Programming and Data Structure Lab [CS19003][Section-1] Assignment-8

Satrajit Ghosh Sudebkumar P Pal

June 1, 2022

Write a C program for the task as described below. Save the file as **A08_Roll_Number.c**. Build and run to check your program. Upload the .c file for the assignment in MS teams.

1. Generate a matrix M of dimension $n \times n$, where n is an input form the user. Call two functions with appropriate parameters to populate the matrix with 1's and 0's. After each pattern generation call a print matrix function with appropriate parameters to print the matrix. Output must be two well-defined patterns of your choice (e.g. a pyramid with 0's and 1's). [20]

2. Take an input from the user to encrypt either the matrix with pattern-1 or pattern-2. The encryption function BadEnc generates a random key $k \in \{0, 1\}$ and encrypts each entry of the matrix by computing the following function:

   $$Enc(M[i][j], k) = (M[i][j] + k)\%2, \forall i, j\cdot$$

   From the main function call the print matrix function to print the encrypted matrix. Note that BadEnc must return the key k. Call a function BadDec to decrypt the encrypted matrix. The decryption function takes the encrypted matrix M' and the key k as input and decrypt each entry by computing the following function:

   $$Dec(M'[i][j], k) = (M'[i][j] + k)\%2, \forall i, j\cdot$$

   Print the decrypted matrix. Observe that this mode of encryption is not good to hide the pattern. [30]

3. Use a better mode of encryption to encrypt the matrix M. Let us call it GoodEnc. GoodEnc generates random keys $k_{i,j} \in \{0, 1\}$ for each $M[i][j]$ and generate the encrypted entries of the matrix by computing:

   $$Enc(M[i][j], k) = (M[i][j] + k_{i,j})\%2\cdot$$

   From the main function call the print matrix function to print the encrypted matrix. Note that GoodEnc must return all the keys used to encrypt the matrix. Call a function GoodDec to decrypt the encrypted matrix. The decryption function takes the encrypted matrix M' and the keys as input and decrypt each entry by computing the following function:

   $$Dec(M'[i][j], k_{i,j}) = (M'[i][j] + k_{i,j})\%2, \forall i, j\cdot$$

   Print the decrypted matrix. Observe that this mode of encryption actually hides the pattern. [50]

Sample Output: Here is one sample output:

Please enter n: 8

Matrix with pattern-1:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Matrix with pattern-2:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Enter 1 to play with pattern-1 matrix
Enter 2 to play with pattern-2 matrix
Enter 3 to exit

Please Enter your choice: 1

Bad Encrypted Matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix after Bad Decryption:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Good Encrypted Matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrix after Good Decryption:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Enter 1 to play with pattern-1 matrix
Enter 2 to play with pattern-2 matrix
Enter 3 to exit

Please Enter your choice: 3

Good bye!

Note: The encryption function we are using for each entry of the matrix is known as **One Time Pad**. Though, this encryption scheme is **perfectly secure**, from this experiment we can see that one need to be very careful while using this scheme; In fact one should be careful while using any cryptographic primitive to design a secure system. In the first case we are using same key for all the entries of the matrix and clearly that is not sufficient to hide the pattern. However, using independent keys we can hide the pattern. You can also extend this experiment to see why **Electronic Codebook (ECB)** mode of encryption is not secure.