1 Specifications of -FUTURE-

-FUTURE- is a new SPN based block cipher and consists of 10 rounds in a fully unrolled fashion. It accept 128-bit keys and have a block size of 64-bit.

1.1 Round Function.

One encryption round of -FUTURE- is composed of four operations in the following order: SubCells, MixColumns, ShiftRows and AddRoundKey (see illustration in Figure 1). The cipher receives an 64-bit plaintext $P = b_0 b_1 b_2 \dots b_{62} b_{63}$ as the cipher state S, where b_0 being the most significant bit. The cipher state can also be expressed as 16 many 4-bit cells as follows:

$$S = \begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{bmatrix},$$

i.e. $s_i \in \{0,1\}^4$. The i-th round output state is defined as S_i , namely $S_0 = P$.

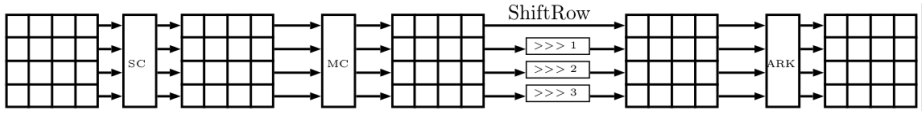

Fig. 1. The round function applies four different transformations: SubCells (SC), MixColumns (MC), ShiftRows (SR) and AddRoundKey (ARK).

SubCells. A 4-bit Sbox S is applied to every cell of the cipher internal state.

$$s_i \leftarrow S(s_i) \quad \text{for } i = 0, 1, \dots, 15.$$

The SBox S is a composition of two low hardware cost SBoxes S_1 and S_2 i.e. $S(s_i) = S_1(S_2(s_i))$ for $i = 0, 1, \dots, 15$. The Sboxes in hexadecimal notation is given by the following table.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S_1(x)$	1	3	0	2	7	5	4	6	9	a	8	b	f	e	c	d
$S_2(x)$	0	1	2	3	4	d	6	f	8	9	e	7	c	5	a	b
$S(x)$	1	3	0	2	7	e	4	d	9	a	c	6	f	5	8	b

MixColumns. -FUTURE- uses an MDS matrix M for the MixColumns operation. M is applied to every 16-bit column of the state S,

$$(s_i, s_{i+1}, s_{i+2}, s_{i+3}) \leftarrow M \cdot (s_i, s_{i+1}, s_{i+2}, s_{i+3})^t$$

for $i = 0, 4, 8, 12$.

The MDS matrix M is constructed by composition of 4 sparse matrices M_1, M_2, M_3 and M_4 of order 4 i.e., $M = M_1 M_2 M_3 M_4$, where

$$M_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & 0 & 1 & \alpha \\ 1 & 0 & 0 & 0 \\ \alpha^3 + 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, M_3 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ \alpha^3 + 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } M_4 = M_1 \tag{1}$$

The multiplications between matrices and vectors are performed over $\mathbb{F}_{2^4}$ defined by the primitive polynomial $x^4 + x + 1$ and α is the primitive element of the field.

ShiftRows. Row i of the array state is rotated i cell positions to the right, for $i = 0, 1, 2, 3$, i.e.,

$$\begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{bmatrix} \leftarrow \begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_{13} & s_1 & s_5 & s_9 \\ s_{10} & s_{14} & s_2 & s_6 \\ s_7 & s_{11} & s_{15} & s_3 \end{bmatrix}.$$

Note that in the ShiftRows operation of AES [2] and LED [3] the row i of the array state is rotated i cell positions to the left, for $i = 0, 1, 2, 3$.

AddRoundKey. Given round key RK_i for $1 \leq i \leq 10$, the i-th 64-bit round key RK_i is XORed to the state S.

1.2 Data Processing

The data processing part of -FUTURE- for encryption consisting of 10 rounds, F, takes a 64-bit data $X \in \{0,1\}^{64}$, whitening keys $WK \in \{0,1\}^{64}$ and 10 round keys $RK_i \in \{0,1\}^{64}$ $(1 \leq i \leq 10)$ as the inputs, and outputs a 64-bit data $Y \in \{0,1\}^{64}$. F is defined as follows:

$$F = \begin{cases} \{0,1\}^{64} \times \{0,1\}^{64} \times \left\{\{0,1\}^{64}\right\}^{10} \rightarrow \{0,1\}^{64} \\ (X, WK, RK_1, RK_2, \ldots, RK_{10}) \rightarrow Y \end{cases}$$

Input: X and $WK, RK_1, RK_2, \ldots, RK_{10}$
Initialization: $S \leftarrow \mathbf{KeyAdd}(X, WK)$;
for $i \leftarrow 1$ **to** 9 **do**
  $S \leftarrow \mathbf{SubCell}(S)$;
  $S \leftarrow \mathbf{MixColumn}(S)$;
  $S \leftarrow \mathbf{ShiftRows}(S)$;
  $S \leftarrow \mathbf{AddRoundKey}(S, RK_i)$;
end
$S \leftarrow \mathbf{SubCell}(S)$;
$S \leftarrow \mathbf{ShiftRows}(S)$;
$S \leftarrow \mathbf{AddRoundKey}(S, RK10)$;
Output: Y

Algorithm 1: Encryption Function of -FUTURE-

The inverse data processing part F^{-1} of -FUTURE- operates as follows:

$$F = \begin{cases} \{0,1\}^{64} \times \left\{\{0,1\}^{64}\right\}^{10} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64} \\ (Y, RK_{10}, RK_9, \ldots, RK_1, WK) \rightarrow X \end{cases}$$

Input: Y and $RK_{10}, RK_9, \ldots, RK_1, WK$
$S \leftarrow \mathbf{KeyAdd}(Y, RK_{10})$;
$S \leftarrow \mathbf{InvShiftRows}(S)$;
for $i \leftarrow 1$ **to** 9 **do**
  $S \leftarrow \mathbf{InvSubCell}(S)$;
  $S \leftarrow \mathbf{AddRoundKey}(S, RK_i)$;
  $S \leftarrow \mathbf{InvShiftRows}(S)$;
  $S \leftarrow \mathbf{InvMixColumn}(S)$;
end
$S \leftarrow \mathbf{InvSubCell}(S)$;
$S \leftarrow \mathbf{KeyAdd}(S, WK)$;
Output: X

Algorithm 2: Decryption Function of -FUTURE-

1.3 Key schedule and round constants.

-FUTURE- uses a 128-bit secret key $K = k_0 k_1 \ldots k_{127}$. It splits K in two equal parts K_0 and K_1 for the round key and whitening key generation i.e. $K = K_0 || K_1$, where $K_0 = k_0 k_1 \ldots k_{63}$ and $K_1 = k_{64} k_{65} \ldots k_{127}$ are two 64-bit key. It uses K_0 as whitening key and the round key RK_i $(1 \leq i \leq 10)$ generation is as follows (see Figure 2):

$$RK_i = \begin{cases} K_0 \leftarrow 5\lfloor i/2 \rfloor <<< K_0 & \text{if } 2 \mid i \\ K_1 \leftarrow 5\lfloor i/2 \rfloor <<< K_1 & \text{if } 2 \nmid i \end{cases}$$

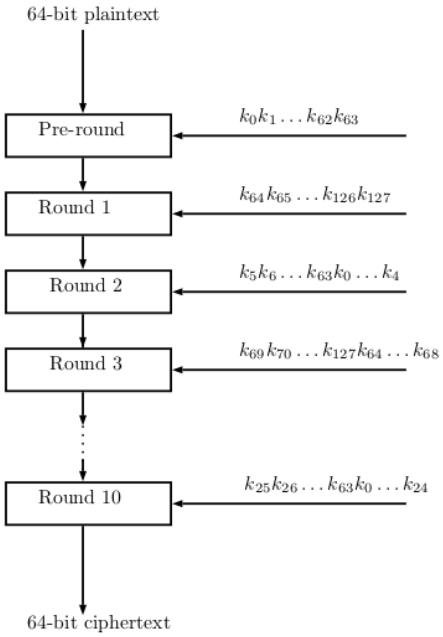

Fig. 2. Round Key Generation