD, XX, XX, 1998, pages 1-23, XP001028802 page 2, line 23 - page 4, line 11 -----	1-83, 102-106, 108-119
Y,P	WO 2007/043045 A (NDS LTD [GB]; MANTIN ITSIK [IL]; WAISBARD EREZ [IL]; KIPNIS AVIAD [IL]) 19 April 2007 (2007-04-19) cited in the application the whole document -----	1-83, 102-106, 108-119
A	P. JUNOD AND S. VAUDENAY: "FOX: a New Family of Block Ciphers" SELECTED AREAS IN CRYPTOGRAPHY 2004, 9 August 2004 (2004-08-09), XP002478578 Waterloo, CANADA the whole document -----	1-83, 102-106, 108-119

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2007/054592

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers allsearchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-83, 102-106, 108-119

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-83, 102-106, 108-119

Block encryption of data, comprising combining a block of data with a key, performing a plurality of S-BOX substitutions in layers, mixing and condensing the obtained transformed data.

2. claims: 84-101, 107

In a FEISTEL- like encryption structure, combining non-linearly data obtained from the right-part with data obtained from the left part to produce an output data.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2007/054592

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 1133100	A	12-09-2001	CN 1312630 A	12-09-2001
			DE 60119410 T2	02-11-2006
			JP 3499810 B2	23-02-2004
			JP 2001324924 A	22-11-2001
			KR 20010087344 A	15-09-2001
			US 2007058805 A1	15-03-2007
			US 2001024502 A1	27-09-2001
WO 2007043045	A	19-04-2007	AU 2006300744 A1	19-04-2007
			EP 1934883 A2	25-06-2008