CIS-427 UM-Dearborn

HW5 – additional credit

With Dr. Zheng Song

Student: Demetrius Johnson

December 5, 2022

Firm Submission Deadline: 11:59PM, Dec/05/2022.

Question 1 (15 points)

- “rmij’u uamu xyj” cipher text = “wasn’t that fun”

Question 2 (15 points)

- No, you cannot “decrypt” or “reverse” the hash of a message to get the original message. That is the very purpose for why we have developed and use hash functions; they are complex functions where you cannot compute the inverse of those functions, and thus you cannot compute the inverse of their outputs (the hashed message). There is only one way to “decrypt” a hashed message, and that is to guess/predict the original message, hash that prediction, and see if the hash matches the hashed message you wish to “decrypt”.

Question 3 (15 points)

Suppose N people want to communicate with each of N – 1 other people using symmetric key encryption. All communication between any two people, i and j, is visible to all other people in this group of N, and no other person in this group should be able to decode their communication. How many keys are required in the system as a whole? Now suppose that public key encryption is used. How many keys are required in this case?

- With symmetric key encryption, the same key is used to encrypt and decrypt a message.
- If there are N users in the network, and each user wants to communicate with each of the N-1 other users, each group of 2 communicators will need 1 set of identical keys.
- This is now a question of combinations: how many different scenarios are possible where each user is paired with another user?
- If user A communicates with user B using a symmetric key, that is the same as user B communicating with user A using an identical symmetric key. So, AB communication and BA communication are the same and thus only 1 pair of keys is assigned to the two users.
- Thus, we will use a combination (not permutation) since order of two users communicating does not matter:

$${}_{n}C_{r} = \frac{n!}{r!(n-r)!}$$

${}_{n}C_{r}$ = number of combinations

n = total number of objects in the set

r = number of choosing objects from the set

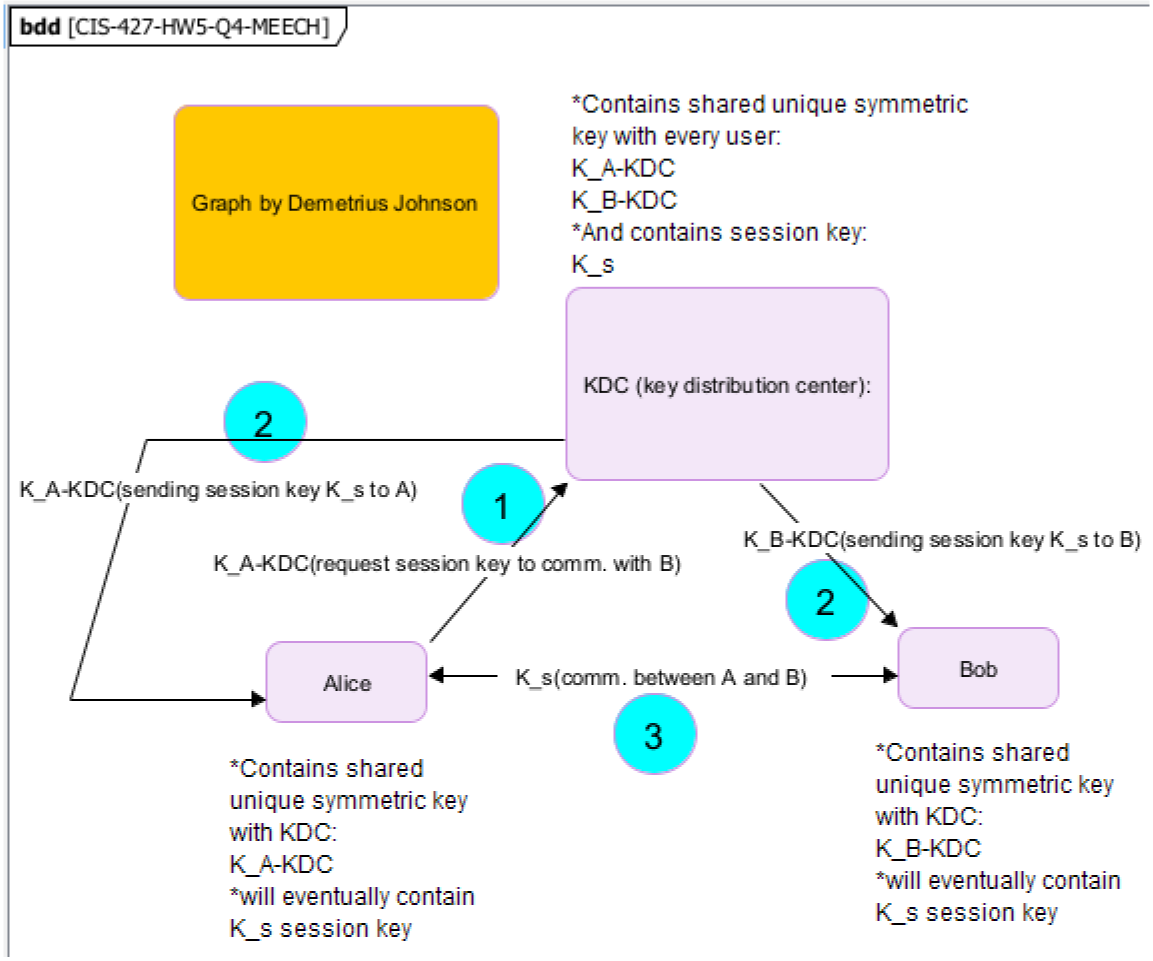
○

- Thus:
  - $_{N}C_{2}$ = N! / 2!(N – 2)!
  - = N! / 2(N-2)!
    - N! can be broken down into N(N-1)(N-2)!
  - = N(N-1)(N-2)! / 2(N-2)!
  - Cancel numerator and denominator terms:
  - = N(N-1) / 2
  - (N^2 – N) / 2
- **Thus, with symmetric key, you need (N^2 – N) / 2 *unique* identical *pairs* of keys so that all N users in the network can communicate securely with each of the other N-1 users.**
- In the same scenario with N users, if instead public key encryption is used, then you would need a private key for each of the N users, and also a public key for each of the N users.
- Secure communication between any 2 users would work as such: if user A wants to send a secure (encrypted) message to user B, then A should encrypt the message using the public key of B; finally, B will receive the message and decrypt it using the private key. If B wishes to reply to A, it should encrypt its response using A's public key, and A will then receive it and decrypt the message using its private key.
- In this scenario, in order for secure end to end communication between any two users, all private keys should only be used to decrypt messages and never to encrypt them, otherwise everyone else has the public key and will receive the broadcasted message in the network and be able to capture and decrypt the message using the pubic key.
- **Thus: You need N unique private keys and N unique public keys:**
  - **You need 2*N *unique* keys in total.**

Question 4 (20 points)

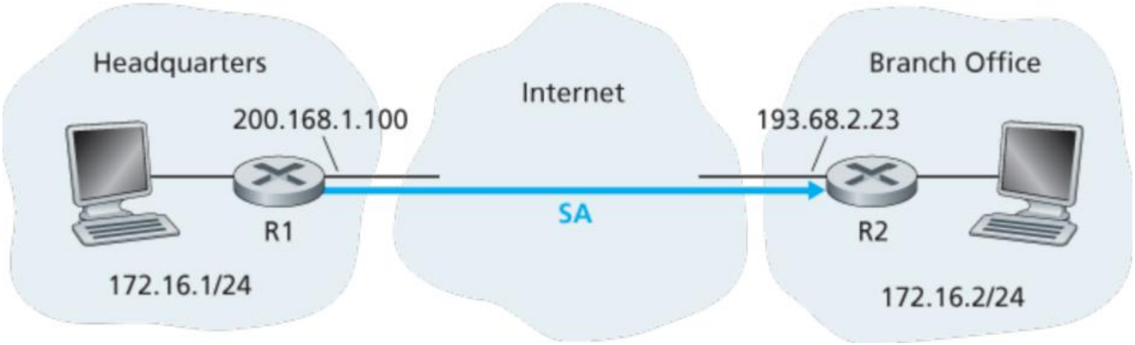
Suppose Alice wants to communicate with Bob using symmetric key cryptography using a session key Ks. We learned how public-key cryptography can be used to distribute the session key from Alice to Bob. In this problem, we explore how the session key can be distributed—without public key cryptography—using a key distribution center (KDC). The KDC is a server that shares a unique secret symmetric key with each registered user. For Alice and Bob, denote these keys by KA-KDC and KB-KDC . Design a scheme that uses the KDC to distribute Ks to Alice and Bob. Your scheme should use three messages to distribute the session key: a message from Alice to the KDC; a message from the KDC to Alice; and finally a message from Alice to Bob. The first message is KA-KDC (A, B). Using the notation, KA-KDC , KB-KDC, Ks, A, and B, what are the 2nd and 3rd messages? Explain your answer.

The following diagram I generated using Visual Paradigm:


bdd [CIS-427-HW5-Q4-MEECH]
Graph by Demetrius Johnson
*Contains shared unique symmetric key with every user:
K_A-KDC
K_B-KDC
*And contains session key:
K_s
KDC (key distribution center):
2
K_A-KDC(sending session key K_s to A)
1
K_A-KDC(request session key to comm. with B)
K_B-KDC(sending session key K_s to B)
2
Alice
K_s(comm. between A and B)
Bob
3
*Contains shared unique symmetric key with KDC:
K_A-KDC
*will eventually contain
K_s session key
*Contains shared unique symmetric key with KDC:
K_B-KDC
*will eventually contain
K_s session key

- **The first message** is an encrypted request to the KDC server from A, where A is asking for a session key that can be distributed to A and B so that A and B can communicate with each other using the distributed session key.
- **The second message** is a response from the KDC server which sends the session key K_s to both A and B, using the KDC symmetric key encryption associated with the respective user (i.e. KDC-A or KDC-B) so that no one can intercept the key while it is being distributed to a given client.
- **The third and final message** is a message between A and B, where A sends its communication to B using the sessions key K_s. Both A and B have the shared key, and so B will be able to decrypt A’s message.

Question 5 (15 points)



a) T or F: When a host in 172.16.1/24 sends a datagram to a host in 172.16.2/24, the router R1 will change the source and destination address of the IP datagram.

- TRUE
  - R1 will change it to the address of the public port on the router: 200.168.1.100 and use NAT (network address translation) to keep track of the private-to-public IP and port number mapping.

b) T or F: Suppose a host in 172.16.1/24 initiates a TCP connection to a Web server in 172.16.2/24. The routers between R1 and R2 can see PORT number=80 in the TCP headers.

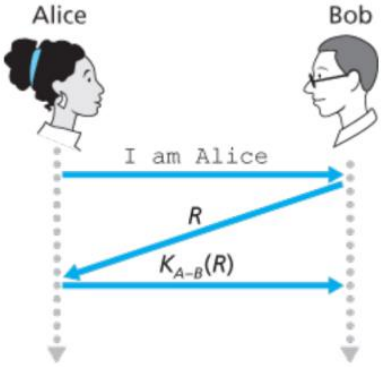
- FALSE
  - The routers between R1 and R2 are essentially intermediary hops/nodes across the internet and (with the exception of NAT for translating and tracking private to public IP addresses and vice versa) encapsulates L4 (frames) data inside a L3 packet. The routers merely forward the L3 packet out of one of their respective ports using longest prefix matching based on their current routing table.

c) T or F: Consider sending a TCP segment from a host in 172.16.1/24 to a host in 172.16.2/24. Suppose the acknowledgment for this segment gets lost, so that TCP resends the segment. Because IPsec uses sequence numbers, R1 will not resend the TCP segment.

- FALSE
  - Sequence numbers in IPsec as part of the L3 header information (IPsec is a Layer 3 security protocol) always start at 0 and simply increments by 1 each time sender transmits a packet. The idea is solely to prevent replay attacks.
  - On the other hand, if a TCP (L4) segment is not acknowledged, it will be retransmitted after some timeout period occurs. The same sequence number is sent in TCP, but in IPsec a new sequence number is sent since a new packet is being transmitted. They do not have any overlap with the sequence number as IPsec operates in L3 and TCP operates in L4.

Question 6 (20 points)

Consider an authentication protocol where Alice authenticates herself to Bob. We use KA-B to denote the symmetric secret key shared by Alice and Bob. Assume Alice and Bob can add more rounds of communications but they can only use the symmetric secret key.



a) Give a new authentication protocol, in which while Alice is authenticating herself to Bob, Bob will also authenticate himself to Alice.

1. Alice sends an encrypted message of her name to Bob using the shared key.
2. Bob will receive the message and attempt to decrypt it. If the message can be decrypted using the shared key,
3. Then, Bob will encrypt another message of his own name and append a random "once-in-a-lifetime" nonce value R and send it to Alice.
4. If Alice can decrypt the message using the shared key, she will have authenticated that it was Bob who has responded.
5. Now Alice will encrypt only the nonce value and send it back to Bob.
6. When Bob receives the encrypted data, he will try to decrypt it. If he can decrypt it, he will also compare it to the original encrypted nonce value he generated and sent to Alice. If the nonce value matches the one which he sent to Alice, then he will have authenticated Alice.

b) Show how to perform a man-in-the-middle attack on the new authentication protocol (i.e., Trudy, pretending to be Alice, can now authenticate herself to Bob as Alice and to Alice as Bob).

1. Since Alice and Bob already have the shared keys, and since in my protocol they are required encrypt every communication between each other using the shared key during the authentication process, there is no way for a man in the middle attack to occur unless an attacker somehow obtains the shared key.
2. If a shared key is obtained, then attacker can accept communication from both Alice and Bob and forward them along as if Trudy (the attacker) were not there, and make sure to capture the

nonce value, which can be decrypted and encrypted since the attacker should have the shared key.

3. Then, Trudy can use the shared key to encrypt and send messages to either Alice or Bob pretending to be one of them (Alice or Bob) since Alice and Bob think they have authenticated each other without any middleman.
4. Remember, this attack can only work against my protocol if an attacker somehow obtains the shared key.