

Information Security

Base64

Encoding and Decoding

32171550 박다은

2020-9-30

Contents

1. Project Overview	3
(1) Project Introduction	3
(2) Program Goals	3
(3) Program Structure	3
(4) Build Environment	3
2. Code & Consideration	4
(1) 주요 코드	4
(2) Screenshot	5
(3) Personal Feeling.....	6

1. Overview

(1) Introduction

Base64 는 바이너리 데이터를 문자 코드에 영향을 받지 않는 공통 ASCII 문자로 표현하기 위해 만들어진 인코딩이다. 8 비트 이진 데이터를 64 개의 문자를 이용하여 아스키 영역의 일련의 문자열로 바꾸는 인코딩 방식을 말한다.

구체적으로 A-Z, a-z, 0-9 까지 62 개의 문자와 +, /의 2 가지 기호가 사용된다. 변환 단계는 우선 8 비트짜리 바이트 3 개를 6 비트씩 4 개로 쪼개어 Base64 코드 4 개로 바꾸어 표현한다. 이 과정을 반대로 구현하면 디코딩이 된다.

(2) Goals

이메일, 전자게시판 등에서 사용되는 Base64 를 직접 구현해보며 이해하고 분석한다.

(3) Program Structure

Open_file

```
+-----  
|      encoding()  
|      decoding()  
+-----
```

Close_file

각 단계에서 각자의 역할은 결과를 출력하는 것이다.

Encoding 과 decoding 단계에서 각각 새로운 텍스트 파일에 output 을 쓰게 된다.

(4) Build Environment

Compilation: Python 3

2.Code & Consideration

(1) 주요 코드

- encoding 과 decoding 을 하는 함수를 각각 따로 만들어주었다.

```

1 def encoding(s):
2     value = []; i = 0
3     while i < len(s):
4         value.append((ord(s[i]) >> 2) & 0x3f)
5         value.append(((ord(s[i]) & 0x3) << 4) | (ord(s[i+1]) & 0xf0) >> 4)
6         value.append(((ord(s[i+1]) & 0xf) << 2) | (ord(s[i+2]) >> 6) & 0x3)
7         value.append(ord(s[i+2]) & 0x3f)
8         i = i + 3
9
10    out = ""
11    for j in value:
12        out = out + base64[j]
13    print("Encoding: ",out)
14
15    return out

```

- value 라는 리스트를 만들어서 그 안에 인코딩할 symbol 의 index 를 넣도록 했다. Shift 연산자와 AND, OR 을 사용하여 차례대로 인코딩해주었다. 이 과정에서 문자를 아스키코드로 변환해주는 ord 함수를 사용하였다. 또 ASCII 3 개가 Base64 4 개로 바뀌므로 i 에 3 을 더해주었으며, while 문 한바퀴를 돌 때마다 value 에 4 개의 값이 추가된다.
- 그리고 for 문을 통해 value 에 있는 index 들을 symbol 로 바꾸어주었다.

```

1 def decoding(s):
2     value = []; i = 0
3     while i < len(s):
4         value.append((base64.find(s[i]) << 2) | ((base64.find(s[i+1]) & 0x30) >> 4))
5         value.append(((base64.find(s[i+1]) & 0xf) << 4) | ((base64.find(s[i+2]) & 0x3c) >> 2))
6         value.append(((base64.find(s[i+2]) & 0x3) << 6) | base64.find(s[i+3]))
7         i = i + 4
8
9     out = ""
10    for j in value:
11        out = out + chr(j)
12    print("Decoding: ",out)
13
14    return out

```

- Decoding 도 비슷한 방법으로 구현하였다. shift 연산자와 AND, OR 을 통해 4 개의 base64 문자를 3 개의 ASCII 로 바꾸었다. Base64 에서 아스키로 바꾸는 과정이므로, ord 함수 대신 base64 문자열에서 해당하는 문자의 위치를 반환하는 함수인 find 를

사용하였다. 또한 i 에 4 를 더하도록 했으며, while 문 한바퀴를 돌 때마다 value 에 3 개의 값이 추가된다.

- Encoding 과 같은 이유로 for 문을 사용하였다. 아스키코드를 이용해 문자로 변환하는 함수인 `chr` 을 사용하였다.

```

1 #input 파일 열기
2 in_f = open('input.txt', 'r')
3 in_c = in_f.readline()
4 print("Original Text: ", in_c, "\n")
5
6 #encoding
7 en_f = open('en_output.txt', 'w')
8 en_f.write(encoding(in_c))
9
10 #decoding
11 de_f = open('de_output.txt', 'w')
12 de_f.write(decoding(in_c))
13
14 in_f.close()
15 en_f.close()
16 de_f.close()

```

- Input.txt 파일의 original text 를 인코딩해 en_output.txt 에 쓰고, 디코딩해 de_output.txt 에 쓰도록 하였다.

(2) Screenshots

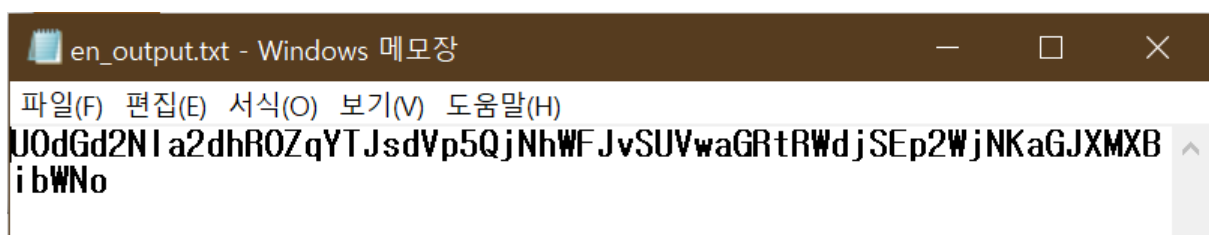
① 출력화면

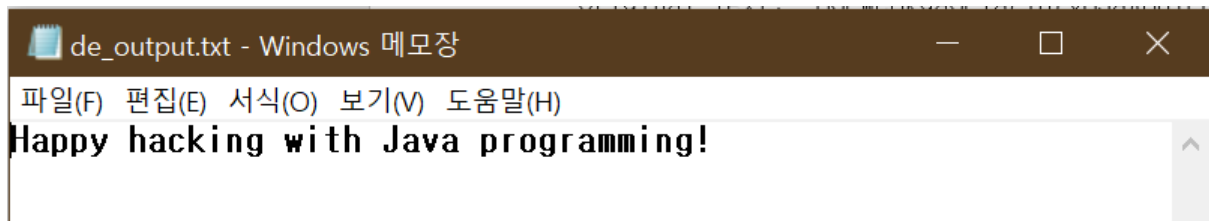
Original Text: SGFwcHkgaGFja2luZyB3aXRoIEphdmEgcHJvZ3JhbW1pbmch

Encoding: U0dGd2Nla2dhR0ZqYTJsdVp5QjNhWFJvSUVwaGRtRWdjSEp2WjNKaGJXMXBibWNo

Decoding: Happy hacking with Java programming!

② output.txt





다음과 같은 두개의 아웃풋 텍스트파일이 생성된다.

(3) Personal Feeling

수업시간에 과제 설명을 제대로 듣고 구현을 시작해서 그런지 큰 어려움이 없었다. 하지만 처음부터 함수를 만든 게 아니라 코드가 굉장히 지저분했었다. 함수를 사용하면 훨씬 간결해지고 편리하지만, 코드를 짤 때는 그 사실을 인지하지 못하는 것 같다. 처음 코드를 짤 때부터 큰그림을 그리고 함수를 유용하게 사용하는 것을 연습해야겠다.

또한 이번 과제는 정보보호개론의 실질적인 첫 과제였다. 그 동안은 강의만 듣느라, 이해하고 있는 게 맞는건지, 잘하고 있는건지 헷갈렸었는데 과제를 끝 마치고 나니 더 흥미가 생기고 이해가 잘 되는 것 같다. 이론으로만 배울 때보다 이렇게 직접 구현해보며 원리를 이해하는 게 훨씬 이해도도 높고 재미도 있다.

앞으로도 과제가 몇 개 남아있는 것으로 알고 있는데, 다음 과제들도 충분한 이해 후에 과제를 시작해서 어려움 없이 끝내고 싶다.