1. Consider the Diffie-Hellman implementation `exchange::exchange_keys` located in `exchange.h`. No authenticity check is performed in either the `network::recv` or `network::send` functions. Does this present a vulnerability in the key exchange? If so, how might an attacker exploit it?
2. Recall that the HMAC message authentication code uses a cryptographic hashing algorithm alongside a key, where the hashing algorithm breaks the message into fixed sized chunks which is then fed into a running, fixed length hash value. AES-GCM does not require an HMAC, as it ensures message integrity through the `aes::gcm::GHASH` function in `aes.h`. How is this function similar to HMAC, and what advantage does AES-GCM with `GHASH` have over AES-CTR+HMAC?
3. In the `main` program, encrypted messages are exchanged via a shared key in the `util::receive_message` and `util::send_message` functions in `util.h`. When using ECB or CTR mode, `main` will generate an HMAC value for the messages to ensure that they are not modified in transit. Recall that ECB is not a recommended mode of AES. Does the inclusion of an HMAC alleviate the issues in ECB, and make it safe to use for secure communication? Why or why not?