## Problem Definition

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.

Due to time constraints or security reasons some features were not implemented. The program only accepts one file at a time but if multiple files need to be encrypted, the user can zip the files and encrypt that. There are multiple encryption/decryption methods as well as a safety feature that won’t allow the user to decrypt the file with a wrong key as it would result in corrupt data. This safety measure could be considered a security flaw but if the length key used to encrypt the file is longer than 1 then it shouldn’t be a problem. Particular features like file splitting and compression were either later removed or only prototyped as they were not seen as crucial to the final program. Better time management and prior experience of working with bytes would’ve made the development process easier and yielded a more complete and polished program.

## Needs

* **Easy to use** - Implemented.
* **Encrypt Files** - Implemented.
* **Decrypt files** - Implemented.
* **Encrypt multiple files at a time.** - Not implemented   
  [ time constraints. ]
* **Ability to compress files** - Not implemented  
   [ All compression features were not fully implemented and was disabled at time of release. Furthermore It would be better if the user compressed their own files and had more control over it. We focused on the encryption, as it is the purpose of our program. ]
* Different encryption methods
  + **Custom Encryption Algorithm** - Implemented.
  + **No password** - Not implemented  
    [ Not secure. Time constraints ]
  + **Reversible** - Implemented.

## Objectives

* Easy to use
  + **The GUI should be easy and fast to navigate** - Implemented.
  + **Tooltips to explain the function of buttons if not obvious** - Not implemented  
    [ time constraints. ]
  + **Easy to access instructions/user manual** - Implemented.
* Encrypt Files
  + **Option of splitting the encrypted file into parts** - Prototyped but not implemented.  
    [ The documented method allowed for checking the integrity of each file part, but would have taken too much time to implement in the timeframe. The other option was to use a simple method, but was not prioritised in the last moments, and was not completed. ]
* Decrypt Files
  + **Detect which encryption algorithm was used** - Not implemented  
    [ This would’ve been too time intensive and reduced the security so it was not included in the final release ]
  + **Decrypt multiple files at a time** - Not implemented  
    [ Each file would have required a different key, this would have made things too complex with the method we used to store the files in the list. ]
  + **Identify if the key provided is correct** - Implemented.
  + **Identify if there are multiple parts to decrypt** - Not implemented  
    [ Not included as file splitting hadn’t been implemented only prototyped towards the end of development so there was no need to join files. ]
  + **Give the decrypted file its original name that is contained in the encrypted container.** - Not Implemented  
    [ Encrypting the name does not provide much security, unless the whole directory name is also encrypted. The file name was never encrypted and this feature was removed. ]
  + **Encrypt multiple files at a time** - Not implemented.  
    [ Time constraints. ]
  + **Encrypt all selected files.** - Not implemented  
    [ Our method of storting did not allow for his feature, it would have been too complex to implement. ]
  + **Encrypt directories.** - Not implemented  
    [ Was not seen as essential. Original plan was to compress the directory and then encrypt it. Compression was not implemented.]
* Different encryption algorithms
  + *Custom Encryption Algorithm*
    - **Encrypted data should appear to be random** - Implemented.
    - **Requires a key to encrypt and decrypt** - Implemented.
    - **Option to use another file as the key** - Not implemented.   
      [ Time constraints ]
  + *Reverse*
    - Data will be encrypted simply by reversing the order of the bytes from the given file. - Implemented.
    - **encrypted data will be written to disk.** - Implemented.