Each problem worth 10 points.

Problem 10.2 from the textbook:
Answer:

a. $\varphi(11) = 10$
$2^{10} = 1024 = 1 \bmod 11$
If you check 2^n for $n < 10$, you will find that none of the values is 1 mod 11.
b. 6, because $2^6 \bmod 11 = 9$
c. $K = 3^6 \bmod 11 = 3$

Problem 15.6 from textbook:
a. A believes that she shares K'_{AB} with B since her nonce came back in message 2 encrypted with a key known only to B (and A). B believes that he shares K'_{AB} with A since N_A was encrypted with K'_{AB}, which could only be retrieved from message 2 by someone who knows K'_{AB} (and this is known only by A and B). A believes that K'_{AB} is fresh since it is included in message 2 together with N_A (and hence message 2 must have been constructed after message 1 was sent). B believes (indeed, knows) that K'_{AB} is fresh since he chose it himself.
b. We consider the following interleaved runs of the protocol:

1. A →C(B) : A, N_A
1`. C(B) →A : B, N_A
2`. A →C(B) : E(K_{AB}, [N_A, K'_{AB}])
2. C(B) →A : E(K_{AB}, [N_A, K'_{AB}])
3. A →C(B) : E(K'_{AB}, N_A)

C cannot encrypt A's nonce, so he needs to get help with message 2. He therefore starts a new run with A, letting A do the encryption and reflecting the reply back. A will accept the unprimed protocol run and believe that B is present.

c. To prevent the attack, we need to be more explicit in the messages, e.g. by changing message 2 to include the sender and receiver (in this order), i.e. to be E(K_{AB}, [A, B, N_A, K'_{AB}]).

Problem 15.7 from the textbook:
A typical PKI consists of seven core components. These are briefly described below:

1. Digital certificates (public-key certificates, X.509 certificates): A digital certificate is a signed data structure that binds one or more attributes of an entity with its corresponding public key. By being signed by a recognized and trusted authority (i.e. the Certification Authority) a digital certificate provides assurances that a particular public key belongs to a specific entity (and that the entity possesses the corresponding private key).

2. Certification Authority (CA): Certification Authorities are the people, processes and tools that are responsible for the creation, issue and management of public-key certificates that are used within a PKI.

3. Registration Authority (RA): Registration Authorities are the people, processes and tools that are responsible for authenticating the identity of new entities (users or computing devices) that require certificates from CAs. RAs additionally maintain local registration data and initiate renewal or revocation processes for old or redundant certificates. They act as agents of CAs (and in that regard can carry out some of the functions of a CA if required).

4. Certificate repository: A database, or other store, which is accessible to all users of a PKI, within which public-key certificates, certificate revocation information and policy information can be held.

5. PKI client software: Client-side software is required to ensure PKI entities are able to make use of the key and digital certificate management services of a PKI (e.g. key creation, automatic key update and refreshment).

6. PKI-enabled applications: Software applications must be PKI-enabled before they can be used within a PKI. Typically this involves modifying an application so that it can understand and make use of digital certificates (e.g. to authenticate a remote user and authenticate itself to a remote user).

7. Policy (Certificate Policy and Certification Practice Statement): Certificate Policies and Certification Practice Statements are policy documents that define the procedures and practices to be employed in the use, administration and management of certificates within a PKI.

Problem 16.4 from textbook:
All three really serve the same purpose. The difference is in the vulnerability. In Usage 1, an attacker could breach security by inflating Na and withholding an answer from B for future replay attack, a form of suppress-replay attack. The attacker could attempt to predict a plausible reply in Usage 2, but this will not succeed if the nonces are random. In both Usage 1 and 2, the messages work in either direction. That is, if N is sent in either direction, the response is E[K,
N]. In Usage 3, the message is encrypted in both directions; the purpose of function f is to assure that messages 1 and 2 are not identical. Thus, Usage 3 is more secure.

Problem 16.6 from textbook:

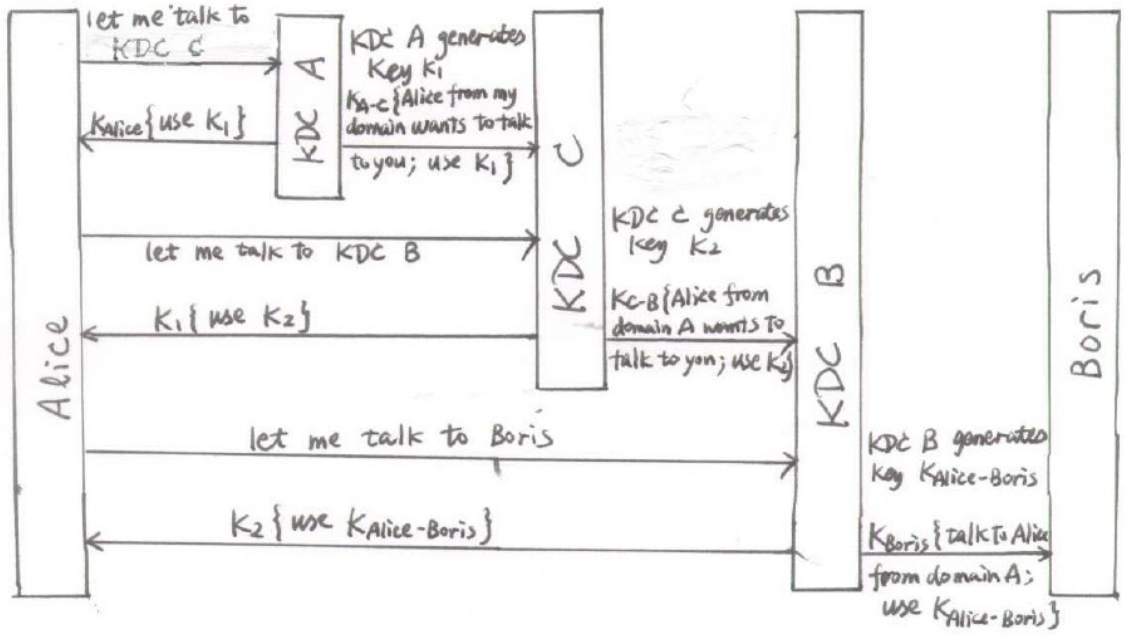
a. This is a means of authenticating A to B. R1 serves as a challenge, and only A is able to sign R1 so that it can be verified with A's public key.
b. Someone (e.g., C) can use this mechanism to get A to sign a message. Then, C will present this signature to D along with the message, claiming it was sent by A. This is a problem if A uses its public/private key for both authentication, signatures, etc.

Problem 6:
Answer:
If Alice's Diffie Hellman public number is encrypted using Bob's key and Bob's public number is encrypted using Alice's key, then Eve cannot decrypt both public numbers. Although Eve can inject its own public numbers into the channel and send to Alice/Bob, she wouldn't be able to compute any of the Diffie Hellman key shared between herself and Alice or Bob, so the man-in-the-middle attack will fail.

Alternatively, there are several possible ways to defend against MitM for DHKE:

(1) Alice and Bob can publish their public numbers in a trusted place;
(2) They can encrypt their public numbers using a secret key that is pre-shared among them;
(3) They can sign their public numbers using their private key and verify it using each other's public key;
(4) If they pre-share a secret key, in the end of the protocol they can compute a keyed hash or MAC over the final agreed secret key in DH, or over the public numbers they have exchanged. This way they can verify the public numbers or the final key hasn't been modified by an attacker.

Problem 7:
Sample Answer:
The procedure will be something like this:



(Note: If a ticket is used to suppress the two messages from KDC to one message, that is also valid answer)

Problem 8:
No. Knowledge of the hash of Alice's password (Y=H(pwd)), which is stored in the server database, is sufficient to impersonate Alice's workstation to Bob, since Eve can use it to compute H(Y||R).

Additional problems for 471 students only:

Problem 10.4 from the textbook:
Answer:

$x_B = 3$, $x_A = 5$, the secret combined key is $(3^3)^5 = 3^{15} = 14348907$.

Problem 15.8 from the textbook:

Answer:

The primary weakness of symmetric encryption algorithms is keeping the single key secure. Known as key management, it poses a number of significant challenges. If a user wants to send an encrypted message to another using symmetric encryption, he must be sure that she has the key to decrypt the message. How should the first user get the key to the second user? He would not want to send it electronically through the Internet, because that would make it vulnerable to eavesdroppers. Nor can he encrypt the key and send it, because the recipient would need some way to decrypt the key. And if he can even get the get securely to the user, how can be he certain that an attacker has not seen the key on that person's computer? Key management is a significant impediment to using symmetric encryption.

Additional problems for 571 students only:

Problem 10.3 from the textbook:
Answer:

Alice and bob could simply exchange x^a and y^a with each other. Then they can obtain a common key $(xy)^a$ by multiplying them together.

However, Eve can easily do the same and obtain the same key, just by eavesdropping and multiplying the public information.

Yes, Eve can find the secret numbers x and y in this case. Note that if participants sent each other x^a for some public number *a,* then the number x can be directly solved by first computing $b = a^{-1}(mod\ q - 1), i.e., ab = 1\ (mod\ q - 1),\ and\ then\ (x^{ab}\ ) \equiv x\ mod\ q$ can be obtained.

Problem 15.2 from the textbook:
Answer:

i) sending to the server the source name A, the destination name Z (his own), and E(K_a, R), as if A wanted to send him the same message encrypted under the same key R as A did it with B

ii) The server will respond by sending E(K_z, R) to A and Z will intercept that

iii) because Z knows his key K_z, he can decrypt E(K_z, R), thus getting his hands on R that can be used to decrypt E(R, M) and obtain M.