Review Questions

1. Data is the actual information or content that is transferred in the package. Metadata is some additional information about the content or package, like routing information, protocol version, etc. In TLS and encrypted email, both the sender and receiver's ip address and port number are leaked. For TLS, if I'm in China, and I made an HTTPs request to a website in another country, which the Chinese government believes there is sensitive content. The Chinese government will know my identity because they know my IP address. For encrypted email, if I'm in Russia, and I'm sending encrypted emails to the US, the Russian government will probably knock on my door and want to see what I have sent. Even though the content in the encrypted email is encrypted and unknown to the Russian government, they can find out who I am because my IP address is not encrypted and therefore leaked.
2. The first benefit by using three nodes in a Tor circuit is enhanced security. If I'm using a single proxy server, the proxy server knows both who I'm and the destination of my request. If this server is compromised by an attacker or the government, they will know everything. If I have three nodes, the attacker needs to compromise all three node servers to eventually know who I'm and the destination of my request, because each node only knows the previous node and the next node of the Tor packet, instead of the full topology.
   The second reason we should use three nodes in a Tor circuit is we can't trust a single proxy server. The proxy server could be run by an arbitrary person or company, and there is no audit over what they are doing with my data. In fact, the proxy server can just read all of my data or even sell my data.

Task2 Writeup

In task2, I'm setting up the tor circuit by mainly implementing the CREATE cell(the cell sent to the guard node) and the EXTEND cell(the cell sent to the middle node and the exit node). Also, I implemented the authentication using multiple sets of keys (client's private and public keys, node server's private and public keys, and the node server's public and private online keys). In the function extend(circuit, node_router), I created the extend_cell by using the online_skin(node_ID + public_B + public_X), and sent it to the middle node and the exit node to get an extended_cell back. Then, I implemented the shared_X__y, shared_X__b, and secret_input (according to section 5.1.4) to verify the client and the node server get the same shared secret. In the function circuit_build_hops(circuit, middle_router, exit_router), I extend my circuit to a middle node and an exit node by calling extend() twice. In the function circuit_from_guard(guard_router, circuit_id), I generated a random number of 20 bytes to use as my x, and sent a CREATE cell containing that to a guard node. After receiving the CREATED cell, I calculated the shared secret by following the steps in 5.1.5. Finally, in the function get(), I initialized the TCP stream and called the functions above to build the Tor circuit, and eventually sent my HTTP GET request and got the html file.

Task3 Writeup

In task3, I implemented the hidden service of Tor by first setting up a rendezvous point, and informing the server at one of the introduction nodes. In the function extend_to_hidden(), I

generated a random cookie of 20 bytes first. Then, I sent a RendezvousEstablish cell to the rendezvous server and got a RendezvousEstablished cell back. After that, I requested for a hidden service directory and from there, I picked an introduction point. Then, I called the function set_up_intro_point to inform the server about my cookie and the information of the rendezvous server. I generated a introduce_cell to the introduction node with the information like my public key, the rendezvous server, the circuit id, etc. Then, I got the IntroduceAck from my target server. I got g^y from the server, and I computed the shared secret by calculating g^yx (DH). Eventually, I sent my HTTP request and successfully got the html file.