The Reverse Cipher

David W. Agler

November 18, 2023

Reverse Cipher

One of the simplest ciphers is the reverse cipher. To encrypt a plain text, one simply reverses the order of the letters in the plain text. The result is the cipher text. For example, the message "HAT" is encrypted as "TAH" using the reverse cipher. The message "HELLO WORLD" is encrypted as "DLROW OLLEH" using the reverse cipher.

The method of decrypting a message is exactly the same as the method of encrypting the message: we simply reverse the order. So, for example, the message "TAH" would be decrypted as "HAT". The message "DLROW OLLEH" would be decrypted as "HELLO WORLD".

The strength of the reverse cipher is that it is extremely easy to understand and use. Children sometimes use it to pass "secret messages" to their friends. If the goal of encryption is to give our message a level of security such that a third-party cannot read our message, then the reverse ciphere does not do a very good job as it is an easy cipher to decrypt. That is, the reverse cipher is not a very secure cipher.

Python Implementation

Let's create a simple script that encrypts a plain text message using the reverse cipher. There are many ways to do this. There are several ways. Let's consider two ways to encrypt and decrypt a message using the reverse cipher.

Method 1: using a Loop

First, we'll start by defining a variable called `plaintext` that stores our message. We'll use the message "The government knows our secret." as our plain text message.

```
plaintext = "The government knows our secret."
```

Next, we'll create a function called `reverse_func` that takes a string as an argument and returns the string but in reversed. This is achieved by first

defining a variable called `ciphertext` that is equal to an empty string. We'll then define a variable called `i` that is equal to the length of the plain text message minus 1. This will give us the index of the last letter in the plain text message. We'll then use a `while` loop to iterate through the plain text message. So, we start at the end of the `plain_text` and add a letter to `ciphertext`. Once we do that, we'll decrement `i` by 1. This will take us to the second to last letter in the plaintext. We'll keep repeating the process until i is no longer greater than or equal to 0. We'll then return the `ciphertext` string.

```
plaintext = "The government knows our secret."

def reverse_func(plain_text):
    ciphertext = ""
    i = len(plain_text) - 1
    while i >= 0:
        ciphertext += plain_text[i]
        i -= 1
    return ciphertext

print(reverse_func(plaintext))
```

So, the above will print out the following:

```
.terces ruo swonk tnemnrevog ehT
```

Method 2: Slicing

Let's consider a second way to encrypt the message (although there are many other ways). Again, we'll start by defining a variable called `plaintext` that stores our message. We'll use the message "The government knows our secret." as our plaintext message.

```
plain_text = "The government knows our secret."
```

To encrypt, we'll use slicing. The syntax of slicing is `sequence[start:stop:step]`. If we specify a negative step and the `start` and `stop` take their default values, then the slice starts at the end of the sequence and steps down the sequence until it reaches just past beginning of the sequence. We can store this slice as a variable and print the result.

```
plaintext = "The government knows our secret."
def reverse_func2(plain_text):
    ciphertext = plain_text[::-1]
    return ciphertext
print(reverse_func2(ciphertext))
```

Decrypting the Message in Python

Now that we've seen how to encrypt a message using the reverse cipher, let's consider how to decrypt ciphertext that has been encrypted using the reverse cipher. To decrypt a message, we simply reverse the order of the letters in the cipher text message. So, for example, if the cipher text message is "TAH", then the plain text message is "HAT". If the cipher text message is "DLROW OLLEH", then the plain text message is "HELLO WORLD".

Using Python to decrypt a message is very easy. We can use the same function that we used to encrypt the message. We simply pass the ciphertext into the functions and they will both return plaintext.

```
ciphertext = ".terces ruo swonk tnemnrevog ehT"
print(reverse_func(ciphertext))
print(reverse_func2(ciphertext))
```

Both print statements will return "The government knows our secret."

This concludes the reverse cipher.

.wonk em tel ,snoitcerroc ro sthguoht evah uoy fI .gnidaer rof sknahT

Code Used in this Post

```
plaintext = "The government knows our secret."

# Encrypting with a for loop
def reverse_func(plain_text):
    ciphertext = ""
    i = len(plain_text) - 1
    while i >= 0:
        ciphertext += plain_text[i]
        i -= 1
    return ciphertext

print(reverse_func(plaintext))

# Encrypting with slicing
def reverse_func2(plain_text):
    ciphertext = plain_text[::-1]
    return ciphertext
print(reverse_func2(plaintext))

# Decrypting with slicing
ciphertext = ".terces ruo swonk tnemnrevog ehT"
print(reverse_func(ciphertext))
print(reverse_func2(ciphertext))
```