First of all we generated 1 lakh random strings using generate_random.cpp. Then we generate corresponding strings having XOR equal to the characteristic value using file gen_pairs.cpp.

```
$ g++ generate_random.cpp -o generate_random
$ ./generate_random

$ g++ gen_pairs.cpp -o gen_pairs
$ ./gen_pairs
```

After getting these pairs having XOR equal to particular characteristic value, we used file generate.exp to get corresponding cipher pairs from the server. For this expect scripting was used.

```
$ sudo apt-install expect
$ sudo chmod 777 generate.exp
$./generate.exp
```

This script will generate the output.log file having all the ciphertexts with some extra output. Then we used clean.py to clean this output. Then we generate a file named output_binary.txt having binary values corresponding to ciphertexts. Then compute_xor.cpp was used to generate three files: xor_s_out.txt, xor_exp_out.txt, alpha1.txt. After this s_box.cpp was used to generate key.txt which gives the frequencies of the keys.

Then we derived partial key (36 bits) manually using the round6_key_positions.txt file generated from generate_round6_key_positions.cpp. Then we generated remaining 20 bits using generate_keys.py. Then we checked each of this key using DES_bruteforce.cpp and got the original key. After getting the key, we used DES_decryption.cpp to decrypt the password. To automate this process after getting the output.log file, a run.sh script is present.

```
$ sudo chmod +x run.sh
$ ./run.sh
```