Author 1: Mukund Manikarnike (1208597425)

Author 2: Lakshmi Srinivas (1208635554)

Algorithm

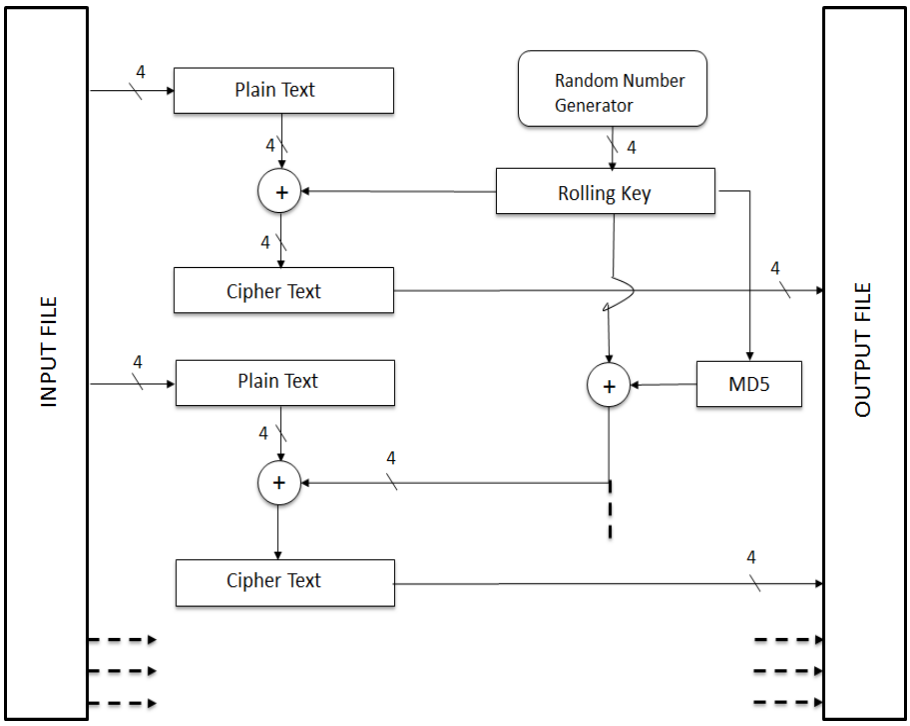
This section answers question 1 of the Homework.

The summary of the algorithm is that it generates a random key and performs an XOR operation of this key with the plain text to obtain the cipher text, the details of which are explained below.

Description

1. The first 4 byte random key is generated by using the random number generation function – dev/urandom. This random number called the rolling key is further passed through an md5 function in order to compute subsequent rolling keys. An XOR operation of the output of md5 function and the previous key gives the next 4 byte rolling key.
2. The input is read 4 bytes at a time from the input file and XORed with the rolling key to obtain the cipher text. The combined output is then stored onto an output file.
3. Given that the XOR operation is reversible, the decryption algorithm, operates exactly in the reverse fashion.

The figure below illustrates how the algorithm operates



Analysis

This. Section answers questions 1 and 3 of the Homework. Following were the few weak points of the algorithm.

- The algorithm XORs only the first 4 bytes of the input file with they provided key. XOR being a reversible operation, this algorithm becomes easily reversible if the file format is known.

- A typical property of a good encryption algorithm would be to perform confusion and diffusion which this algorithm doesn't do.
- It does derive sub-keys from an input key. However, these keys aren't used to permute the input plain-text at different rounds of encryption. But, they are used to encrypt the next plain text which establishes a very linear relationship between the plaintext and the cipher-text.
- The algorithm also doesn't do any integrity checks and hence there is no way to know whether decryption was successful or not unless the file format is known.

Brute Force Attack

This section answers questions 2 and 4 of the homework.

The brute force attack was attempted by going through all keys from 0 – 0xFFFFFFFF and using the decryption algorithm. Since the brute force attack was taking incessantly long time, the keys were figured out faster using the weaknesses of the algorithm mentioned above. The brute force attack was performed as follows in each of the following cases of input files.

File Type	Description	Key Obtained
Text File	A Text file would contain only the ASCII character set (0x00 – 0x7F). By using this property and the weakness of the algorithm described in the previous section we can optimize the attack in the following way 1. Read the first word of the file. In the given text file, it is 0xB8B354C2. 2. In order to have a plain text which is within 0x7F bits 8, 16, 24 and 32 of the plain text have to be 0 3. This cipher text indicates that bits 8, 16, 24 and 32 need to be set to 1, 0, 1 and 1 respectively to obtain a plain text that satisfies the desired conditions. This reduces the key set to be tried for a brute force decryption to 2^28 from 2^32. The attack needs to be performed by attempting decryption on the encrypted file using all keys in the reduced key set. The algorithm was implemented as mentioned above. However, the key wasn't obtained at the time that the report was made because the attack was still in progress.	-
PNG File	The first 4 bytes of the PNG file format contain 0x89, 0x50, 0x4E and 0x47. Given that the encryption algorithm XORs they input key only with the first 4 bytes of the input file, we have the cipher text and the plain text for the first 4 bytes and hence performing an XOR operation between the two would give the key. This key can later be used to decrypt the encrypted file. The key obtained by using this method is mentioned. The decryption attempt was successful. It found the picture of a cat.	0xc9034bf4
PDF	The first 4 bytes of the PDF format contain 0x25, 0x50, 0x44 and 0x46. Given that the encryption algorithm XORs they input key only with the first 4 bytes of the input file, we have the cipher text and the plain text for the first 4 bytes and hence performing an XOR operation between the two would give the key. This key can later be used to decrypt the encrypted file. The key obtained by using this method is mentioned. The decryption attempt was successful. It found a PDF with material about BitCoin.	0x1739d398