Power analysis attacks

A set of final year students of Department of Computer Engineering of Faculty of Engineering is doing a project on power analysis attacks.
Power analysis attack is a form of side channel attack in which, the adversary exploits power consumed by a cryptographic device during encryption to obtain the secret key.

The idea of a power analysis attack is to compare electrical power leakage of a cryptographic device with a set of key dependent leakage predictions, in order to identify which key most probably gave rise to the actual leakage. The set of key dependent leakage predictions is usually known as a power model. Comparison between the two sets of data is made through a statistical method known as a side channel distinguisher. Types of power analysis attacks vary depending on the distinguisher.

They are using two sets of data to extract the key. The first data set includes data collected by measuring the power consumption of the encryption process. The second includes hypothetical data generated by a power model. The key is extracted by comparing these data sets.

Input Format

Wave data matrix: (double)10x50 matrix
Hypothetical data matrix: (double)10x50 matrix

Constraints

All the values are in type double
Both matrices are of the same size: 10x50

Output Format

Similarity matrix: (double, rounded to 6 decimal points)10x10 matrix

Sample Input

Input file

Sample Output

Output file

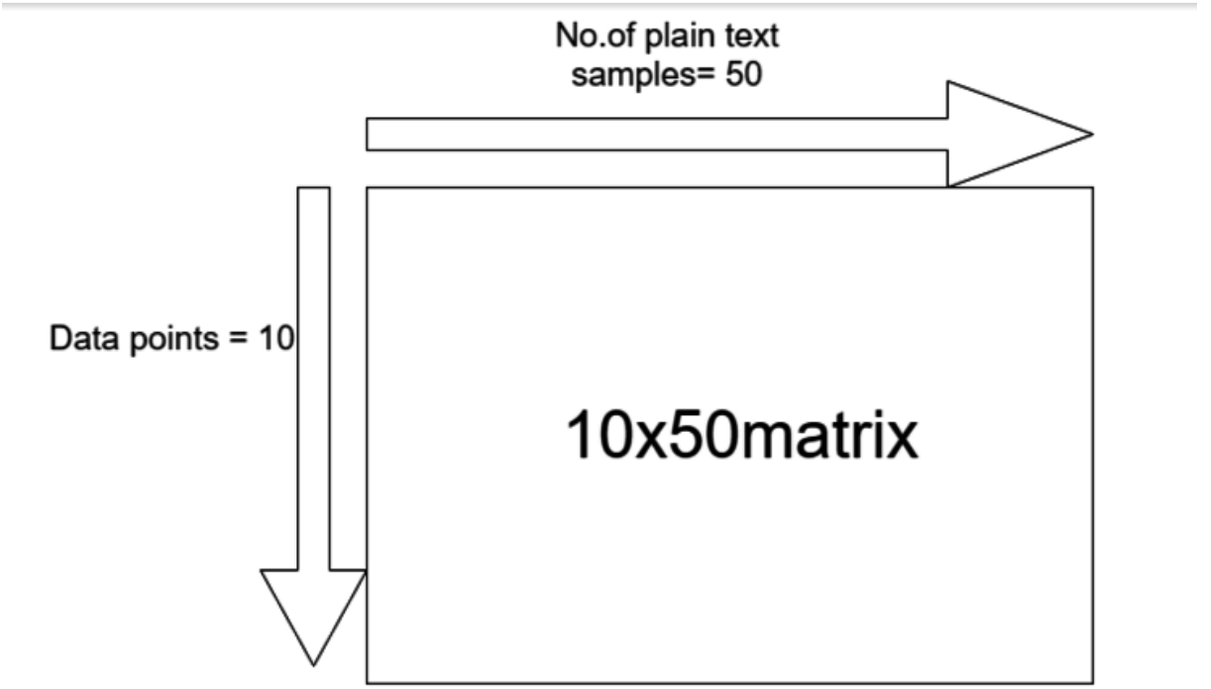
Explanation

The two data sets are given in matrix form. The first represents the collected power consumption data. Using 50 plain text samples, 10 data points have been collected. For instance the following matrix includes data collected from 3 plain text samples, using 4 data points. Please note that the original data has been multiplied by a factor of 100.

256.8	326.6	586.56
205.42	415	125
323	565.65	148.7
486	256	454

Similarly, the second matrix contains the hypothetical data. The columns represents each plain text sample and the values represent the hypothetical power consumption values for each key byte of the sample. Since each plain text sample contains 10 bytes, this matrix is of size 10x50.

The diagram explains the configuration of the matrix.



To find the similarity, the students need to calculate the similarity matrix. The similarity matrix is calculated by comparing each row (hence one row of the matrix) of one data set with every row of the other set. Hence the similarity matrix must be of size 10x10. If you are calculating the similarity between the second sample of the first data set and the third sample of the second data set then the result is stored in cell (2, 3) of the similarity matrix.
However in order to calculate the similarity three variables must be calculated. They are the marginal probability of each of the rows from matrix1 and matrix2, and the joint probability of each pair of rows (a pair consists one row from matrix 1 and one row from matrix 2) .
To calculate the marginal probability of each row, histogram method can be used.

Here, first the values must be normalized. Take the normalizing range between the minimum and the maximum values of each row. When normalizing, values should be rounded using floor. If the value is X and

minimum value of the row is min then,

normalized value of X = floor(X) - floor(min)

numberOfStates1 (explained below) is calculated as

numberOfStates1 = floor(max)-floor(min) + 1

Then the number of instances of each value must be calculated and by dividing the number of instances by the length of the row (i.e. 50), marginal probability array can be calculated.

Note that there could be 0 instances of some values. Make sure to include them as well. Hence the size of the probability array will be 'numberOfStates1'.

After calculating the marginal probability, the joint probability can be calculated. This must be calculated for each pair of rows from set1 and set2. Use the following equation to calculate this.

For i=0 to number of readings

jointProbability[normal2[i] * numberOfStates1 + normal1[i]] += 1/(number of readings)

normal1[] is the normalized row of the first matrix, normal2[] is the normalized row of the second matrix and numberOfStates1 is derived as mentioned above. numberOfStates2 is the corresponding value of matrix 2.

Hence jointProbability array is of the size numberofStates1*numberofstates2

The similarity value for each pair of rows is calculated as follows.

Similarity = 0

For i=0 to (numberOfStates1 * numberOfStates2)

fIndex = i%numberOfStates1

sIndex = i/numberOfStates1

val1 = jointProbabilty[i]

val2 = marginalProbabilityOfSet1[fIndex]

val3 = marginalProbabilityOfSet2[sIndex]

if (val1>0 and val2>0 and val3>0)
Similarity += val1*log(val1/(val2*val3))

Similarity /= log(2)

Your program must take the power consumption data and the hypothetical power values as inputs. (i. e.

10x50 matrix and another 10x50 matrix)

It must output the Similarity matrix of size 10x10. Give the output values upto 6 decimal points.