

At the input of the first round, a register 204 takes as input the four 8-bit subwords resulting from the masking by the key k_0^i . This register makes it possible to synchronize the different binary flows, a flow corresponding to an input 8-bit subword. A first encryption key k_0^c , subdivided into four 8-bit subkeys denoted $k_0^c[0]$, $k_0^c[1]$, $k_0^c[2]$, $k_0^c[3]$, is applied in four XOR gates 206, 207, 208 on the masked subwords presented as output of said register 204. The key k_0^c is associated with the first encryption round. The four subwords of data masked by k_0^i and encrypted by k_0^c are then respectively processed by four processing blocks 209, 210, 211, 212 implemented in a memory, for example of ROM type. It should be noted that the complexity of implementation of functions, notably nonlinear functions, in a ROM memory increases exponentially with the number of inputs. The cryptographic algorithms incorporate this constraint and process the words to be encrypted as subwords of smaller size at the nonlinear function level in order to minimize the complexity of implementation.

At the input of each processing block, the 8-bit subwords are unmasked by applying the four 8-bit submasks $k_0^i[0]$, $k_0^i[1]$, $k_0^i[2]$, $k_0^i[3]$ using XOR gates 214. A nonlinear function is then applied, an S-box 213 being able to be used to implement it.

Upstream of the output of each processing block, an XOR gate 215 is used to mask the data at the output, so that the sensitive data are masked at the output of the first round stage by a mask k_1^i subdivided into four submasks $k_1^i[0]$, $k_1^i[1]$, $k_1^i[2]$, $k_1^i[3]$. A linear transformation $L_0()$ is applied at the round output, this having to be taken into account for the masking in the processing blocks. For this, the masking is carried out by using a modified mask $L_0^{-1}(k_1^i)$ of the mask k_1^i . This is subdivided into four 8-bit modified submasks $L_0^{-1}(k_1^i[0])$, $L_0^{-1}(k_1^i[1])$, $L_0^{-1}(k_1^i[2])$, $L_0^{-1}(k_1^i[3])$, the transformation $L_0^{-1}()$ corresponding to the inverse of the linear transformation $L_0()$.

At the output of the processing blocks 209, 210, 211, 212, the transformation $L_0()$ is applied to the binary words presented as output from said blocks. The result of the first encryption round corresponds to the binary subwords at the output of the linear transformation 216, said subwords corresponding to sensitive data masked by the mask k_1^i and therefore protected against attacks by observations.

CLAIMS

- 1- A cryptography circuit protected by masking, said circuit comprising
 5 means for encrypting binary words using at least one key k_r^c , means
 for applying linear processing operations (216) and nonlinear
 processing operations (226) to said words, means for masking said
 words, said circuit being characterized in that the binary words are
 10 unmasked (214) upstream of the nonlinear processing operations by
 using a mask k_r^i and masked (215) downstream of the nonlinear
 processing operations by using a mask k_{r+1}^i , the masks k_r^i and k_{r+1}^i
 forming part of a set of masks specific to each instance of the circuit;
- ~~2- The circuit as claimed in claim 1, characterized in that~~
 15 wherein the nonlinear processing operations, the unmasking (214)
 upstream of the nonlinear processing operations and the masking
 (215) downstream of the linear processing operations are
 implemented in ROM-type memories (209); and;
- ~~3- The circuit as claimed in any one of the preceding claims, characterized~~
 20 ~~in that the masks k_r^i are secondary masks deduced from primary~~
~~masks k^i such that $k_{r+1}^i = P(k_r^i)$ and $k_0^i = k^i$, the function $P(x)$~~
~~corresponding to a function for permutating elements of x , the ROM~~
~~memories remaining unchanged.~~
- 24- The circuit as claimed in claim 13, characterized in that the function
 25 $P(x)$ is a circular permutation, a secondary mask of index $r+1$ being
 deduced from a secondary mask of index r by circularly permutating
 the mask k_r^i of a chosen number of bits d .
- 35- The circuit as claimed in claim 13, characterized in that the main
 30 masks k^i are of length W and composed of an integer number of
 submasks of length S , the secondary masks k_r^i being generated by
 permutation of said submasks.

46- The circuit as claimed in claim 35, characterized in that the submasks of the secondary masks are chosen by using the expression:

$$k_{r+1}^i[x] = k_r^i[\text{mod}(x - Q, W / S)]$$

5

in which:

r is the round number;

i is a 4-bit number drawn randomly;

10

Q is an integer for controlling the rate of permutation between two consecutive secondary masks k_r^i and k_{r+1}^i ;

S is the length of a submask expressed in bits;

W is the length of the main mask expressed in bits;

mod() is a defined function such as $\text{mod}(a, b) = a \text{ modulo } b$, a and b being integer numbers.

15

57- The circuit as claimed in any one of claims 13 to 46, characterized in that the main encryption mask k^i is modified regularly by randomly choosing a mask k^i from a set of main masks stored in the circuit.

20

68- The circuit as claimed in claim 57, characterized in that the set of main masks stored in the circuit (303) is different from one circuit to another.

79- The circuit as claimed in claim 68, characterized in that the set of main masks is obtained using an internal mask generation circuit.

25

840- The circuit as claimed in any one of the preceding claims, characterized in that the Hamming distance between two masks k_r^i and k_{r+1}^i is equal to $S/2$.

30

944- The circuit as claimed in any one of the preceding claims, characterized in that the Hamming weight of a mask k^i is equal to $W/2$.

35

102- The circuit as claimed in any one of the preceding claims, characterized in that the nonlinear processing operations are implemented using S-boxes (213, 310).

- 113- The circuit as claimed in any one of the preceding claims, characterized in that the nonlinear processing operations are applied after the linear processing operations in one and the same combinatory block just before the sampling of the result in a register.
- 5 124- The circuit as claimed in any one of the preceding claims, characterized in that it is implemented in an FPGA.
- 10 135- The circuit as claimed in claim 113, characterized in that the set of main masks is obtained using the modification of the FPGA circuit configuration file.
- 15 146- The circuit as claimed in any one of claims 113 or 124, characterized in that it comprises dynamic reconfiguration means making it possible to update the set of main masks and the tables implementing the parts of the circuit corresponding to the nonlinear processing operations.
- 157- The circuit as claimed in any one of claims 1 to 113, characterized in that it is implemented in an ASIC.