

```

import socket

HOST, PORT = "127.0.0.1", 5004

def xor_bytes(data: bytes, key: bytes) -> bytes:
    if not key:
        return data
    out = bytearray()
    for i, b in enumerate(data):
        out.append(b ^ key[i % len(key)])
    return bytes(out)

def enc_line(text: str, key: bytes) -> bytes:
    ct = xor_bytes(text.encode(), key)
    return ct.hex().encode() + b"\n"

def dec_line(hexline: bytes, key: bytes) -> str:
    raw = bytes.fromhex(hexline.decode().strip())
    return xor_bytes(raw, key).decode()

def main():
    print("=== Battleship 8x8 Client (with auth + encryption) ===")
    with socket.create_connection((HOST, PORT)) as s:
        f = s.makefile("rwb")
        # banner & help (plaintext)
        print(f.readline().decode(), end="")
        print(f.readline().decode(), end="")

        # --- AUTH (PLAIN) ---
        while True:
            line = input("> (REGISTER user pass | LOGIN user pass): ").strip()
            f.write(line.encode() + b"\n"); f.flush()
            resp = f.readline().decode().strip()
            print(resp)
            if resp == "AUTH OK":
                # set key as password (third token)
                key = line.split()[-1].encode()
                break

        # After this, all lines are encrypted hex
        print("(Encrypted mode: use NEW | GUESS r c | STATUS | REVEAL | QUIT)")
        print(dec_line(f.readline(), key)) # "READY encrypted_mode"

        while True:
            cmd = input("> ").strip()
            if not cmd: continue
            f.write(enc_line(cmd, key)); f.flush()

            if cmd.upper().startswith("REVEAL"):
                # read until "END"

```

```
        while True:
            msg = dec_line(f.readline(), key)
            print(msg)
            if msg == "END":
                break
    else:
        msg = dec_line(f.readline(), key)
        print(msg)
        if msg == "BYE":
            break

if __name__ == "__main__":
    main()
```