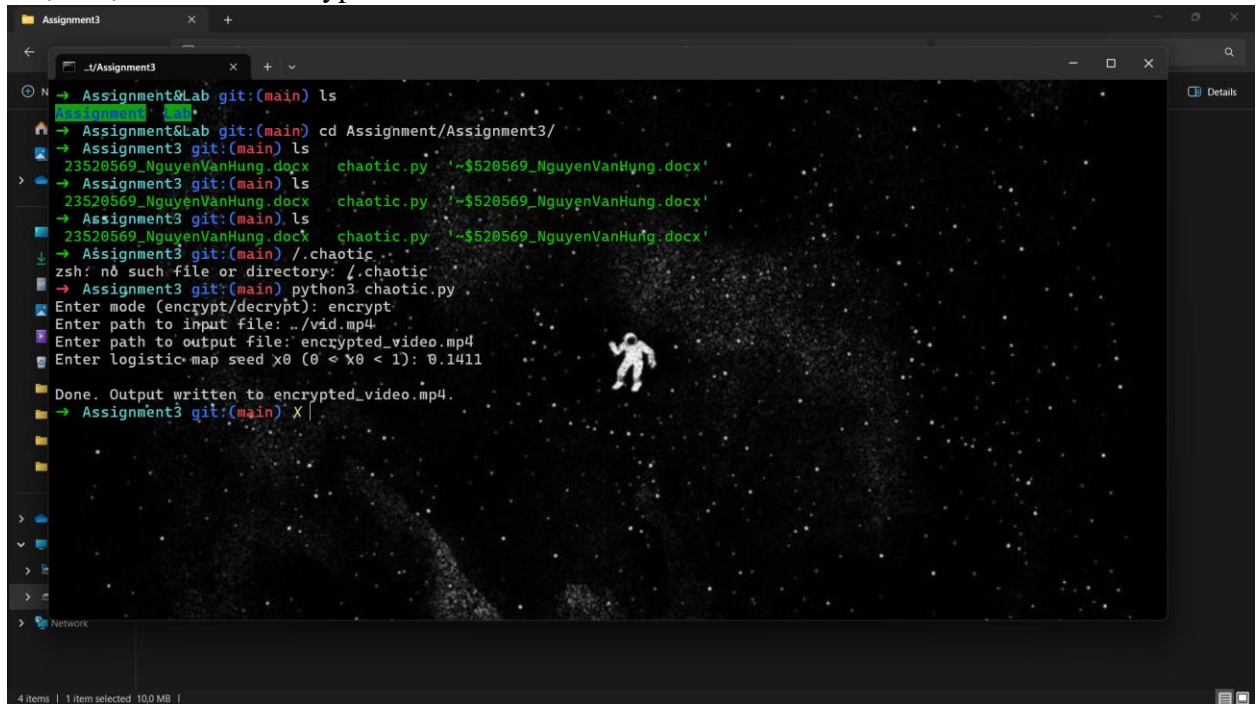


# Report Week 3

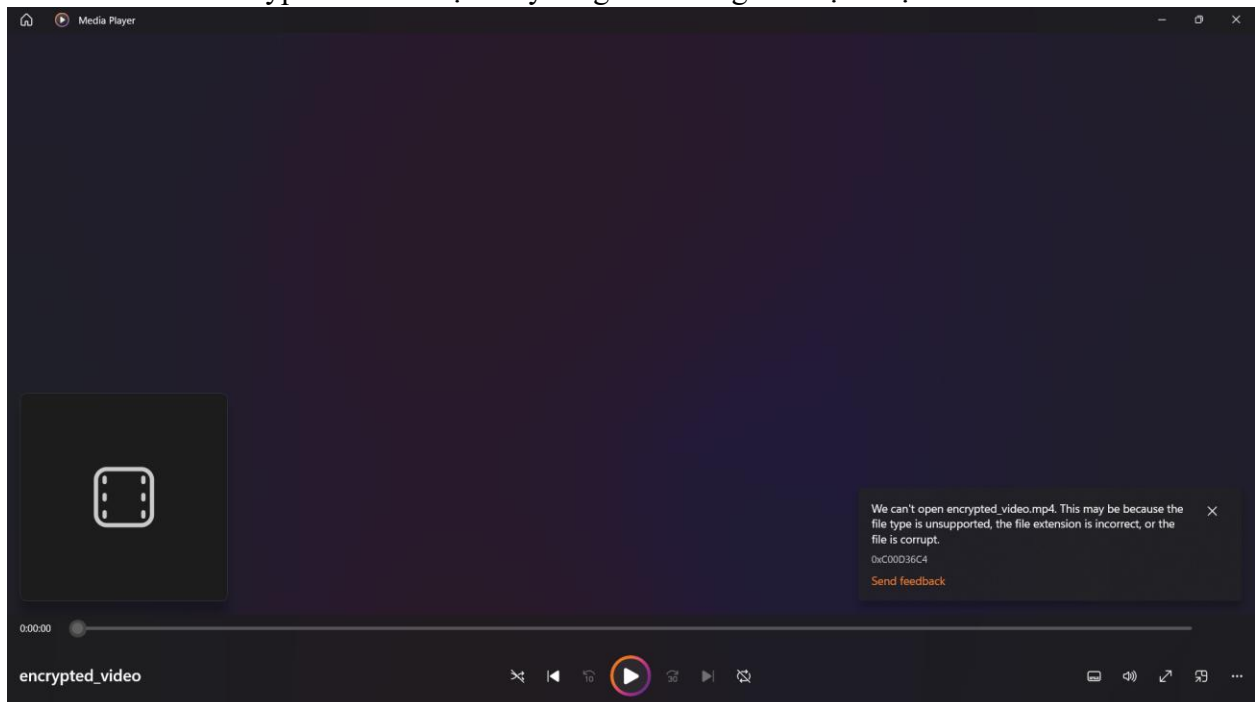
## 1. Code chaotic-based stream cipher

- Sử dụng dyadic transformation làm logistic map.
- Code: [code](#)
- Thực hiện kiểm thử encrypt:

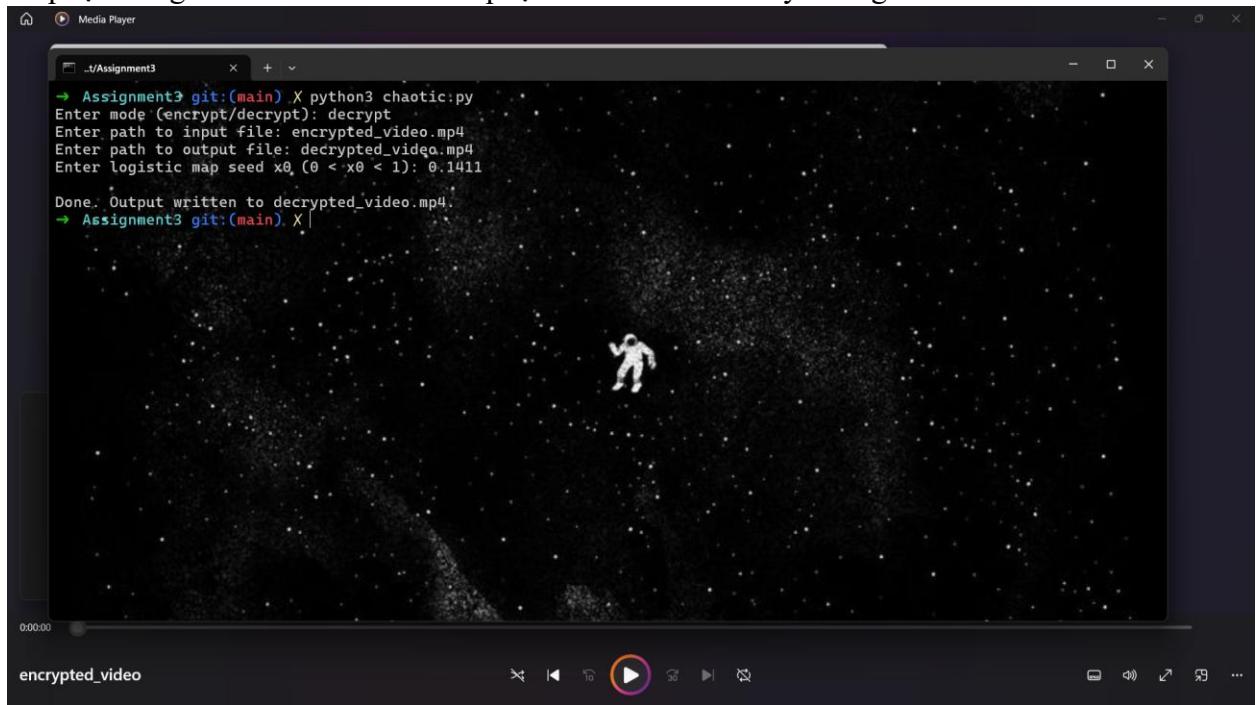


```
Assignment3
└─ .t/Assignment3
   └─ Assignment3
      └─ git:(main) ls
         23520569_NguyenVanHung.docx  chaotic.py  '~$520569_NguyenVanHung.docx'
      └─ git:(main) cd Assignment/Assignment3/
         23520569_NguyenVanHung.docx  chaotic.py  '~$520569_NguyenVanHung.docx'
      └─ git:(main) ls
         23520569_NguyenVanHung.docx  chaotic.py  '~$520569_NguyenVanHung.docx'
      └─ git:(main) ./chaotic
         zsh: no such file or directory: ./chaotic
      └─ git:(main) python3 chaotic.py
         Enter mode (encrypt/decrypt): encrypt
         Enter path to input file: ./vid.mp4
         Enter path to output file: encrypted_video.mp4
         Enter logistic map seed x0 (0 < x0 < 1): 0.1411
         Done. Output written to encrypted_video.mp4.
      └─ git:(main) X
```

- Mở file sau khi encrypted ra thì nhận thấy rằng file không còn đọc được nữa:



- Tiếp tục thử giải mã xem có thể khôi phục về file ban đầu hay không:



- Mở file sau khi decrypt ra thì thấy rằng file đã được khôi phục:



## 2. Cryptanalysis Stream Cipher

- Stream Cipher sẽ không an toàn nếu sử dụng một key để mã hóa nhiều file vì khi chúng ta mã hóa 2 file M1 và M2 bằng một key thì có được hai file E1, E2. Chỉ cần lấy E1 xor E2 thì chúng ta đã có được giá trị của M1 xor M2.
  - Nếu attacker có được M1 hoặc M2 thì có thể dễ dàng biết được file còn lại.
  - Tương tự, nếu có nhiều cặp M1 và M2 thì attacker cũng có thể dựa vào chúng để tăng độ chính xác và giảm được nhiều thời gian trong việc tìm ra key.