**HMAC**

**import hashlib**

**l = 8**

**ipad = int('01011100', 2)**

**opad = int('00110110', 2)**

**key = int('10000101', 2)**

**IV = int('11100100', 2)**

**def xor(a, b, length):**

**return format(a ^ b, f'0{length}b')**

**def hashlib\_dummy\_hash(binary\_string):**

**input\_bytes = int(binary\_string, 2).to\_bytes((len(binary\_string) + 7) // 8, byteorder='big')**

**hash\_output = hashlib.sha1(input\_bytes).digest()**

**truncated\_hash = format(hash\_output[0], '08b')**

**return truncated\_hash**

**def hmac\_simple\_binary(plaintext, key, IV, ipad, opad, block\_size):**

**k\_xor\_ipad = xor(key, ipad, block\_size)**

**z0 = format(IV, '08b') + k\_xor\_ipad**

**z1 = hashlib\_dummy\_hash(z0)**

**chunks = [plaintext[i:i + block\_size] for i in range(0, len(plaintext), block\_size)]**

**current\_hash = z1**

**for chunk in chunks:**

**z\_concat = current\_hash + chunk**

**current\_hash = hashlib\_dummy\_hash(z\_concat)**

**k\_xor\_opad = xor(key, opad, block\_size)**

**p = format(IV, '08b') + k\_xor\_opad**

**q = hashlib\_dummy\_hash(p)**

**r = q + current\_hash**

**final\_hmac = hashlib\_dummy\_hash(r)**

**return final\_hmac**

**plaintext = '1101100101001100100011011110011101000001010001110001100010111010'**

**final\_hmac\_result = hmac\_simple\_binary(plaintext, key, IV, ipad, opad, l)**

**print("Final HMAC Output:", final\_hmac\_result)**

**MD5**

**import hashlib**

**def md5\_binary(binary\_input):**

**input\_bytes = int(binary\_input, 2).to\_bytes((len(binary\_input) + 7) // 8, byteorder='big')**

**md5\_hash = hashlib.md5(input\_bytes).digest()**

**md5\_binary = ''.join(format(byte, '08b') for byte in md5\_hash)**

**return md5\_binary**

**binary\_input = '1101100101001100100011011110011101000001010001110001100010111010'**

**md5\_result = md5\_binary(binary\_input)**

**print("MD5 hash in binary:", md5\_result)**