Will Schell

Crypto problem sets

The Creation of the Program/Code

The code supplied to us by Professor Graci gave us the tools needed to write and complete the code to run properly. When given the code, we were able to produce crypto problems of length 2, 3, and 4. This left us to write the code for the crypto of length 5. Once we finished creating the length 5 crypto we had to make a solver for this. We coded the solver but we wanted to have all the code in one file. We combined the crypto problem maker, v3 of the crypto sets, and we combined it with the crypto solver we coded. Then we had to load previous files for, global variables and our combosets file. Once these were added we then added some code to combine the results of making a crypto and solving the crypto. We put these into a convenient demo method that allows users to input a number which will create N number of cryptos and their solutions, or they can enter their own crypto manually.

Running the Program

In this we had to add the functionality to allow these 2 processes to run together. We used the following code to make this happen.

establishProblem :: this is used in order to get a crypto of length 5 and a goal for it for a total of 6 parameters

getProblemFromKB :: this function passes a problem information to the crypto solver.

addCryptoSolutionToKB :: this removes the solution stored in the KB, if one exists, then asserts the new one into the KB. If we have no solution there will be a message to be displayed saying there is no solution to the problem.

The previous two functions are used within:

solveDecompositionally :: this takes a problem that is created from the KB using the getProblemFromKB.

displaySolution :: this will display a single solution that is stored in the KB by using displayResult.

displayResult :: this will allow a user to see a single solution, iterate through all the possible solutions, or display a message saying that there are no solutions to be displayed.

solve :: this take a crypto problem as numbers(...) and goal(G). This will instantiate a new KB problem using establishProblem from above, and will display the problem for you to see.

solve :: will also take 'random' as a parameter. This creates a single random crypto problem and solves it.

demo :: this function allows a user to input an integer as a parameter. What this does is solves that many N number of random crypto problems. It displays the problem and its solution.