Demonstration of the MAC Forgery Attack

a. Intercept a valid (message, MAC) pair

We assume the attacker has access to a valid message and its corresponding MAC (as would be typical when observing network traffic).

From the insecure server output:

Original message: amount=100&to=alice
MAC: 614d28d808af46d3702fe35fae67267c

This MAC was computed using the insecure formula:

MAC = MD5(secret || message)

b. Perform a Length Extension Attack to append new data to the message

Using the intercepted MAC and original message, the attacker attempts to append malicious data &admin=true **without knowing the secret key**.

We used the Python library hashpumpy, which exploits the MD5 structure to perform length extension attacks.

client.py

C: > Users > MissS > Downloads > mac attack > client.py > ...

```python
import hashpumpy
from server import verify

original_message = b"amount=100&to=alice"
original_mac = "614d28d808af46d3702fe35fae67267c"
data_to_append = b"&admin=true"

print("=== Trying key lengths ===")
for key_len in range(10, 26):
    new_mac, new_message = hashpumpy.hashpump(
        original_mac,
        original_message.decode(),
        data_to_append.decode(),
        key_len
    )

    if verify(new_message, new_mac):
        print(f"\n>>> Attack Successful with key length = {key_len}!")
        print("Forged Message:", new_message)
        print("Forged MAC:", new_mac)
        break
else:
    print("\n>>> Attack Failed for all key lengths.")
```

c. Generate a valid MAC for the extended message without knowing the secret key

The forged message is generated automatically by hashpumpy, including the internal MD5 padding between the original message and the appended data.

Successful forged output:

Forged Message: b'amount=100&to=alice\x80\x00...\x08admin=true'
Forged MAC: 97312a73075b6e1589117ce55e0a3ca6

This MAC is **valid** for the new message, and the attacker never knew the secret key.

d. Demonstrate that the server accepts your forged message+MAC as valid

We used the insecure server's verify() function to check the forged data:

>>> Attack Successful with key length = 14!
MAC verified successfully. Message is authentic.

The server incorrectly accepted the **forged message** and **forged MAC**, confirming that it is vulnerable to a **length extension attack**.