2014

Question 2

a

- $p = 227$
- $\alpha = 5$
- $x = 12$
- $y = 3$

$$(\alpha^x)^y \mod p \equiv (\alpha^y)^x \mod p$$

Alice's public: $5^{12} \mod 227 \equiv 82$

Bob's public: $5^3 \mod 227 \equiv 125$

Secret: $82^3 \mod 227 \equiv 125^{12} \mod 227 \equiv 212$

b

A substitution cipher offsets all the characters in the alphabet. If Trudy knows the encrypted key, she knows the length of the key too. She can now organise the message in columns and re-arrange the message till it makes sense, with a maximum of $k!$ combinations.

c

Alice can use the root CA's public key to decrypt the X.509 certificate. If it is genuine, she can extract the public key from the certificate.

To check that Bob has the corresponding private key. She encrypts a nonce using Bob's public key, and sends it to him. If he correctly sends back a plaintext version, she knows it's Bob.

d

$K_{A,B}$ is shared secret

This protocol can be broken by taking shortcuts, i.e. Alice sends a nonce immediately as the connection opens. In this case, as displayed in the diagram, Trudy can initiate a connection with Bob, and when Bob attempts to authentication Trudy, she can open another connection with Bob and send that same challenge. Once Bob authenticates using this challenge, Trudy can send that response in the first connection to authenticate.

To fix this, Alice/Trudy should be authenticated first, i.e. split message 1 into two steps so that R_T is sent after Alice/Trudy have sent $K_{AB}(R_B)$. Once Bob has verified Alice/Trudy, then then send their own challenge so that Alice/Trudy can authenticate Bob.

The fixed protocol (challenge-response):

1. Alice initiates connection.
2. Bob sends Alice a nonce.
3. Alice responds by encrypting it with the shared secret. Alice sends this to Bob with another nonce.
4. Bob decrypts the message and verifies it. He takes Alice's nonce and encrypts it with the shared secret to send
5. Alice decrypts the nonce and verifies it.

Question 3

a

	TCP	UDP
Connection	P2P, connection orientated	Connectionless
Function	Connection based	Used for message transport and transfer
Usage	High reliability transmission	Fast, efficient transmission time
Reliability	Yes (rdt)	None
Packet Ordering	In-order	Out-of-order
Speed of Transfer	Slow	Fast (best effort)
Data Flow Control	Set window size	None
Error Checking	Yes, and recovery	Yes, no recovery
Handshake	3-Way (SYN, SYN-ACK, ACK)	None (connectionless)

	TCP	UDP
Examples	HTTP, telnet, ssh, ftp, smtp	VoIP, DHCP, DNS

b

Domain Name System is a hierarchical decentralised naming system for resources connected to the internet. It assigns a hierarchy of names to IP addresses.

Examples: `google.ie.`=`209.85.203.94`

- Root servers resolve Top-Level Domains (`ie`, `com`, `net`, etc.)
- TLD servers responsible for their respective subdomains (`google.ie`, `amazon.com`, etc.)
- Authoratize servers responsible for subdomains (`foo.bar.net`, etc.)

If all the DNS servers went down, domain names wouldn't be resolvable. However, if you knew the IP of a given domain, you could still access it.