Question 2

In the previous question we computed the SBox for Simplified DES. There are multiple ways to compute the inverse SBox. You can find each element of L in the SBox and figure out which element maps to it. Or you can reverse each of the steps in the previous algorithm. Write a Sage function to calculate the inverse SBox matrix.

Solution

In the previous question we computed the SBox for Simplified DES. There are multiple ways to compute the inverse SBox. You can find each element of L in the SBox and figure out which element maps to it. Or you can reverse each of the steps in the previous algorithm. Write a Sage function to calculate the inverse SBox matrix.

def SAES\_InverseLinearTransformElements(M):

next\_M = [];

for j in xrange(4):

row = [L(A.transpose().inverse()\*(V(M[j][k]) + b)) for k in xrange(4)];

next\_M.append(row);

return next\_M;

def SAES\_ComputeInverseSBoxMatrix():

M0 = SAES\_InitSBoxMatrix();

M1 = SAES\_InverseLinearTransformElements(M0);

M2 = SAES\_MapToInverse(M1);

InverseSBox\_matrix\_output = Matrix(L, M2);

return InverseSBox\_matrix\_output;