**Deliverable 5: Implementation of a secure messaging system**

Our objective is to implement Model 1 from deliverable 4. You have already implemented DES (deliverable 1), RSA (deliverable 2), and a Hashing function (deliverable 3). To ensure everyone is using the same codebase here, we will not use those components to implement Model 1 (though we could have done so!). Instead, we will use a real-world library, PyCryptodome to implement this model.

To complete this deliverable, you must install the PyCryptodome library on your computer. Please see this link (<https://pycryptodome.readthedocs.io/en/latest/src/introduction.html>) for instructions on how to install it.

Look at the sample python file, **library\_code\_examples.py**, that gives short code snippets of how to use DES, RSA and a hashing function from this library. My examples use some additional options for DES (the OFB mode) and RSA (padding done with OAEP). Please use exactly those options. Now, implement Model 1 using the code snippets as a starting point. Note that I have referenced this library as Crypto in my imports. Depending on which option you chose when you installed it, you may have to reference the library as Crypto or Cryptodome.

You are provided with public and private keys for A and B (in PEM file format), and a 64-bit secret key file for DES stored as an ASCII string and which is always 8 characters in length.

Your encryption implementation method(s) should read the plaintext message string to be encrypted (stored in ASCII format) from an input file called ‘plaintext.txt’, read all necessary keys, perform the encryption as described in the model, and finally write the three pieces of the ciphertext, C1, C2 and C3 *to three separate files*.

Your decryption implementation method(s) should read the three pieces of the ciphertext, C1, C2 and C3, from their files, read all necessary keys, perform the decryption as described in the model, and then write the decrypted plaintext message string to a new output file called ‘plaintext2.txt’. If the ciphertext cannot be verified (i.e. it is repudiable), then do not write the decrypted message but simply print a “REJECTED” error message to the standard output.

**Deliverable**

Submit your code files and a README file. Your code file must have clear instructions on how to run it. The README file must provide your self-assessment on whether you successfully completed this deliverable.