Problem Definition

# Client Problem

Your client is looking to protect their files. They want an application which will accept 1 or more files, encrypt them using a couple of different methods (no password, password, reversable), have the option of splitting them into parts and be able to decrypt into their original files.

## Breakdown of Client Problem

* Protect their files
* Accept 1 or more files
* Encrypt the files
* Different encryption methods
  + No password
  + Password
  + Reversible
* Option of splitting files into parts when encrypting
* Be able to decrypt into their original files

# Needs

* Easy to use
* Encrypt Files
* Decrypt files
* Encrypt multiple files at a time
* **Encrypt file(s) with multiple methods easily [?]**
* Ability to compress files
* Different encryption methods
  + Custom Encryption Algorithm
  + No password
  + Reversible

# Objectives

* Easy to use
  + The GUI should be easy and fast to navigate
  + Tooltips to explain the function of buttons if not obvious
  + Easy to access instructions/user manual
* Encrypt Files
  + Read bytes of the file
  + Apply encryption algorithm
  + Add file identifiers to encrypted container
    - File name
    - Part Number
    - File hash to check if data is valid
  + Option of splitting the encrypted file into parts
  + Write the encrypted data to disk
* Decrypt Files
  + Detect which encryption algorithm was used
  + Decrypt multiple files at a time
  + Identify if the key provided is correct
  + Identify if there are multiple parts to decrypt
  + Read file bytes into memory
  + Apply Decryption algorithm
  + Write decrypted data to disk
  + Give the decrypted file its original name that is contained in the encrypted container
* Encrypt multiple files at a time
  + Encrypt all selected files
  + Encrypt directories
  + Give user the same options as with encrypting a single file
* Ability to compress files
  + Option to compress files before encryption to save disk space
  + Uses 7z for better compression ratio
* Different encryption algorithms
  + Custom Encryption Algorithm
    - Encrypted data should appear to be random
    - Requires a key to encrypt and decrypt
    - Option to use another file as the key
  + No Password
    - Data will be encrypted with a key generated by the program
    - Method of storing the key
    - Can be used for any encryption algorithm [?]
  + Reversible
    - File bytes will be reversed [?]

# Boundaries

* Hardware
  + RAM needs to be taken into account when reading data into memory
  + Speed is limited by hardware
* Software
  + Ability to manipulate bytes
* Knowledge and Experience
  + Very little prior experience with working with bytes
  + No experience with encryption algorithms
* Ability to create an efficient and creative GUI

# Process

## User wants to encrypt a single file (Not yet in the File List)

1. [Add File]
2. User chooses a single file from the file browser dialogue
3. Encryption Options are displayed:

* Method
* Key
* Split into parts?
* Compress before encrypting?
* Overwrite original file?
  + If the user does not want to remove the original file they will be given the option to select a path for the encrypted file

1. [Confirm]
2. File is added to the File List
3. File path is stored in an encrypted file. All files in the File List will be stored in it.
4. File is encrypted with the specified options
5. The encrypted file is now displayed in the File List.

## User wants to decrypt a single file (Already added to File List)

1. User selects the encrypted file from the list
2. [Decrypt]
3. Encryption method is detected
   1. No Password
      1. The file is decrypted
   2. Reversible
      1. File bytes are reversed
   3. Password
      1. User is asked for the password or key file
      2. Attempt to decrypt the file and alert the user if the key is incorrect
4. File list is updated with the new file status

## User wants to encrypt a single file (Already added to File List)

1. User selects the decrypted file in the File List
2. [Encrypt]
3. Encryption Options are displayed:

* Method
* Key
* Split into parts?
* Compress before encrypting?
* Overwrite original file?
  + If the user does not want to remove the original file they will be given the option to select a path for the encrypted file

1. [Confirm]
2. File is encrypted