Criptografia RSA

A segunda parte da avaliação deverá ser construído um módulo de criptografia RSA que será postado no github, com data final de entrega em 12 de abril, também deverá ser feito um relatório em LaTeX explicando o método usado para construir o módulo de criptografia e um módulo cliente que irá usar o módulo de criptografia, além da documentação do código, a classe RSA deverá ter no mínimo as seguintes especificações:

1. Função setup, recebe opcionalmente as chaves pública e privada caso não sejam fornecidas são geradas chaves aleatórias (seguindo as restrições do algoritmo, sugere-se números pequenos), a chave pública deve ser retornada ao usuário em uma tupla .
2. Uma função encrypt, que recebe uma string consistindo na mensagem a ser criptografada, junto com um tupla representando a chave pública, e devolve uma string a mensagem criptografada.
3. Uma função decrypt, que recebe uma string consistindo na mensagem a ser descriptografada e devolve uma string a mensagem descriptografada.

A classe Cliente deverá efetuar a adição, deleção e verificação de existência dos usuários, quando o programa for encerrado devem ser geradas chaves aleatórias no RSA os usuários devem ser criptografados e salvos juntos com as chaves no formato:
(chave privada) (chave pública)
usuario1
senha1
.
.
.
usuarioN
senhaN

quando o programa iniciar o cliente ler, decriptografar e carregar os usuários já existentes no programa.

As mensagens de entrada do algoritmo sempre serão compostas por caracteres do alfabeto inglês em letras minúsculas e números, o código deverá incluir tratamento de erros (por exemplo o uma tentativa de passar um texto inválido como entrada para a criptografia).

O esquema para a conversão de caracteres em inteiro(e vice versa) seguirá o padrão UTF-8. Esse padrão pode converter inteiros menores que 1 milhão em caracteres(por isso o incentivo a números pequenos para a formação das chaves), por padrão as funções ord e chr do python já usam UTF-8.