*Note: accuracy of these solutions cannot be guaranteed – feel free to comment or fix any errors you may see. Also, it is recommended to use the desktop version of Word due to the equations (from the web version, File -> Info).*

## Question 1

### Part a

*The question has a typo in that the sum of the probabilities of finding each plaintext character is not 1, but the impact is very minor in practice.*

Standard calculations:

### Part b

Use the conditional probability formula:

Notice that

Hence,

To get the other side, rewrite the conditional probability formula on the other side:

and are both known. Hence the result is simply .

### Part c

No. . Also, by Definition 10 in the notes, , which is enough to disprove perfect security. Yet another approach is to appeal to Shannon’s notion of perfect security (fails the second condition).

### Part d

1. **False**. That’s only part of the proof. You need for instance, and Shannon’s notion also requires the presence of a unique *k* such that . There are other possible answers.  
     
   Note that the answer is still false if the statement is taken as a “necessary” condition – the below EdSTEM post explains why:  
     
   Text

   Description automatically generated
2. **False**. The modified shift cipher (Example 13 in the lecture notes) is another example.
3. **True** (*not 100% sure, especially on the explanation)*. The one-time pad was patented in the US.
4. **False**. Disadvantages including needing the key to be as long as the message, and the fact that the key is essentially ephemeral.
5. **False** (*could have misunderstood this one though*). Just having a random number generator does not guarantee security in any form – conditions such as in d iv) must be satisfied. Also, OFB isn’t secure despite being pseudo-random (note: that isn’t the same as being perfectly random due to the former technically being deterministic).

## Question 2

### Part a

as that’s the secret ().

The remaining calculations are shown below:

Background pattern

Description automatically generated

Wolfram Alpha can be used, like so.

Graphical user interface, application

Description automatically generated

### Part b

Note that

Graphical user interface, application, Word

Description automatically generated

Applying that to the problem:

Now, we need to find the secret.

This can be done using the relevant formula:

A picture containing schematic

Description automatically generated

We do that:

The secret is hence recovered.

For the last part, use the Berlekamp-Welch algorithm (*not covered for 2021-22*).

* Logically, no erasures can be tolerated as we need at least 4 shares to recover the secret for a polynomial f of degree 3, and c only contains 4 shares.

### Part c

1. **True**. It is information-theoretic secure after all, which is stronger.
   1. Would be inclined to say this is **False**, as it is information-theoretically secure.
   2. Would be inclined to say it's true as it is true if its information-theoretically secure
2. **True**. This can be shown by Lagrange interpolation – nothing new can be learned about the secret *s*. See Theorem 26 of the lecture notes.
3. **False**. The trusted dealer needs to know the secret.
4. **False**. The recombinant vector does not depend on the secret or *f*, hence it can be reused.
5. **True**. Notice that it is possible to prevent any less than *t* users from getting the secret in Shamir Secret Sharing Schemes, which can be applied in this case.

## Question 3

## Part a

ai) We can use ElGamal Commitment Scheme. The chose p should be a safe prime, i.e. p = 2q + 1.

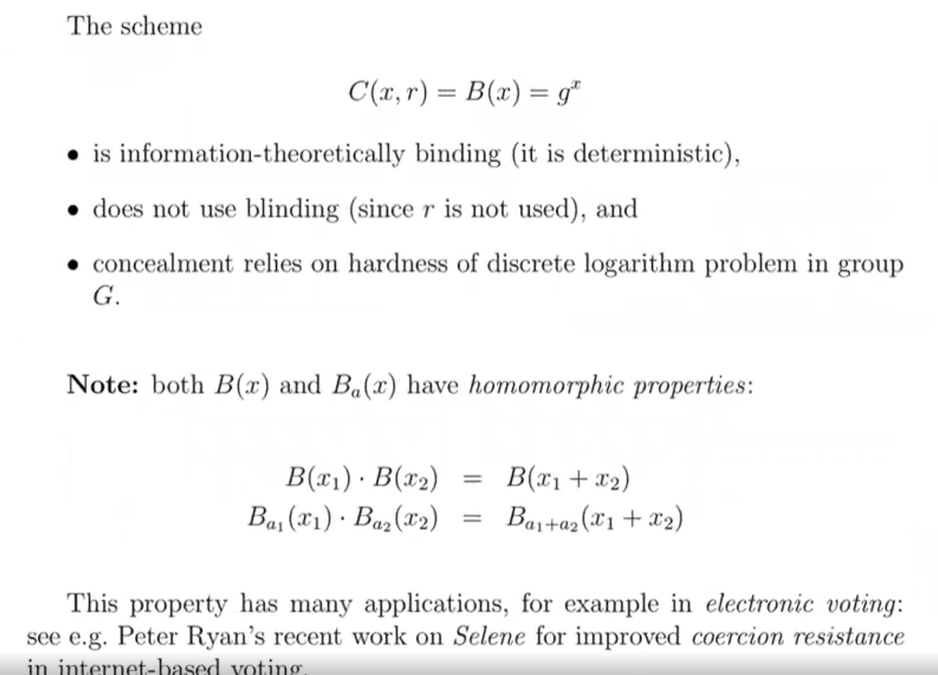
ii) Depending on the choice of commitment scheme in the first subsection, the answer may vary – but it is otherwise not different from an ElGamal Commitment Scheme as the idea is same – Alice and Bob want to confirm that the other actually committed whatever they chose. Similar to the rock-paper-scissor example given in the lecture slides.

### Part b

1. ElGamal: Information-theoretically binding, computationally concealing
2. Cannot learn anything from transcript. (Like zero-knowledge) One can generate x\_a then r then C(x\_a, r). so from the history cannot gain any advantage

### Part c

1. No, they can be either information-theoretically binding and computationally concealing or computationally binding and information-theoretically concealing.
2. No, from Lecture 39 (slide 138) at 8:21, this scheme does not use random variable ***r***



1. False, commitment schemes only provide a method for parties to avoid having their committed value revealed before they do so themselves, as well as prevent parties from repudiating their committed values once they have been committed. As such they do nothing to improve accountability since they offer no method of tying the commitment to a particular user (vs e.g. MACs).
2. True, (cryptographically secure) ECs could be used in place of hash functions to compute commitments. Security of this scheme relies on hardness of EC DLP.
3. False, as information-theoretic binding already assumes the attacker has unlimited computational resources.