Hidden Service Writeup

In the `set_up_intro_point` function, I filled out the instantiation of `introduce_cell` by providing the required parameters, as per the Tor Rendezvous Specification (Section 1.8+). These parameters include the introduction point's onion key, the client's public key (`X_bytes`), the rendezvous point's router fingerprint, the rendezvous cookie, the hidden service's authentication type, the hidden service's descriptor cookie, and the introduction point's router fingerprint. This cell initiates the introduction process between the client, the introduction point, and the rendezvous point. The shared secret is calculated using the Diffie-Hellman key exchange; Here, `Y` is the public key of the hidden service, and `x` is the client's private key. By raising `Y` to the power of `x`, we generate a shared secret. For the `HASH_LEN`, I set it to 20 bytes, which is the length of the SHA1 hash digest, as specified in the Tor Protocol Specification (Sections 5.1.3 and 5.2). This length is used to validate the received authentication data.

In the `extend_to_hidden` function, I filled in the missing code to set up the introduction point and extend the circuit to include the introduction point. The hidden service directory and introduction point routers were chosen using the generator functions `get_directories` and `get_introduction_routers`. The rendezvous point router was selected randomly from the available Tor nodes. Finally, I then called the `set_up_intro_point` function with the above variables as parameters.

In the `get` function, I set up the tor stream similar to the first part of the assignment, called the `extend_to_hidden` function to extend the already built 3-hop circuit to the introduction node, and then called the `connect_tor_stream` function to open the stream to the hidden service.