What is the difference between Tor and a 3-hop cascade Mix network in terms of the adversary/threat model, the set up and data transmission phases of communications, and their performance?

THREAT MODEL

(Global/local adversary? Passive/active adversary (i.e. Active would be doing timing attacks)?
What is the attacker able to do? What is the attack and how do they win?)

- Tor cannot protect against a global adversary while a mix network can.
- In Tor, both the guard and exit cannot be colluding, or else timing attacks can help deanonymize a user.
- In mix networks, only a single honest mix is necessary to ensure anonymity.

SETUP & DATA TRANSMISSION PHASES

- To set up a Tor circuit, you create it hop by hop until it is the desired length (3 hops). Once the circuit is set up, all traffic flows over that same path across the network (until a new circuit is set up).
- In mix networks, each message takes an independent path than all the other messages (even from the same user to the same destination).

PERFORMANCE

- Tor is a low-latency network, meaning that it delivers traffic with little additional delay.
- Mix networks actively delay messages to break timing correlations between sending and receiving times. This makes mix networks higher-latency systems.

When we looked at TLS, what is the authentication problem?

- The authentication problem is that on the Internet, we cannot be sure if we are talking to the person we intended to talk to. Someone could be impersonating that person, or machine.
- The problem to solve is to be sure that you are communicating with the party you intended to using only network communications. TLS solves this with public key cryptography and certificates.

Describe how the following about website certificates work: creation, usage, and revocation.

CREATION

- A website owner will ask for a certificate issued for their domain.
- The CA will ask for proof of control of that domain.
    - One way to do this is to ask the domain owner to put a file (full of some numbers or other data) at a particular path in the URL (e.g., domain.com/file.html).
- If the CA is able to visit this URL and see the expected contents, then they have proof that they own the domain. They may do further checks if a higher-level certificate is required (say if the website is a bank where the CA will also ask for business documents as proof).
- If all goes well, the CA will issue a certificate that has the domain name and signature of the CA along with other data like the expiry date, etc.

USAGE

- When a user visits a website, it sends its certificate to the user.
- The user's browser verifies the signature of the certificate using the CA's verification (public) key.
- If this check passes, then the browser allows the loading of the website.
- Otherwise, it blocks the site with an error message.

REVOCATION

- The CA maintains a revocation list of certificates that have not yet expired but are nonetheless not be used.
- This can happen because it was issued in error, or some malicious activity is noticed.
- The user's browser checks the revocation list when it does the certificate verification step to make sure the certificate has not been revoked.

How could one compromised Certificate Authority (CA) lead to security issue for a large number of users and websites?

- A compromised CA can issue certificates for any website, without doing the usual checks of domain ownership.
- Then any user that tries to visit a legitimate website can be served the falsely issued certificate (received before the real certificate response from the website) will use the public key to encrypt the communication between the user and the website.
- Then the adversary can intercept these connections and decrypt them (MiTM attack)

Forward secrecy, what is it and how do we achieve it (using e.g. Diffie-Hellman Key exchange)?

- Forward secrecy is the property that private keys (signing and decryption) that are compromised in the future do not allow the decryption of verification of messages that were sent in the past.
- The DHKE allows us to establish shared session keys for each (or a small number of) message using our long-term keys. Session keys are used only once or for a brief window of time.
- That way, if a session key is compromised in the future, it cannot be used to read messages recorded from the past since a difference session key would have been used in the past.
- Note that long-term keys are only there to create an authenticated tunnel between two people to do the DHKE (to prevent MiTM which can cause the two parties to do DHKE with the adversary themselves instead of each other).

Imagine Ahmad, Begül, and Cameron are participating in a 3DC protocol. You secretly observe the following coin flips among them: AB: 1, AC:1, BC: 0. You then hear their public announcements: A: 1, B: 1, and C: 1. Who paid? Don't forget that the NSA could also have paid.

- 'A' paid for the meal.
- 'A' saw a 1 and a 1. The XOR of 1 and 1 is 0. However, he announces 1, which the negation. This is only done is 'A' has paid.
- The others report the XOR of their observed coin flips.

Why do we believe that the 3DC protocol is secure?

- The only reason we are able to tell who paid is because we have complete information about the coin flips. These need to be secret between the two people flipping. Otherwise, the whole scheme is insecure.

Tor is susceptible to timing attacks and can only defend against a local adversary. Describe how a global adversary (an adversary with the ability to observe the entire Tor network, and able to interfere with the network traffic) could mount a timing attack. In a nutshell, timing attacks are where the time of packets/messages that entering and leaving the network are recorded. This information can then be used to identify entering and leaving packets as the same.

- An adversary can drop and delay packets entering the Tor network.
- They can choose one user and one website and add timing patterns using delays or packet drops from the client side to see which website the user is communicating with.
- The timing patterns will be preserved as they cross the network since Tor is a low-latency network, does not delay traffic and tries to deliver it with almost no delay overhead.