
MODULE *des*

EXTENDS *Naturals, Sequences, Integers*

VARIABLES *state, roundKey, round, Nr, encrypt*

$IP \triangleq \langle 58, 50, 42, 34, 26, 18, 10, 2,$
60, 52, 44, 36, 28, 20, 12, 4,
62, 54, 46, 38, 30, 22, 14, 6,
64, 56, 48, 40, 32, 24, 16, 8,
57, 49, 41, 33, 25, 17, 9, 1,
59, 51, 43, 35, 27, 19, 11, 3,
61, 53, 45, 37, 29, 21, 13, 5,
63, 55, 47, 39, 31, 23, 15, 7 \rangle

$FP \triangleq \langle 40, 8, 48, 16, 56, 24, 64, 32,$
39, 7, 47, 15, 55, 23, 63, 31,
38, 6, 46, 14, 54, 22, 62, 30,
37, 5, 45, 13, 53, 21, 61, 29,
36, 4, 44, 12, 52, 20, 60, 28,
35, 3, 43, 11, 51, 19, 59, 27,
34, 2, 42, 10, 50, 18, 58, 26,
33, 1, 41, 9, 49, 17, 57, 25 \rangle

$E \triangleq \langle 32, 1, 2, 3, 4, 5,$
4, 5, 6, 7, 8, 9,
8, 9, 10, 11, 12, 13,
12, 13, 14, 15, 16, 17,
16, 17, 18, 19, 20, 21,
20, 21, 22, 23, 24, 25,
24, 25, 26, 27, 28, 29,
28, 29, 30, 31, 32, 1 \rangle

$P \triangleq \langle 16, 7, 20, 21, 29, 12, 28, 17,$
1, 15, 23, 26, 5, 18, 31, 10,
2, 8, 24, 14, 32, 27, 3, 9,
19, 13, 30, 6, 22, 11, 4, 25 \rangle

$SBox1 \triangleq \langle \langle 14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7 \rangle,$
 $\langle 0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8 \rangle,$
 $\langle 4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0 \rangle,$
 $\langle 15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13 \rangle \rangle$

$SBox2 \triangleq \langle \langle 15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10 \rangle,$
 $\langle 3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5 \rangle,$
 $\langle 0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15 \rangle,$
 $\langle 13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9 \rangle \rangle$

$$SBox3 \triangleq \langle \langle 10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8 \rangle, \\ \langle 13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1 \rangle, \\ \langle 13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7 \rangle, \\ \langle 1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12 \rangle \rangle$$

$$SBox4 \triangleq \langle \langle 7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15 \rangle, \\ \langle 13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9 \rangle, \\ \langle 10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4 \rangle, \\ \langle 3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14 \rangle \rangle$$

$$SBox5 \triangleq \langle \langle 2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9 \rangle, \\ \langle 14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6 \rangle, \\ \langle 4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14 \rangle, \\ \langle 11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3 \rangle \rangle$$

$$SBox6 \triangleq \langle \langle 12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11 \rangle, \\ \langle 10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8 \rangle, \\ \langle 9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6 \rangle, \\ \langle 4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13 \rangle \rangle$$

$$SBox7 \triangleq \langle \langle 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1 \rangle, \\ \langle 13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6 \rangle, \\ \langle 1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2 \rangle, \\ \langle 6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12 \rangle \rangle$$

$$SBox8 \triangleq \langle \langle 13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7 \rangle, \\ \langle 1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2 \rangle, \\ \langle 7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8 \rangle, \\ \langle 2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11 \rangle \rangle$$

$$Permute(bitSeq, perm) \triangleq \\ [i \in 1 \dots Len(perm) \mapsto bitSeq[perm[i]]]$$

$$FinalPermutation(s) \triangleq \\ Permute(s[1] \circ s[2], FP)$$

$$GenerateRoundKey(k, r) \triangleq \\ [i \in 1 \dots Len(k) \mapsto k[((i + r - 1) \% Len(k)) + 1]]$$

$$AccessSBox(SBox, Row, Col) \triangleq \\ \text{IF } Row \in 1 \dots 4 \wedge Col \in 1 \dots 16 \text{ THEN } SBox[Row][Col] \text{ ELSE } 0$$

$$F(R, K) \triangleq$$

LET

$$E_R \triangleq Permute(R, E)$$

$$XOR_Result \triangleq [i \in 1 \dots 48 \mapsto (E_R[i] + K[i]) \% 2]$$

$$SBox_Output1 \triangleq AccessSBox(SBox1, (XOR_Result[1] * 2) + XOR_Result[6] + 1, ((XOR_Result[1] * 2) + XOR_Result[6] + 1) \% 16)$$

$SBox_Output2 \triangleq AccessSBox(SBox2, (XOR_Result[7] * 2) + XOR_Result[12] + 1, ((XOR_Result[7] * 2) + XOR_Result[12] + 1) \% 2)$
 $SBox_Output3 \triangleq AccessSBox(SBox3, (XOR_Result[13] * 2) + XOR_Result[18] + 1, ((XOR_Result[13] * 2) + XOR_Result[18] + 1) \% 2)$
 $SBox_Output4 \triangleq AccessSBox(SBox4, (XOR_Result[19] * 2) + XOR_Result[24] + 1, ((XOR_Result[19] * 2) + XOR_Result[24] + 1) \% 2)$
 $SBox_Output5 \triangleq AccessSBox(SBox5, (XOR_Result[25] * 2) + XOR_Result[30] + 1, ((XOR_Result[25] * 2) + XOR_Result[30] + 1) \% 2)$
 $SBox_Output6 \triangleq AccessSBox(SBox6, (XOR_Result[31] * 2) + XOR_Result[36] + 1, ((XOR_Result[31] * 2) + XOR_Result[36] + 1) \% 2)$
 $SBox_Output7 \triangleq AccessSBox(SBox7, (XOR_Result[37] * 2) + XOR_Result[42] + 1, ((XOR_Result[37] * 2) + XOR_Result[42] + 1) \% 2)$
 $SBox_Output8 \triangleq AccessSBox(SBox8, (XOR_Result[43] * 2) + XOR_Result[48] + 1, ((XOR_Result[43] * 2) + XOR_Result[48] + 1) \% 2)$
 $ToBits4(n) \triangleq \langle (n \div 8) \% 2, (n \div 4) \% 2, (n \div 2) \% 2, n \% 2 \rangle$
 $P_Input \triangleq ToBits4(SBox_Output1) \circ ToBits4(SBox_Output2) \circ ToBits4(SBox_Output3) \circ ToBits4(SBox_Output4) \circ ToBits4(SBox_Output5) \circ ToBits4(SBox_Output6) \circ ToBits4(SBox_Output7) \circ ToBits4(SBox_Output8)$
 $P_Output \triangleq Permute(P_Input, P)$
IN
 P_Output

$Round(L, R, K) \triangleq$
LET $newR \triangleq [i \in 1 \dots 32 \mapsto (L[i] + F(R, K)[i]) \% 2]$
 $newL \triangleq R$
IN $\langle newL, newR \rangle$

$EncryptRound(L, R, K) \triangleq$
LET $result \triangleq Round(L, R, K)$ IN $result$

$DecryptRound(L, R, K) \triangleq$
LET $result \triangleq Round(R, L, K)$ IN $result$

$DESProcess(e, s, k, r) \triangleq$
IF e THEN $EncryptRound(s[1], s[2], GenerateRoundKey(k, r))$
ELSE $DecryptRound(s[1], s[2], GenerateRoundKey(k, Nr - r + 1))$

$NextRound \triangleq$
 $\wedge round < Nr$
 $\wedge roundKey' = roundKey$
 $\wedge Nr' = Nr$
 $\wedge round' = round + 1$
 $\wedge encrypt' = encrypt$
 \wedge IF $round' = Nr + 1$ THEN $state' = FinalPermutation(state')$ ELSE $state' = DESProcess(encrypt, state', k, Nr')$

$Init \triangleq$
 $\wedge state = \langle [i \in 1 \dots 32 \mapsto 0], [i \in 1 \dots 32 \mapsto 1] \rangle$
 $\wedge roundKey = [i \in 1 \dots 48 \mapsto 1]$
 $\wedge round = 0$
 $\wedge Nr = 16$
 $\wedge encrypt = TRUE$

$$Spec \stackrel{\Delta}{=} Init \wedge \square [NextRound]_{\langle state, round, roundKey, Nr, encrypt \rangle}$$
