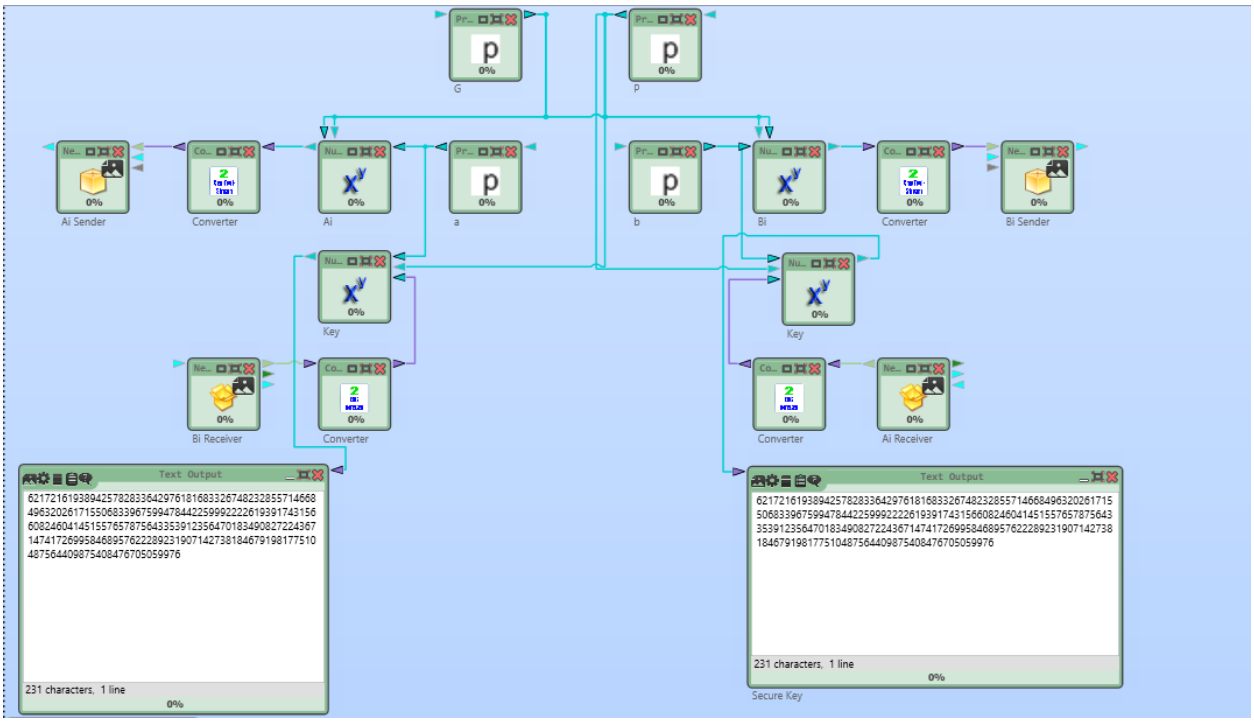
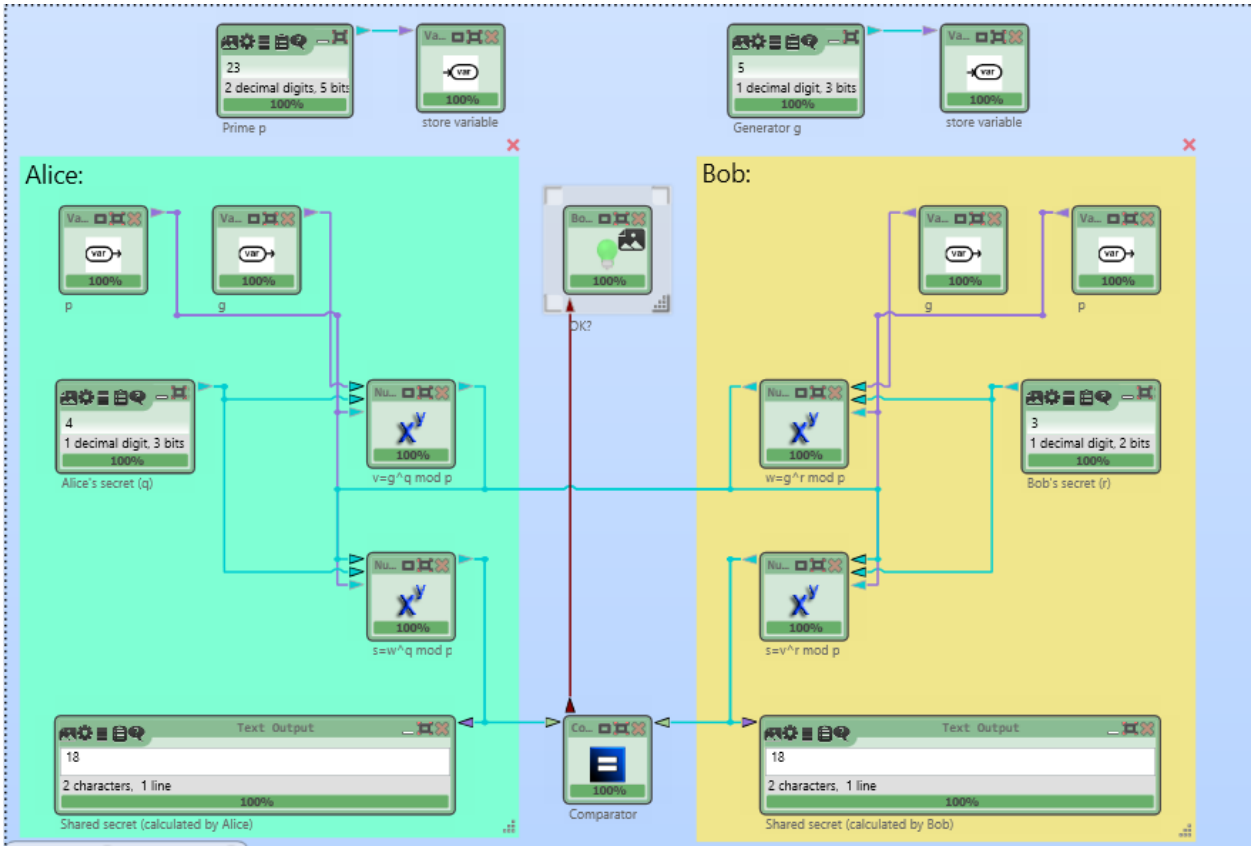
Task 1 [13 points] Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange over Network



Diffie-Hellman Key Exchange

Sample Calculation:

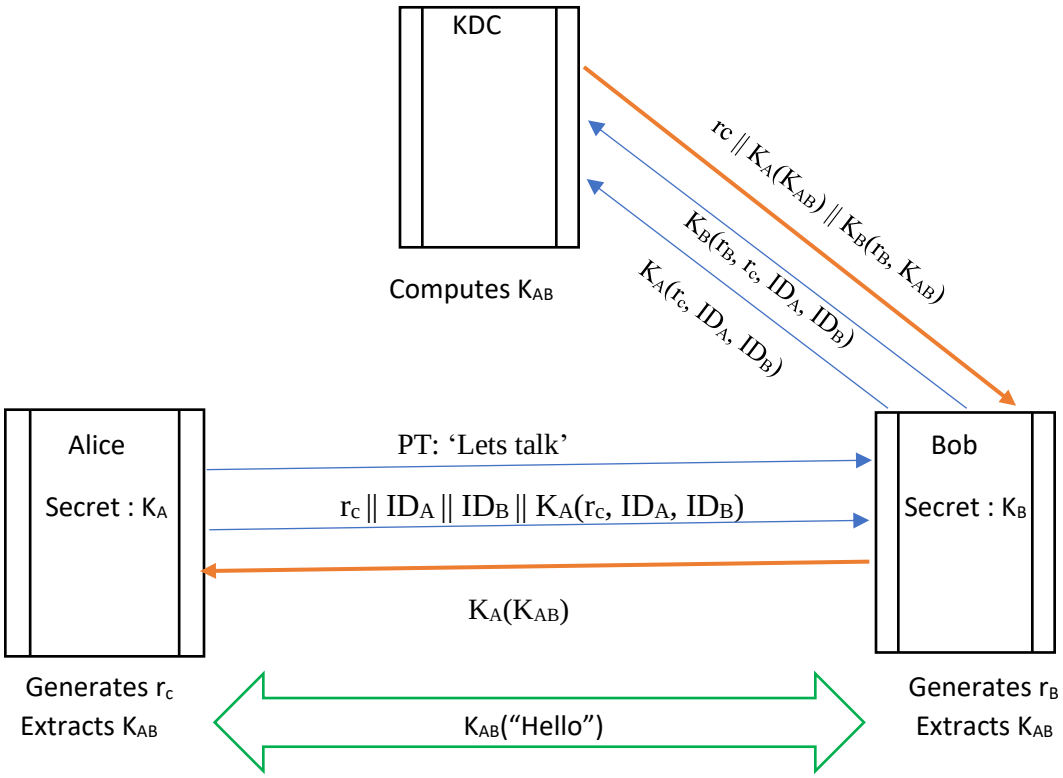
Known parameters: mod=p=23; base=g=5

On Alice's side	On Bob's side
Secret=a=4	Secret=b=3
$A=g^a \bmod p$ $=5^4 \bmod 23$ $=4$	$B=g^b \bmod p$ $=5^3 \bmod 23$ $=10$

Exchange A and B

Shared secret=$B^a \bmod p$ $=10^4 \bmod 23$ $=18$	Shared secret=$A^b \bmod p$ $=4^3 \bmod 23$ $=18$

Task 2 [45 points] Key Management



Graphical Representation of Protocol

Steps of protocol:

1. Before the interaction Alice and Bob share a secret key with the KDC, respectively denoted K_A and K_B. Alice and Bob both have unique identifiers, ID_A and ID_B that are known to each other and the KDC.
2. Alice contacts Bob in plain text and tells him that she wants to communicate confidentially. She next sends Bob a message $r_c \,||\, ID_A \,||\, ID_B \,||\, K_A(r_c, ID_A, ID_B)$ where r_c represents a random nonce and || represents concatenation, $K_A()$ indicates encryption using key K_A.
3. Bob sends two messages to the KDC. The first message is forwarded message from Alice, $K_A(r_c, ID_A, ID_B)$, and the second message is the confirmation that Bob also wants to communicate with Alice, $K_B(r_B, r_c, ID_A, ID_B)$, where r_B represents a fresh nonce generated by Bob
4. The KDC generates a fresh secret key for Alice and Bob, K_{AB}, and sends the message $rc \,||\, K_A(K_{AB}) \,||\, K_B(r_B, K_{AB})$ to Bob
5. Bob extracts the shared secret key, K_{AB}, and forwards it to Alice the part intended for Alice, $K_A(K_{AB})$.
6. Upon extracting the key K_{AB}, Alice starts sending confidential messages to Bob, K_{AB}("Hello").

Bob's Impersonation Attack:

This protocol is not secure against Bob's Impersonation Attack as Bob distributes the secret key K_{AB} to Alice which is dangerous. Eve can record the conversation, use her own key $K_{B'}$ and send it to KDC, receive messages from KDC and further extract K_{AB} using $K_{B'}$. Now once K_{AB} is known to Eve, the protocol is broken.

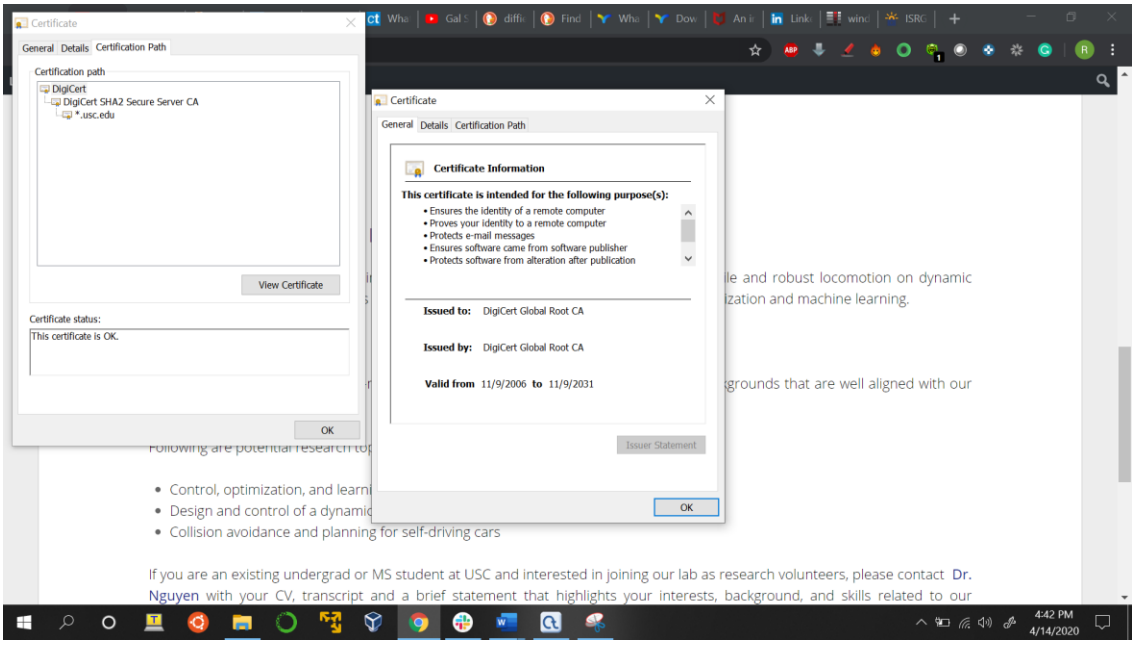
Other Attacks:

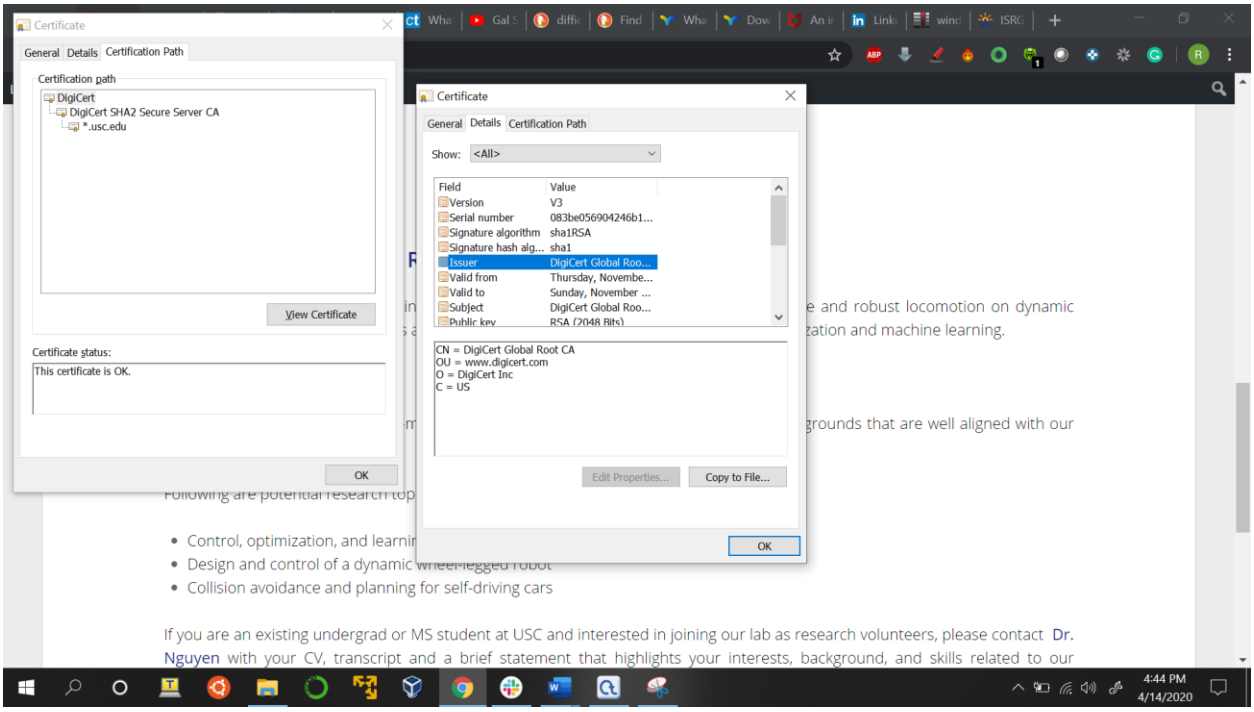
- Stealing the KDC database allows impersonation of all users and decryption of all previously recorded conversations.

Task 3 [10 points] Digital Certificates

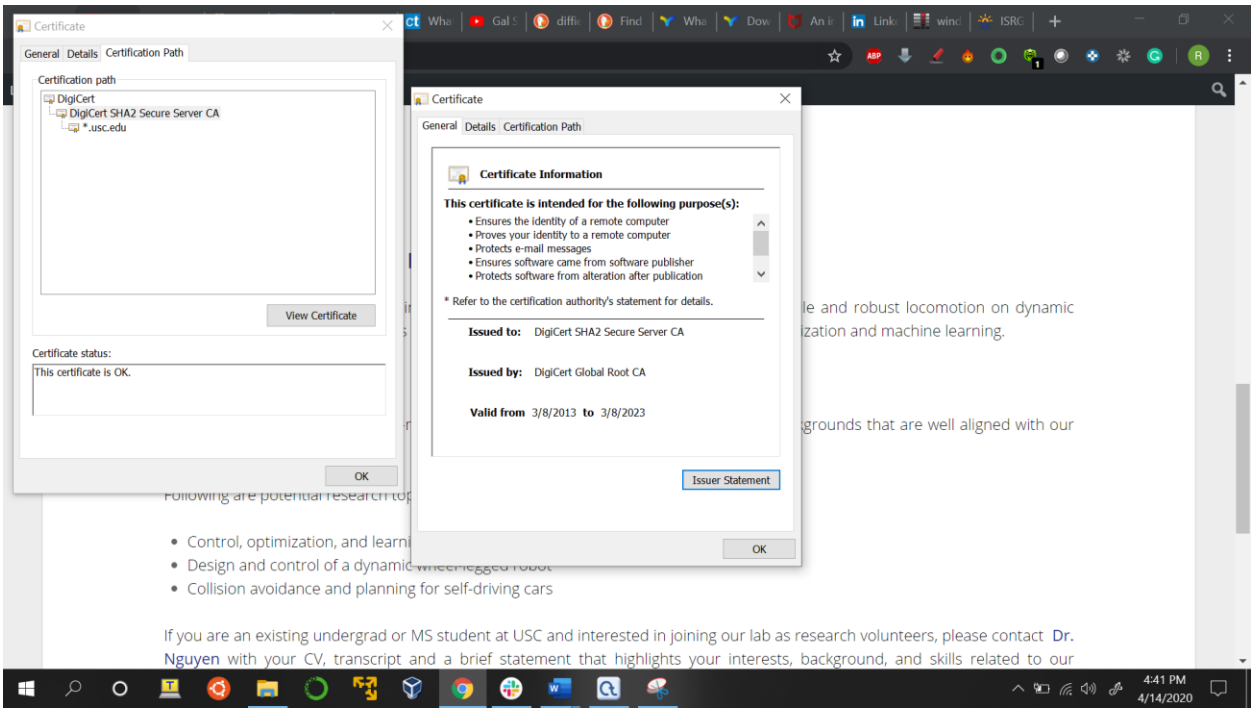
CA's root certificate is not used to sign end users' certificates directly but they use intermediate certificates that have been signed by the root certificate, and those in turn are used to validate end users' certificates. Check out the example below:-

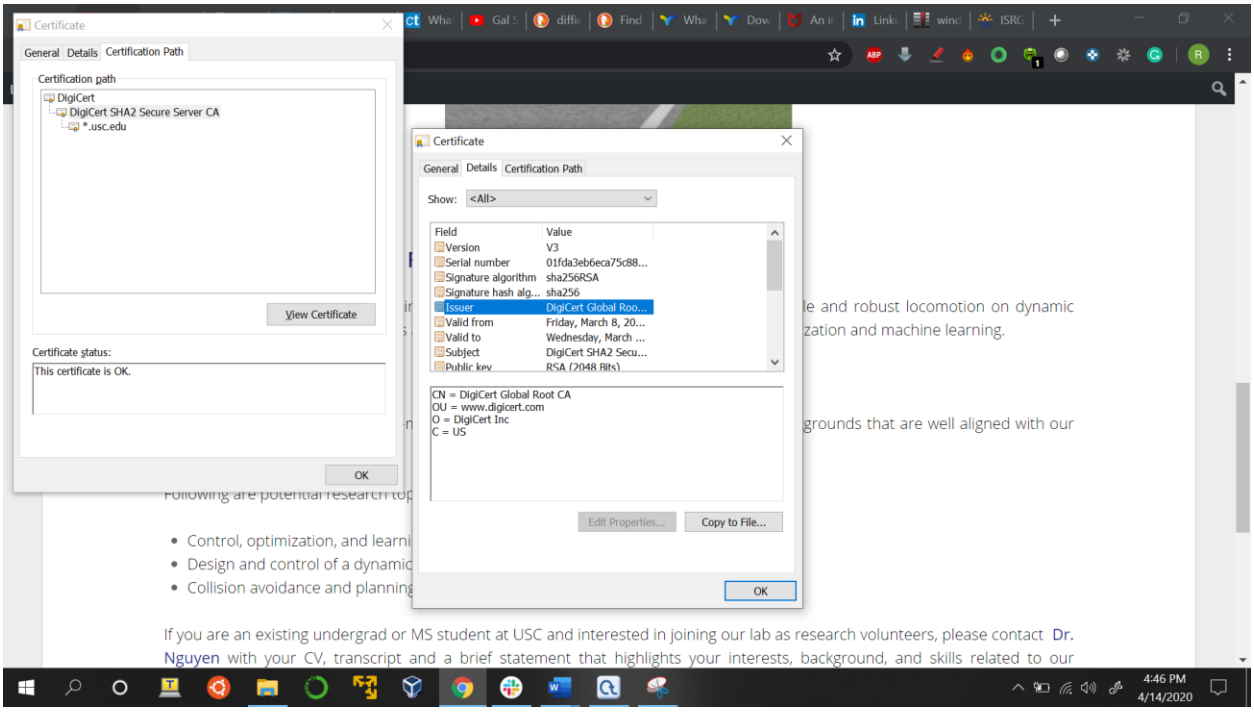
**Root CA Certificate*





Intermediate Certificate





DigiCert SHA2 Secure Server CA is intermediate authority certified by Root CA DigiCert

Task 4 [32 points] Web Security

Brute-force cryptanalytic attack: SSL implementations support a variety of public and secret key crypto algorithms which use key lengths ranging from 40 bits to 168 bits. Most common SSL software is limited to 40-bit secret keys and 521-bit RSA keys because of export restrictions. There have been several demos of cracking 40-bit secret keys but there has been no public demo of cracking a 512-bit public key.

Known-plaintext dictionary attack: SSL protects against dictionary attacks by not really using a 40-bit key. The key is actually 128 bits long with only 40 bits of the key kept secret and the rest of the key is constructed from data that is disclosed in the 'Hello' messages. This means that the dictionary cannot be simply 40 bits long as the 40-bit secret key is combined with 88-bit "disclosed" key, the resulting encryption does in fact use all 128 keys bits. Thus, dictionary must also have separate entries for all of the 128-bit keys and this makes the attack impractical.

Replay attack: Random numbers uses the first 4 bytes as the time stamp in each session. Also, client and server use nonce when they send session keys. Before the message is signed the content of the message is hashed along with the nonce and they are attached to the message. Handshake Protocol makes sure messages are sent and received with a signature hash.

Man-in-the-middle attack: Certificate validation process is done in which the domain name is compared with the domain in server's certificate. This step makes sure that the server is in the same network address specified by domain name in the certificate. This step alone protects from this attack although it is not part of SSL.

Password sniffing: Passwords are not communicated in plain text but instead encrypted to prevent password sniffing.

IP spoofing: It will still work if application authenticates based on IP address. In this case SSL may not authenticate the client. IP address is not part of SSL authentication scheme.

IP hijacking: If the attacker hijacks the connection after authentication, he has no way of knowing the encryption key. Even if the attacker hijacks it during handshaking, the attacker does not know the password and hence cannot succeed during the password authentication phase.

The two original hosts have agreed on a temporary secret key, so the host trying to hijack the connection would need knowledge of the key to stand in for one of the hosts. Otherwise it cannot send any message that will be seen as valid by the other side.

SYN flooding: SSL does not protect SYN flooding as it occurs at the TCP. SSL is built on top of TCP.