Onewayx.js

Apply addition, exclusive or, or multiplication to a matrix by rows, columns, left diagonals and right diagonals and then take the modulus 256 to create a one way function.

0 1 2 3

4 5 6 7

8 9 10 11

12 13 14 15

Xxxxxxxxxxxx

//onewayx.js

function oneway1(mode,A,B,C,D,E){

switch(mode){

case 0:

return A+B+C+D+E;

case 1:

return A^B^C^D^E;

case 2:

return A\*B\*C\*D\*E;

}//end switch

}//end oneway1

function modit(A,B){

for(var i=0;i<16;i++)A[i]=B[i]%256;

}//end modit

function fillit(A,b){

for(var i=0;i<16;i++)A[i]=b;

}//end fillit

function oneway(mode,A,B){

//rows

A[0]=oneway1(mode,A[0],B[0],B[1],B[2],B[3]);

A[1]=oneway1(mode,A[1],B[4],B[5],B[6],B[7]);

A[2]=oneway1(mode,A[2],B[8],B[9],B[10],B[11]);

A[3]=oneway1(mode,A[3],B[12],B[13],B[14],B[15]);

//columns

A[4]=oneway1(mode,A[4],B[0],B[4],B[8],B[12]);

A[5]=oneway1(mode,A[5],B[1],B[5],B[9],B[13]);

A[6]=oneway1(mode,A[6],B[2],B[6],B[10],B[14]);

A[7]=oneway1(mode,A[7],B[3],B[7],B[11],B[15]);

//diagonals\

A[8]=oneway1(mode,A[8],B[0],B[5],B[10],B[15]);

A[9]=oneway1(mode,A[9],B[1],B[6],B[11],B[12]);

A[10]=oneway1(mode,A[10],B[2],B[7],B[8],B[13]);

A[11]=oneway1(mode,A[11],B[3],B[4],B[9],B[14]);

//diagonals/

A[12]=oneway1(mode,A[12],B[3],B[6],B[9],B[12]);

A[13]=oneway1(mode,A[13],B[0],B[7],B[10],B[13]);

A[14]=oneway1(mode,A[14],B[1],B[4],B[11],B[14]);

A[15]=oneway1(mode,A[15],B[2],B[5],B[8],B[15]);

}//end oneway

xxxxxxxxxxxxxx

Xxxxxxxxxxxxxxxxxx

The inital data

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

use mutltiply

Data after applying oneway function and modit

24 144 104 160 73 144 137 0 32 136 208 184 56 208 8 32

Do it again using xor

The inital data

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Data after applying oneway function and modit

5 13 5 29 1 1 1 17 29 5 13 5 5 13 5 29

Do it again using addition

The inital data

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Data after applying oneway function and modit

11 27 43 59 29 33 37 41 35 35 35 35 35 35 35 35

Xxxxxxxxxxxxxxxxxxxxxxx