1.

The Diffie-Hellman (DH) key agreement protocol is extended by the Station-to-Station (STS) protocol, which includes an additional authentication step utilizing digital signatures. By adding this extra step, STS protocol can counter against man-in-the-middle (MITM) attacks.

2.

No, the described use of a hash function as a commitment scheme does not offer the same security guarantees. After the commitment is submitted, a malicious party can easily find another message m' such that (r XOR m') = (r XOR m). For instance, when opening a commitment, the malicious party can flip any bit of the committed message by flipping the corresponding position of the random number.

3.

No. With perfect forward secrecy (PFS), messages from previous sessions should not be able to decrypted even if an attacker obtains the long-term shared secret key k later. Since k is used in the protocol to encrypt the session key Ks, compromising k would enable an attacker to obtain the session key from the recorded messages and then decrypt messages from previous sessions.

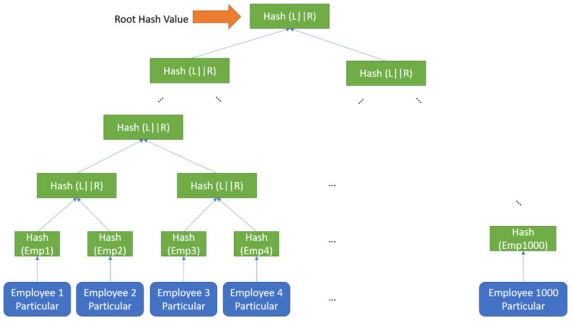
4.

- The session key can be used by C for encrypting the print job $E(K_{C,V},p)$ using AES-CTR mode.
- Then, MAC for the cipher text can be computed by C by using HMAC-SHA1 with $K_{C,V}$
- The MAC is appended to the cipher text and then sent together by C.
- The printer V checks MAC is valid, and then decrypt p the cipher text using $K_{C,V}$.

While the above answer gets full mark for this exam, ideally different keys should be used for encryption and MAC calculation. In particular, if CBC mode encryption is used with CBC-MAC, different key MUST be used. Also Encrypt-then-MAC should be used, instead of MAC-then-encrypt.

5.

Alice can generate a hash tree as follows. The blue box below is a set of data items of each employee. For the intermediate nodes, L indicates the left child node and R indicates the right child node.



Then, Alice will upload all employee data (with ID) to the cloud. After uploading employee data can be removed. But Alice keeps (at least) the root hash value she calculated for later verification.

When Alice downloads some employee's data later on, it needs to get a set of hash values that are required for Alice to calculate the root hash value. For instance, if Alice downloads the data of Employee 1 to 500 (let's assume it corresponds to the first half of the tree for the sake of simplicity), the right hash value below the root hash value in the above figure needs to be downloaded additionally.

Then, Alice uses the downloaded employee data and the hash value to calculate the root has value and see if it matches the one she kept locally. If they match, Alice can be convinced that the data is intact.
*** If Alice stores all hash values locally, actually there is no need to use MHT. So subtracted 2pt.*

6.

1. In Needham-Schroeder protocol, there is no check on the validity/expiration of a ticket. Eve can therefore retrieve the session key and ticket that were used to communicate with Bob using the logged messages and the out-of-date key. Now Eve can talk to Bob pretending to be Alice.

Eve recovers 'ticket' and 'Ka,b ' from the following logged message:

KDC →Alice: E(Ka,kdc , N1 ‖ "IDB " ‖ Ka,b ‖ ticket)

Now, Eve does the following:

Eve→Bob : Ticket ‖ E(Ka,b , N2)

Bob →Eve: E(Ka,b , N2 - 1, N3)

Eve →Bob: E(Ka,b , N3 - 1)

Afterwards, Eve can use Ka,b for secure communication.

*** If there is no mention about replay of ticket or session key Ka,b , 1pt is deducted*

2. The tickets can include a timestamp so that their validity could be verified before each session started. This is one of the causes for the addition of timestamp in Kerberos.

** Gave full point for answers using nonce instead of timestamp

** if the answer does not clearly mention that timestamp (nonce) is in ticket, deducted 1 pt.

** Use of DH key exchange or public key crypto would lose the point of using KDC. So subtracted 2pt.

7. *** No point is given if Yes/No is wrong. If it is correct but if description is wrong, max 2 points are deducted*

1. The attack is not possible when CBC mode encryption is used. CBC mode encryption alone does not offer integrity protection, thus manipulation on the network is possible. However, what an attacker needs to do is mount meaningful manipulation of the 2nd block of the cipher text so that Alice's IP address is replaced with Eve's IP address. To do so, Eve needs to tweak the 1st block of the cipher text. However, if Eve does it, the first block is not correctly decrypted to "Alice", and thus Eve cannot accomplish her goal.

*** Padding Oracle Attack is highly dependent on bad implementation of decryption module. Thus, no point is given.*

2. On the other hand if CTR mode is used, Eve can be successful in the attack. Knowing the plain text format and Alice's IP address (e.g., by means of network sniffing), Eve can flip bits in the 2nd block of the cipher text so that the decrypted IP address becomes Eve's IP. The first block is intact and thus will be decrypted to "Alice".
3. By adding message authentication code or using authenticated encryption (e.g., AES GCM mode), the attack can be prevented since the receiver can detect the message manipulation.

8.

The malicious C can collude with a monitor X (or C itself may run monitor X). This is possible because anybody can be a monitor in Certificate Transparency ecosystem.

C generates a second, misissued, certificate for D, submits it to CT, and receives an SCT for it. Then, C notify X without delay and tell X to report it to InsuredCert. Because X is the first person to learn about the misissued certificate and its SCT directly from C, no one else can report this misissued certificate *before* the colluding X. InsuredCert's policy states that InsuredCert pays $1500 to X. (and X can provide majority of the money to C).

***Answers, such as waiting expiration of policy only got 5 points.*

9. *** No point is given if Yes/No is wrong. If description alone is wrong, 1.5 point is deducted.*

(1) No. There is no randomness involved. This always results in the same number.
(2) Yes. As the secret K is incorporated, it becomes impossible for an attacker to anticipate the hash value. Also note that the timestamp changes and thus ISN also changes.
(3) No. In this case, even though the hash value varies with the timestamp, an attacker can still easily view or anticipate all the information used and thus it is feasible for an attacker to calculate the valid ISN.
[4] No. While secret key K is used as a seed, this results in the same sequence number always.
[5] Yes. As K serves as the secret key, it becomes impossible for an attacker to anticipate the encrypted value even if counter value is predictable.

10. *** No point is given if Yes/No is wrong. If description alone is wrong, 1.5 point is deducted.*

a. 1. No. M is unable to counterfeit A's digital signature. Consequently, it is not feasible for M to accomplish this even if they have access to the packet transmitted by B.

a.2. Yes. This attack works, since the shut-off message sent by A includes hash value of a packet that were actually sent by B and also the message is signed by A. Thus, the shut-off message is accepted by B's machine and future message to A will be filtered.

b.1. Yes. S needs to check the validity of overwhelming amount of fake. shut-off messages, which can exhaust the resource of S, preventing it from handling other legitimate task. Note that verification requires computationally heavy signature verification.

b.2. Yes. This attack is effective, and solely relying on AIP does not prevent the generation of counterfeit EIDs. It is worth noting that the paper proposed some mitigating measures, such as restricting the number of new EIDs generated per unit time, which is verified by an immediate router. If such an additional mechanism is put in place, the attack can be alleviated.

b.3. Yes, this attack works. As the attack packets are not received by S, it becomes impossible for S to transmit a shut-off message. This attack resembles the Crossfire attack.

11.

Below are some example answers, but not limited to these.

- Run more honeypot instances.
    - Just as attempted in CAUDIT paper, more instances will result in more chance to get found by attackers.
- Advertise the SSH honeypot to the Internet (e.g., some web site on NUS, Github, online forum, underground web site, etc.) with plausible, attractive description about the server.
    - Attackers may find it on the internet search. If the description is realistic and says something valuable, it would better attract attackers.
- Wait a few more days to get indexed by Shodan.
    - Attackers are watching Shodan search engine to find attractive target.
- Together with the one above, configure Cowrie honeypot to change its banner (e.g., version number information) to make it look outdated or vulnerable (e.g., based on information CVE database).
    - If attacker finds that the SSH server is running vulnerable SSH server, more likely he attempt access to it.