Advanced Web Security - Assignment 3A

November 2020

Questions

1. In the blind signature based eVoting Protocol presented in the lecture, voters sign their vote (step 3). Why? Can you describe an attack if votes were not signed?

2. Give an example of an electronic voting scheme that provides robustness. Describe how robustness is achieved.

3. The lectures present 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”.

4. Why is it not possible to support write-in voting in an electronic voting system based on homomorphic encryption?

5. In the eVoting protocol based on blind signatures, assume we are using the RSA blind signature from Lecture 1. If a voter cannot find their commitment in the published list, they reveal r, so that e can be derived and identified in the list published by the administrator. This does not reveal their vote. What property of the commitment scheme is crucial here?

6. Perform Lagrange interpolation to find f(0) when f(1) = 0, f(2) = 2, f(3) = 4 and f(4) = 10. The degree of the polynomial f(x) is 3.

7. Describe two different usages of secret sharing, one where the secret is reconstructed “explicitly”, and one where it is not.

8. Explain how the zero-knowledge property of a zero-knowledge proof is related to a simulator.

Answers

1. The voters sign their vote so that the administrator can prove their identity and eligibility to the administrator and to stop the voter from voting more than one time. We want to show that we are allowed to vote.
   An attack that can be used when the votes are not signed is where one voter can vote as someone else, or vote multiple times because it is never verified who the voter is and if they have voted already.

2. A e-voting scheme that provides robustness is the threshold scheme using homomorphic encryption. The robustness is achieved through the facts that you need a set amount of users to find the secret, but not everyone. So if one user looses their key, or secret, then we are still able to retrieve the secret because we only need a threshold amount of users to find the "code" or "key". As long as we have at least t authorities that are honest, the result will still be the same.

3. A scheme like that is a commitment scheme using blind signature. Each voter chooses their vote, commits their vote, then computes a blinded value of the commitment and signs this value using a private key. Now an administrator checks the signature, this is to make sure that the voter is eligible and that they haven't submitted a vote earlier. If everything is OK, then the administrator signs the blinded commitment. Now the Administrator doesn't know the vote of the user, but only that a user has applied to vote. The user will now use the blinded signature together with the commitment and sent it to the ballot for approval.
   This scheme ensure "one-voter-one-vote" because of the way the blind signature works. Before the voter is allowed to vote, they need to prove their identity to an administrator. Then they need to provide a blinded version of their vote, which is signed. And as we know from before, using the multiplicative properties of RSA in our blind signature, a blinded signature is still valid for the real value of the committed vote.

4. It is not possible to support write-in voting in e-voting systems based on homomorphic encryption because this type of encryption is malleable, which means that it will be able to infer some information about from other cipher-text about a challenge cipher-texts. The main reason why write-in ballots aren't possible with homomorphic encryption, is with the way we "tally" up the votes. When we tally the votes we have an idea of what the different answers could be, which is important to a degree, because we are adding together encrypted votes to create a result. Since write-in ballots have to each be read specifically, that makes it not possible to use homomorphic encryption from write-in ballots.

5. The property that is crucial here the blinding part. When the administrator signs the blinded commit, the signature being used later does not reveal any information about the voter. Their signature can not be linked

to their vote being published by the counter or any of the other properties published by the voter.

6. $f(0)$ is equal to -6.

$$f(x) = \left(\frac{2}{3}(x-3)(x-2)+2\right)(x-1)$$

$$f(x) = \frac{2x^3}{3} - 4x^2 + \frac{28x}{3} - 6$$

   Which means when we make x = 0, then all the fractions with x in it, will be equal to zero, and we are left with -6.

7. The first usage of secret sharing, being one where the secret is reconstructed "explicitly" is Shamir's secret sharing scheme, here we have one authority who creates a shared secret, then distributes n secrets to the participants before hopefully destroying the secret. To be able to reconstruct the secret, we need t participants in order to find the secret.

   A usage where it is not explicit, is the extension of Shamir's scheme, where the secret is agreed upon by all the participants, not having one authority in charge of creating the shared secret. This means that if one authority is corrupt, then they will not be able to decrypt the secret without having at least t participants joining in.

8. The zero-knowledge property is related to the existence of a simulator that simulates the steps of the proof. Basically, if Victor can produce a transcript of the communication that can't be distinguished from an actual transcript, then Victor will not have learned anything about the proof.

   The property is related to the fact that by the end of a proof, the verified should have no knowledge regarding the actual information being proved, other than the fact that it is correct.