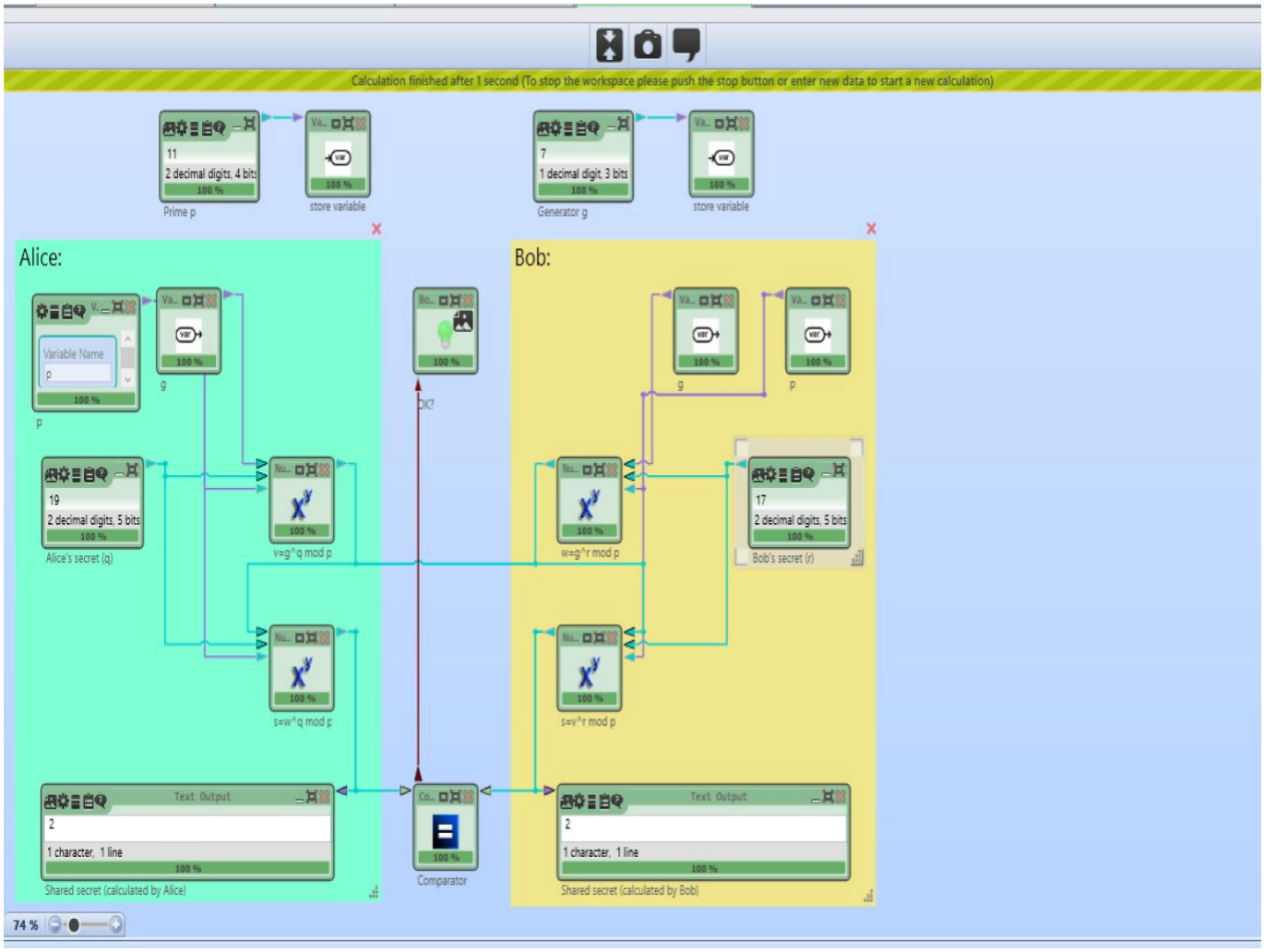
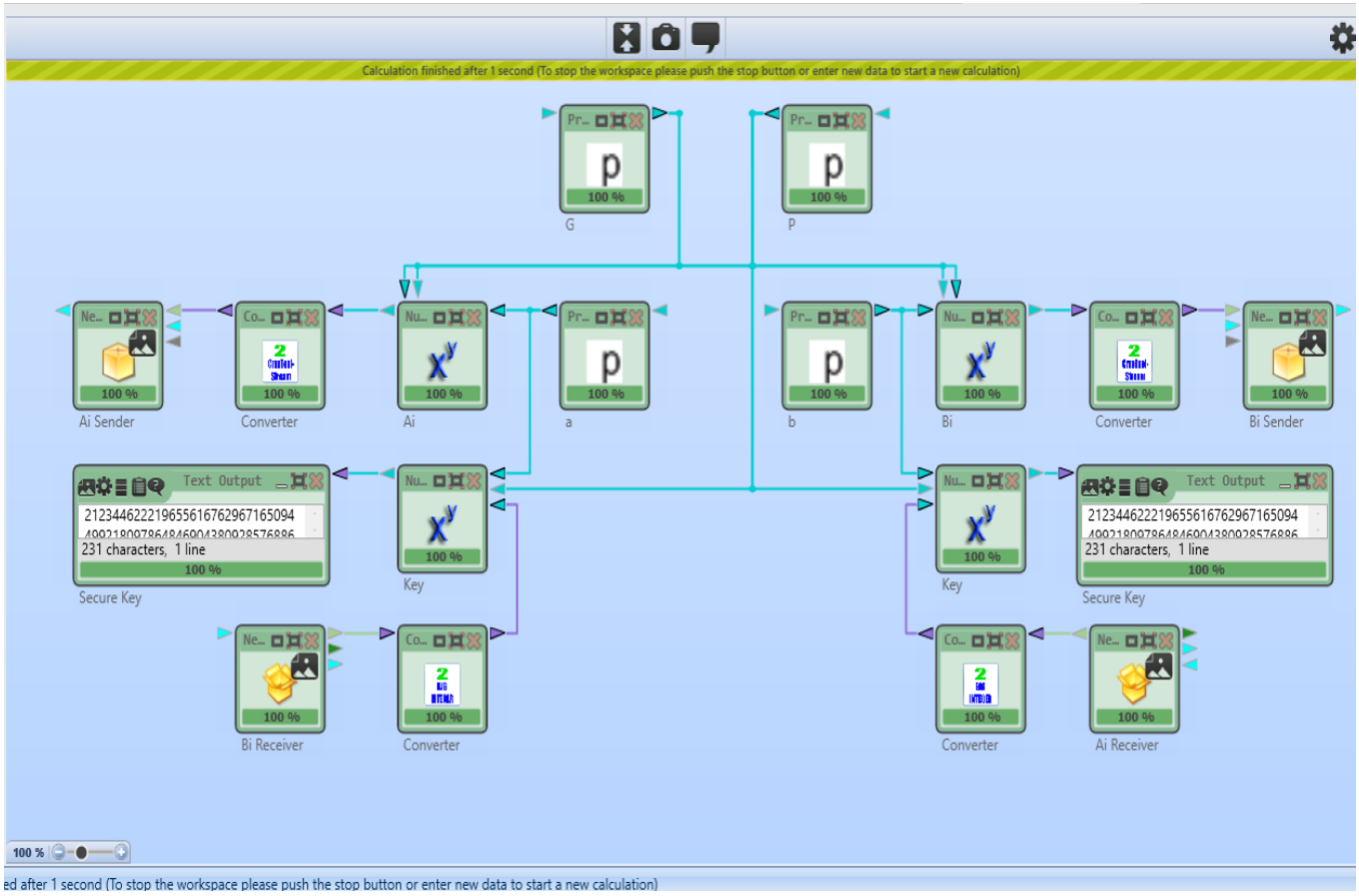
CSCI 531 Assignment 4

Mukesh Dangi

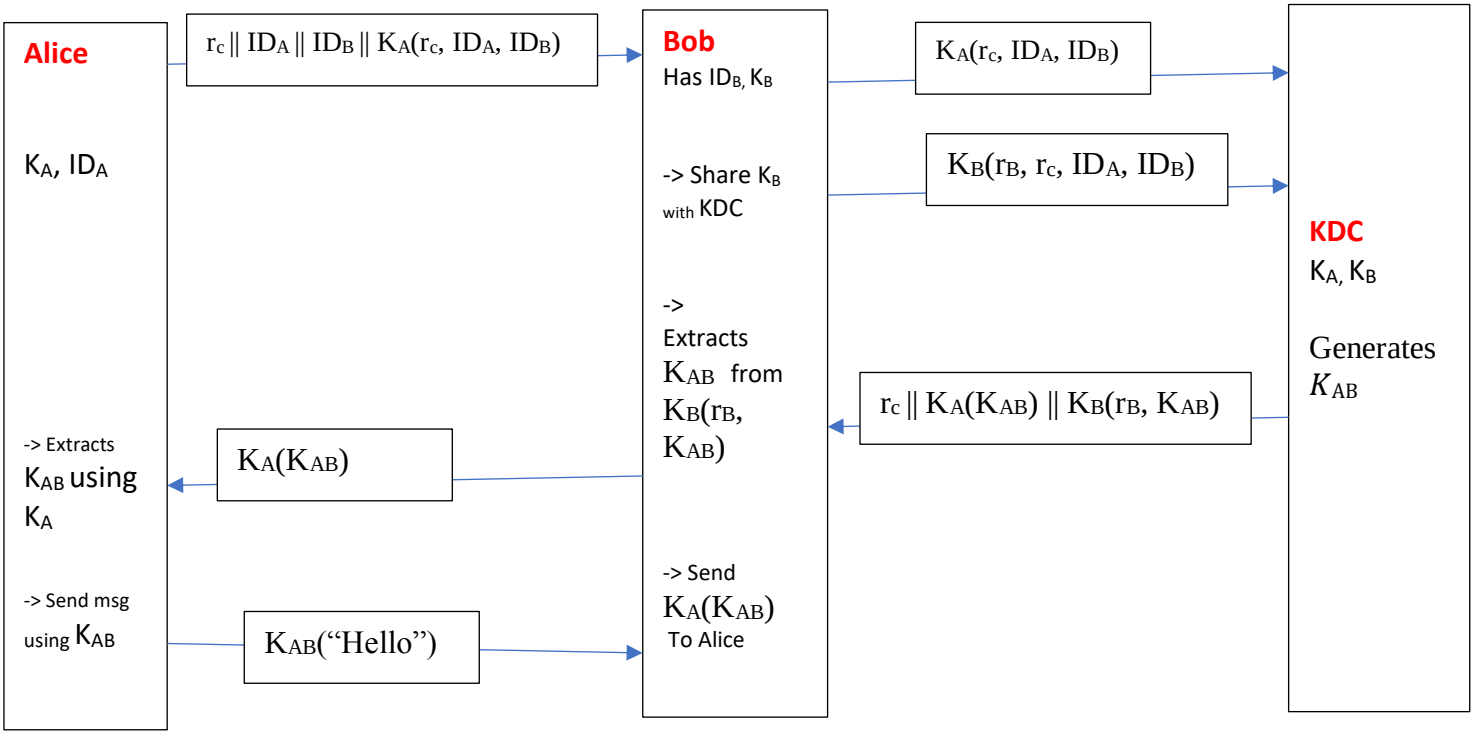
1.1 Diffie-Hellman Key Exchange



1.2 Diffie-Hellman Key Exchange over network



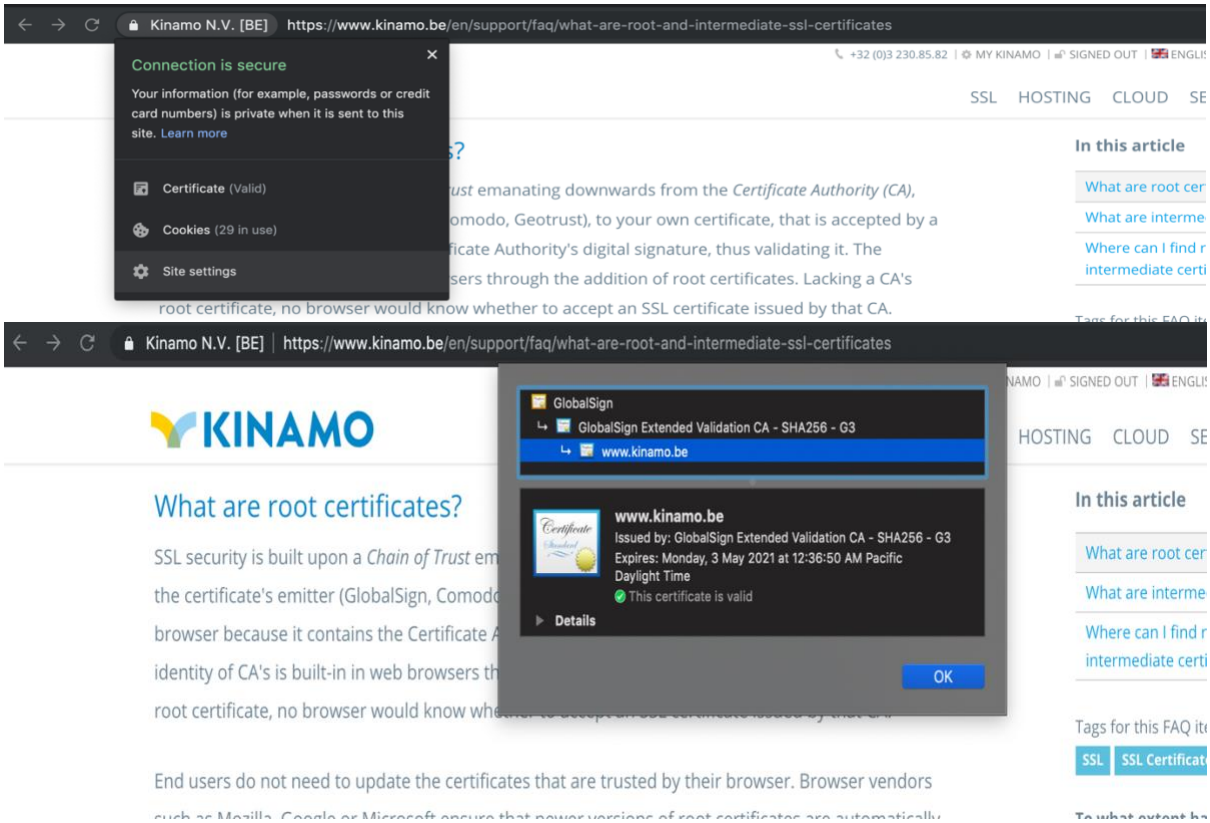
2. a) Key Management



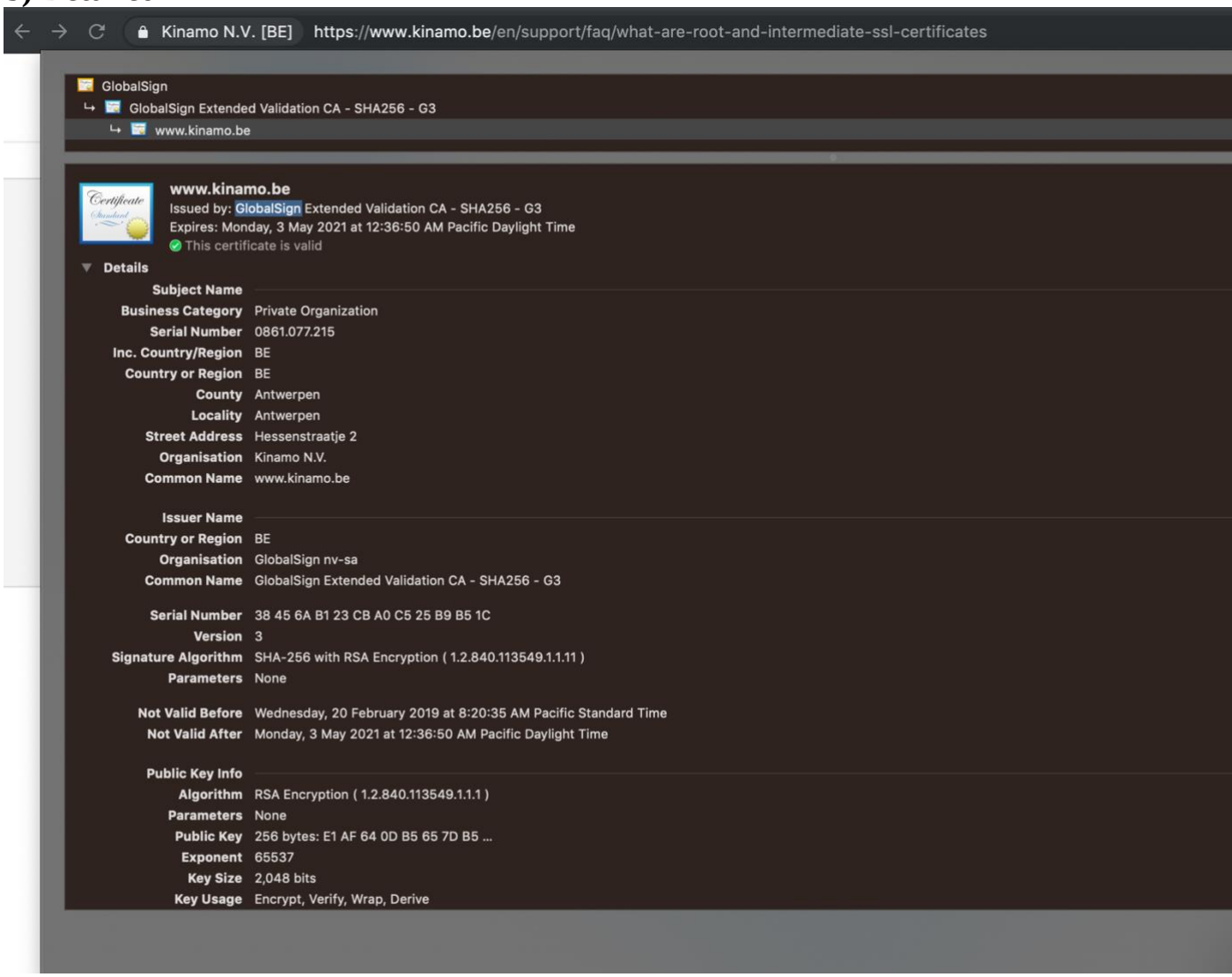
b) Bob cannot impersonate because Alice shared his key K_A with KDC so only KDC can generate K_{AB}. Here we are assuming that initially we shared with KDC and Bob was not able to impersonate himself as KDC to acquire K_A.
Since K_{AB} was generated at KDC side and KDC has K_A, Bob cannot decrypt and send a modified K_{AB} . Because Bob can do it only if he has K_A but that is not the case. So This management is secure against Bob's impersonation attack.

c) Since this key management doesn't take time stamp into consideration so replay attacks could be possible. Here If attacker is able to eve drop the K_{AB} then he can take control the communication channel. Because we are not adding any timestamp or nonce while sending the message, attacker can send and do reply attack using eavesdropped K_{AB} . Other attack may include as key K_{AB} guessing and denial of service. In denial of service attack, an attacker can keep KDC busy to generate the master keys for himself and Bob could end up waiting infinitely.

3. Digital Certificates, intermediate and root CA. a) **General**



b) Detailed



SSL security relies on chain of trust. We can assume it as hierarchy level system where if we can trust a child then parents can be trusted automatically. Here on above three screenshots we can see that intermediate Certificate signed and issued GlobalSign Extended Validation SSL CA - SHA256 - G3, and then this intermediate certificate was signed and issued by GlobalSign's root certificate, GlobalSign Root CA - R3.

4. Web Security: How SSL counters threads?

2.1 Brute Force Attack : SSL implements one time pad concept in session keys. SSL one time per session key concept able to negotiate a stronger encryption algorithm for initial session setup. So basically, exhaustive search in the key would not be helpful because of symmetric and one time session keys used during the session creation or handshake.

2.2 **Known Plaintext dictionary Attack:** SSL/TLS uses random nonce or random number per session for a client or server to generate the session keys. This extra added session helps SSL to randomize the cipher text and protect from known plain text dictionary attack.

2.3 Replay Attack : Random number used in session creation has initial 4 bytes are timestamp which always changes and completely changes for next session key or session creation. So as timestamp changes session prediction would not be possible which in return helps SSL to protect itself from replay attack.

2.4 Man-in-Middle attack: Since SSL is based on mutual authentication where client and server authenticate themselves with a third party and involves certification of mutual authentication, Man -in -middle has less chances to attacking. Man in middle would not be able to get a valid certification unless he get a valid cert from CA authority and prove Alice or Bob that he is real person.

2.5 Password Sniffing: HTTP is not secure however if we use HTTPS, then everything except some header information is end to end encrypted including password. So use of HTTPS would solve password sniffing problem.

2.6 IP Spoofing: SSL implements mutual authentication process instead of IP address to authenticate client and server so IP spoofing would not create any problem.

2.7 Connection Hijacking: Connection hijacking is possible when browser cookies are not stored in encrypted form. If an attacker hijack a connection after session authentication process then attacked still has no idea about encryption keys so SSL protocol will close the connection and would require new handshake process from start.

2.8 SYN Flooding: Since SSL is stateful and maintain state while working on top of TCP protocol so by SYN flooding attacker cannot create an overhead on SSL unless until attacker changes IP address every time for a new connection. So SSL's is not stateless and remembers all recent conversation and requests from client/server/attacker which helps SSL to avoid SYN flooding.