Keyception

Ciphertext: ALWWE RVX ALK NVIFE BUQWS A

Hint 1: In a large (7695 characters) plaintext encrypted with the method used above, the resulting ciphertext had the following frequencies:


Hint 2: The last letter should not be made (directly) back into plaintext.
Solution:

Our cipher uses the plaintext, along with one 'dummy' character as the key for a variation on a shift cipher. For any key in the plaintext, the shift is determined by the next character in the plaintext - thus any character followed by an 'a' would be the same letter in the ciphertext, any followed by a 'b' would be the next sequential character, and so on. The final character is encrypted using the 'dummy' character, which is included at the end of the ciphertext to allow decryption. To decrypt, one must work backwards, starting from the end with the 'dummy' character and using each plaintext character obtained to get the plaintext character that comes before it. The frequencies we gave as a hint should show that it is not a one-to-one shift or substitution cipher, leading away from those approaches. For the given ciphertext, the last letter is the 'dummy', and thus should not be decoded, but used as a key for the second-last letter. Thus, the last plaintext letter is 'S' - 'A' = 's' (18-0=18). The second-last plaintext letter is 'W' - 'S' = 'e' (22-18=4). The plaintext, once decrypted, results to

“these are the ghoul times”
(a pun on “these are the good old times”)

Below is Python code to encrypt/decrypt using ‘the next letter as key’ method:

```python
def encode(plaintext):
    length = len(plaintext)
    ciphertext = "
    i=0
    plaintext+="a"
    while i < length:
        ciphertext += chr((ord(plaintext[i])+ord(plaintext[i+1])-194)%26 + 65)
        i+=1
    ciphertext += 'A'
    return ciphertext

def decode(ciphertext):
    length = len(ciphertext)
    ciphertext = ciphertext.lower()
    plaintext = ciphertext[length-1]
    i = length - 2
    while i > -1:
        plaintext = chr((ord(ciphertext[i])-ord(plaintext[0]))%26 + 97) + plaintext
        i -= 1
    return plaintext[:len(plaintext)]
```