IFT3275 Devoir 1

Riste Popov [20241764]
Richard Lao [20278343]

1.1 Pour cette question on s'est basé sur le fait que si le message est assez petit, l'application de l'exposant e sur le message donnera quelque chose de plus petit que la valeur de N. Alors, en faisant m^e mod N le mod N retournera m^e. En effet, comme e est 3 et le message M est petit, il suffit simplement d'appliquer la racine cubique sur le cryptogramme pour obtenir le message. On avait raison puisque en faisant la racine cubique on a obtenu le message "Umberto Eco" qui est une personne célèbre.

1.2 Pour cette question on a eu de la difficulté à trouver une façon de décrypter le cryptogramme. La même technique que le 1.1 ne marche pas parce que le e est trop grand. Aussi, factoriser le N est trop difficile à faire comme le N est trop long. Alors, on a décidé d'essayer d' utiliser une faiblesse de RSA textbook ou le le cryptogramme pour un message est toujours le même. On a pris une liste de noms d'auteurs et on les a cryptés en utilisant la clé publique et on a comparé avec le cryptogramme. On a utilisé cette source pour avoir 100 auteurs connus: https://jimmymorneau.blogspot.com/2014/09/top-100-ecrivains.html . En cryptant le message "Marcel Proust" on a obtenu le même cryptogramme que celui qu'on devait décrypter. Alors, on peut confirmer que Marcel Proust est l'auteur célèbre qu'on cherchait.

2. Pour cette question, on a utilisé le fait que le texte crypté est un extrait des deux corpus combinés. Comme on connaît aussi le dictionnaire et on sait que chiffrement par substitution remplace toujours un caractère ou bicaractère par une même chaîne de 8-bit, on peut trouver un pattern des caractères qui se répète et comparer avec les deux corpus pour trouver la position du texte crypté. On doit aussi crypter les deux corpus à chaque position du texte, convertir en pattern et comparer avec le pattern du texte crypté. Aussi, on ne connaît pas la taille du message du texte crypté donc elle peut être de n(taille du texte crypté) à 2n. On essaye toutes les possibilités jusqu'à trouver un match avec le cryptogramme. Cela cause une complexité assez grande. Alors, notre code est quand même lent pour des grands textes mais marche mieux pour des petits textes. Pour cette raison, on a décidé de rajouter une condition qui va tester le texte différemment dépendant de la longueur. Si le cryptogramme est plus grand que 6000 on utilise une technique différente qui consiste à prendre le corpus et trouver les symboles les plus fréquent et les associer avec les 8 bits les plus fréquent dans le cryptogramme. Ensuite, on les replace dans le cryptogramme.

Github:
https://github.com/IFT3275-Securite-Informatique/devoir-1-cryptographie-riste_richard