A-3 Why is it natural to think of the communication channel as a bulletin board? A bulletin board can be seen as a broadcast channel with memory. Anyone can write to their own section of the bulletin board and anyone can read everything that is posted on the board. But the difference is that you cant delete or alter anything on the board. If we relate the bulletin board to to participants in the voting scheme, the participants can write to the board and everyone can read the board, and it is not anonymous.

A-4 Why is the homomorphic property of ElGamal encryption not really suitable in an electronic voting system based on homomorphic encryption? The homomorphic property of ElGamal allows ciphertexts to be multiplied resulting in an encryption of the product of the two plaintexts. This is not very useful since votes are summed up to get the final result. This could be solved by using a variant of ElGamal encryption where (a, b) = (g^r, $w^m y^r$) (instead of (a, b) = (g^r, m^{y^r})).Then the homomorphic property will be E(m1, r1)E(m2, r2) = (a1a2, b1b2) = (g^{r1+r2} , w^{m1+m2} y^{r1+r2}) = E(m1+m2, r1+r2). One problem with this is that when decrypting the end result, one obtains g^{a+b}. In order to obtain the actual value of a + b, one would have to solve the Discrete Log problem.

A-6 Consider the zero-knowledge proof in Figure 1 of the lecture notes. Show the special soundness property, i.e., given two transcripts of the protocol (a′, b′, c, r) and (a′, b′, c′, r′), show that it is possible to recover s. Since $h = g^s$ and the verifier gives the prover c which computes $r = w + sc$ then then the verifier checks that $g^r = \hat{a}h^c$. If this is run again then with the new given $\hat{c}$ then the new r will be $r = \hat{w} + s\hat{c}$ then we put $g^r = g^{\hat{r}} \Rightarrow g^{r-\hat{r}} = h^{c-\hat{c}}$ and if you substitute in $h = g^s$ in the equation then you get $g^{r-\hat{r}} = g^{s(c-\hat{c})}$ and then you get that $r - \hat{r} = s(c - \hat{c}) \Rightarrow s = \frac{r-\hat{r}}{c-\hat{c}}$ And since all the values are known then you can extract the secret s.

A-13 Describe two different usages of secret sharing, one where the secret is reconstructed "explicitly", and one where it is not. Decentralized voting is one usage of the secret sharing, suppose a community to perform an election, but they want to ensure that the vote-counters won't lie about the results. Using a homomorphic secret sharing known as Shamir's secret sharing, each member can put its vote into a form that can be split among the counters, then when the voting is over the counters combine the pieces which allows them to reverse the alteration process and to recover the aggregate election results. And since the pieces is designed so the voters cant know how an alteration will affect the whole vote they cant cheat.

Another usage of the secret sharing is problems like the Yao's Millionaires problem, it is a secure multi-party computation problem, the problem discusses two millionaires, Alice and Bob. Who are interested in finding out who is richer, but they do not want to disclose how much money they got. It is analogous to a more problem where there are two numbers a and b, and the goal is to solve the inequality $a \geq b$ without revealing the actual values of a and b.

A-16 In the slides, two main strategies for making an electronic voting scheme are presented. One is that "the vote is posted on the bulletin board in clear text, but the person casting the vote is anonymous". Describe a scheme like this, and in particular explain why this scheme still ensures "one-voter-one-vote". The Mix Network scheme does this, it ensures that the voter is anonymous but the vote can be seen clearly on the bulletin board. The registration phase, the first mix with the public key k_n checks the real identity of the sender to verify the real identity and check so that is an eligible voter before forwarding the message. And the last Mix will then output a batch of digital pseudonyms (PK_1, pk_2). The voter then sends $K_n(R_n, K_{n-1}(..., K_2(R_2, K_1(R_1, PK, V, \sigma(V)))...))$ through the Mixes and the last Mix can post $PK, V, \sigma(V)$ publicly. The important part is that Mixes secretly permute the list of messages. The

final Mix will post the pseudonyms together with the votes. Now each voter can check that each pseudonym has voted and it can also check the vote and verify the result.So since they public key is posted as pseudonym for the voter there can only be one-voter-one-vote, unless both the Mix and the voter is in cahoots. This way everyone can check so that each pseudonym has voted, and can check the vote and verify the result.

A-18 Why does the blind signature based scheme preserve privacy even if the administrator and the counter cooperate? Even if the administrator and the counter cooperate, the privacy is still ensured due to the blind signature. The voter commits to a vote by computing $x_i = \xi(v_i, k_i)$. This will be the voter's ballot. Then the voter computes a blinded value of the commitment $ei = \chi(x_i, r_i)$ and signs this with the private key, the administrator then verifies the signature, identifies the voter and checks that the voter is eligible to vote and have not applied before, then signs the value. The administrator then publishes a list of ID_i, e_i, s_i on a bulletin board, and the voter can at this point, knowing the signature on e_i, and the value r_i, compute A's signature on the ballot $xi, yi = \sigma A(xi)$. Each voter then sends the signed commitment to the counter. Which means that even though the counter and the administrator works together they cant extract the identity of the voter since the voter does not send in the same thing that got signed by the authenticator, the voter instead sends their ballot with the authenticators signature on.

A-19 Consider the blind signature based protocol. No result will be published before all voters has had the chance to verify that their vote is indeed correct. How is this important property achieved? As mentioned in the answer before, this whole scheme relies on that only the voter knows their vote on the published list. So everyone can see the list that the administrator publishes with accepted voters and commitments, and everyone can check so they are included in the list. But it is not possible for either the voter to change their mind since it is published, or anyone else to see what other has voted due to the commitment. Each voter then sends the signed commitment to the counter C, and remember the commitment is signed by the administrator and not the voter. C extracts and verifies the signature and publishes l_i, x_i, y_i on the bulletin board, l_i is just an integer identifying the ballot. And at this point everyone can compare the two lists, and see if their vote is correctly cast. So if the voter does not agree with what is on the bulletin board the voter can by release r_i so everyone can verify that the voters ballot was not received. Unless there is an discrepancy the voter V_i then sends (l_i, k_i) through an anonymous communication channel, using k_i it is possible to open the commitment and retrieve the vote identified by l_i.

A-21 Compare the fairness provided by the Mix-based and blind signature-based protocols given in the lecture notes. Fairness: No partial results should be disclosed before the end of the voting procedure.
In the blind signature scheme, as previously mentioned in the questions above the information is put on a bulletin board, but the information does not disclose what the the voters have voted, but the voters are able to verify that their votes are indeed correct. In the Mix network scheme everyone is anonymous but the votes can be seen. The last Mix will output a batch of digital pseudonyms (PK_1, PK_2, . . .) for all eligible voters, which will be posted publicly. Upon voting each user sends $K_n(R_n, K_{n-1}(..., K_2(R_2, K_1(R_1, PK, V, \sigma(V)))...))$ through the Mixes and the last Mix can post $PK, V, \sigma(V)$ publicly. So the problem here for Mix Networks is if the result is corrupted in any way, then the result can not be trusted, even if its only one Mix that is corrupt and it changes the encryption, then there will be an error in the posted result in the end. And since the result is corrupted then they must hold a re-election which could affect the results. And this takes away the fairness. But for blind signatures, it is not possible to extract any result before all voters have had a chance to verify their result. This does not give away the result of the rest of the votes, so it keeps the Fairness even if there is a few votes which is corrupted.