

This is the working pseudocode solution to the program.

```
# Source Code File: Part 3: CLIENT: Public Key Cryptography
# Name:          part3.py
# Author:        <students name>

IMPORT base64
IMPORT nacl.utils
IMPORT PrivateKey, PublicKey, Box FROM nacl.public
IMPORT requests

# Generate client key pair
SET client_private_key TO PrivateKey.generate()
SET client_public_key TO client_private_key.public_key

DEFINE FUNCTION send_request(server_url):
    """Send a properly formatted request to the server."""
    # Create the request payload
    SET request_payload TO (
        "CRYPTO 1.0 REQUEST\\r\\n" +
        "Name: Student Test Name\\r\\n" +
        "PublicKey: {base64.b16encode(bytes(client_public_key)).decode()}\\r\\n"
    )

    PRINT "Sending request to the server:"
    PRINT request_payload

    # Try to send the request to the server
    TRY:
        SET response TO requests.post(server_url, data=request_payload)
        SET response_text TO response.text
        PRINT "Response from the server:"
        PRINT response_text

        # If the server responds with a success code, parse and decrypt the response
        IF response.status_code IS 200:
            CALL parse_and_decrypt_response(response_text)
        ELSE:
            PRINT "Error from server:", response.status_code, response.reason
    EXCEPT requests.RequestException AS error:
        PRINT "Request failed:", error

DEFINE FUNCTION parse_and_decrypt_response(response_text):
    """Parse the server's response and decrypt the ciphertext."""
    TRY:
        # Split the response into lines
        SET lines TO response_text.split("\\r\\n")
        IF lines[0] IS NOT "CRYPTO 1.0 REPLY":
            PRINT "Invalid response format."
            RETURN

        SET instructor_name TO None
        SET instructor_public_key TO None
        SET ciphertext TO None

        # Extract information from the response headers
        FOR line IN lines[1:]:
            IF line STARTS WITH "Name:":
                SET instructor_name TO line.split(": ", 1)[1]
            ELIF line STARTS WITH "PublicKey:":
                SET instructor_public_key TO PublicKey(base64.b16decode(line.split(": ", 1)[1]))
            ELIF line STARTS WITH "Ciphertext:":
                SET ciphertext TO base64.b16decode(line.split(": ", 1)[1])

        IF instructor_name IS None OR instructor_public_key IS None OR ciphertext IS None:
            PRINT "Incomplete response from server."
            RETURN

        PRINT "Name:", instructor_name
        PRINT "Public Key:", base64.b16encode(bytes(instructor_public_key)).decode()
```

```
        # Decrypt the ciphertext
        SET box TO Box(client_private_key, instructor_public_key)
        SET plaintext TO box.decrypt(ciphertext)
        PRINT "Decrypted Message:", plaintext.decode('utf-8')
    EXCEPT Exception AS error:
        PRINT "Failed to parse or decrypt the response:", error

IF SCRIPT IS RUN DIRECTLY:
    # Define the server URL
    SET server_url TO "http://otp.kisow.org:12001/"
    CALL send_request(server_url)
```