CSD Assignment

Project 2 Report - 70567 & 70568

To determine the country of each node, we use the GeoLite2 database to identify the locations of all relays. Only one relay couldn't be identified; since we can't verify its country, we excluded it from further consideration, as we can't determine whether the client would trust it.

To calculate the trust score of a guard candidate (a relay that could serve as the guard in a circuit), we apply the following rules:

- If the relay is in the same country as the client, we discard it entirely.
- If the relay is part of an alliance that includes the client's country, it receives a trust score of 0.
- If the relay belongs to an alliance that does not include the client's country, it receives the trust score associated with that alliance.
- If the relay does not belong to any alliance, it is assigned a trust score 1.0.

We also included a special case, if the client's trust in a given alliance is 0.0, the relays from that country will be ignored, that is, not even considered as candidates.

We apply the same logic to exit candidates, using the destination country instead of the client's country. Additionally, we check the exit policy and discard any candidates that block the destination IP address.

Next, in the select_path function, we adapt the pseudocode from Algorithm 1 in *"Avoiding the Man on the Wire: Improving Tor's Security with Trust-Aware Path Selection."* Specifically, we run the pseudocode twice: once for guard candidates and once for exit candidates. It's important to note that for the guard candidates, the alpha parameters are the SUGGESTED_GUARD_PARAMS and, for the exit candidates, are SUGGESTED_EXIT_PARAMS.

After identifying the Safe and Acceptable sets for guards and exits, we compute all valid combinations of Safe guard and Safe exit relays. A combination is valid only if:

- The guard and exit have different fingerprints,

- They belong to different families,
- Their countries are not in the same alliance.

Among valid combinations, we select the one with the highest combined trust score and bandwidth. To increase the performance of our implementation, we decided to first do the combination between the safe guards candidates and the safe exit candidates, because this is the optimal combination. If we didn't find any circuit, we then do the other possible combinations.

For the middle relay, we select the one with the highest bandwidth that has a different fingerprint, belongs to a different family and comes from a country outside the alliances of the other two nodes.

Finally, we print the results to the console and save them in a JSON file with the name circuit.json that follows the same structure as tor_consensus.json.

To verify the correctness of our implementation, we conducted several tests using the alliance file provided by the professor. In one test, we assigned all countries to a single alliance with a trust score of 0.0, ensuring that no valid path could be selected and, as expected, no path was found. In another test, we removed all alliance constraints, allowing the system to freely select the top three relays based on bandwidth, excluding those located in the client or destination countries (with the possible exception of the middle relay). We also ran additional tests with various alliance configurations that allowed for valid paths. All tests produced the expected results, confirming the implementation works as intended. All versions of the Project2ClientInput.json were saved in a folder called "input_jsons".

In the test where we don't have any alliance restrictions, that is, there are no alliances, we noted that the program can be optimized in terms of speed. For this to happen, in the part where we select the safe guards candidates and safe exit candidates, we could add a condition presented in the acceptable candidates, where we only select until reaching the bandwidth threshold. This, in turn, reduces the number of safe candidates and, by extension, reduces the number of combinations possible. This brings a concern where we might not be able to select the optimal path in terms of bandwidth. To reduce the probability of not having the best result, we can order by trust score and bandwidth before doing any selection. Because of this and to facilitate the tests with different alliances, we decided to have that condition in the safe selection part.

Extra information: To carry out this project, we used Live Share in Visual Studio Code.