



---

# 18I-0453

---

Assignment 1



[DATE]  
MICROSOFT  
[Company address]

## Contents

Oligomorphic Virus .....	3
Introduction: .....	3
Code Flow: .....	4
Hashing and Signaturing: .....	4

## Table of Figures:

Oligomorphic Flow .....	3
Output of encryptors/ Decryptors .....	4
Output of Hashing/Signaturing .....	5

## Oligomorphic Virus

### Introduction:

So as I had the option to depict any one out of 3 Mentioned avoid detection techniques

So I am going to depict Oligomorphic Viruses

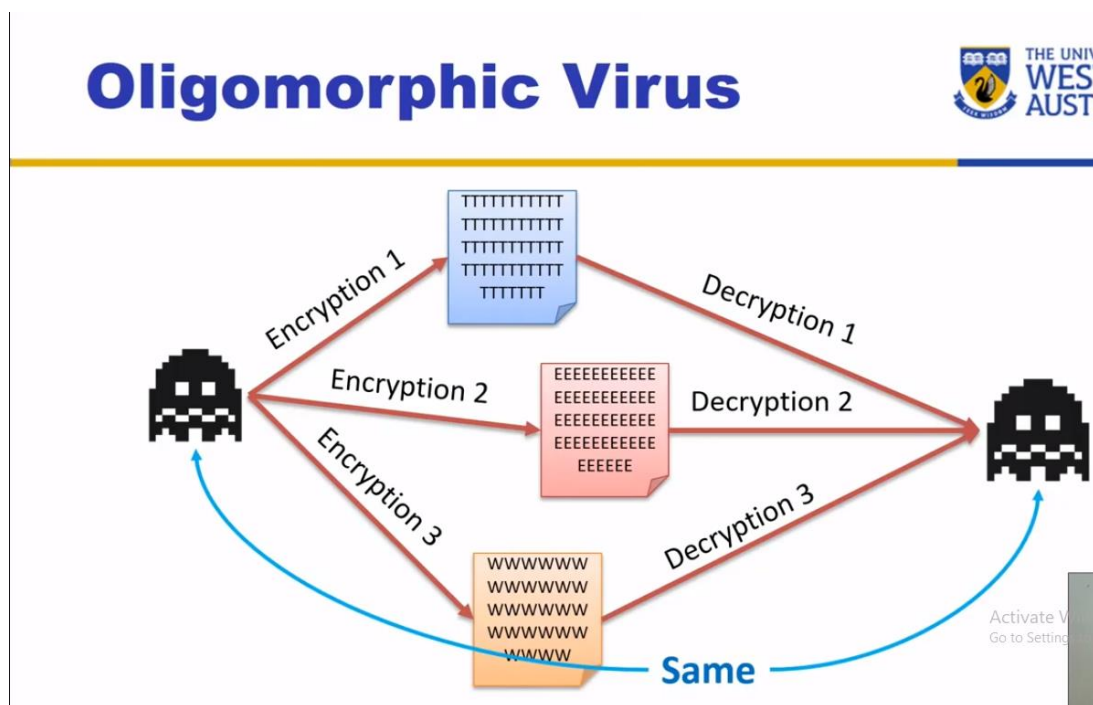


Figure 1

So this is the best diagram I have found to understand oligomorphic virus.

Oligomorphic is a technique to encrypt the virus code so that antivirus system cannot detect the virus code.

So what oligomorphic does that instead of using 1 encryptor/Decryptor, it uses n number of Encryptor and Decryptors. So that it becomes more difficult for an anti-virus to detect the virus code as the Encryption will be unique every time

### Code Flow:

So I have implemented my code on the basis of above diagram

And I have used Ciser Cypher as 1<sup>st</sup> encryptor Decryptor, Rot-13 as 2<sup>nd</sup> encryptor Decryptor, Transposition Cypher as 3<sup>rd</sup> encryptor. And program decides on runtime based on a random number which encryption Decryption to use.

The algorithms of all 3 encryptors decryptors are explained in code

And the result of all 3 Encryptors and decryptos is attached below

```
In [7]: 1 encrypt=CiserCipher.encrypt(virus(),3)#Putting shift of ciser cypher to 3 always
2 print("Encryption of ciser Cypher is ",encrypt)
3 decrypt=CiserCipher.decrypt(encrypt,3)
4 print("Decryption of Ciser Cypher is ",decrypt)
```

```
Encryption of ciser Cypher is  KoorZruog
Decryption of Ciser Cypher is  HelloWorld
```

```
In [8]: 1 encrypt=Rot13.encrypt(virus(),3)
2 print("Encryption of Rot-13 is ",encrypt)
3 decrypt=Rot13.decrypt(encrypt,3)
4 print("Decryption of Rot13 is ",decrypt)
```

```
Encryption of Rot-13 is  koorzruog
Decryption of Rot13 is  helloworld
```

```
In [9]: 1 encrypt=TranspositionCipher.encrypt(4,virus())
2 print("Encryption of TranspositionCypher is ",encrypt)
3 decrypt=TranspositionCipher.decrypt(4,virus())
4 print("Decryption of TranspositionCypher is ",decrypt)
```

```
Encryption of TranspositionCypher is  HoleWdlolr
Decryption of TranspositionCypher is  HelloWorld
```

Figure 2

### Hashing and Signaturing:

I have used a very famous hashing known as SHA-256. To generate hash on the basis of current time. So that hash is always unique. And then I generate the signature based on that hash

```
: 1 print("Hash is ",hash256)
  2 signature = base64.b64encode(hashlib.sha256(hash256).digest())
  3 print("Signature is ",signature)
```

```
Hash is  b'202209090320110764362213f1fd1b-ab2d-4c1f-8e2c-ca61878f2a44'
Signature is  b'yEbza/cI6YE12dewwEJ7Kty6k81YqnWuMa/EwjAvCqg='
```

---

*Figure 3*

The Hash and signature will be unique every time executable (code) gets run.