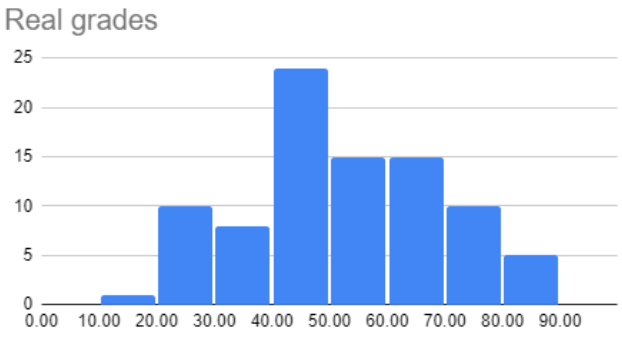
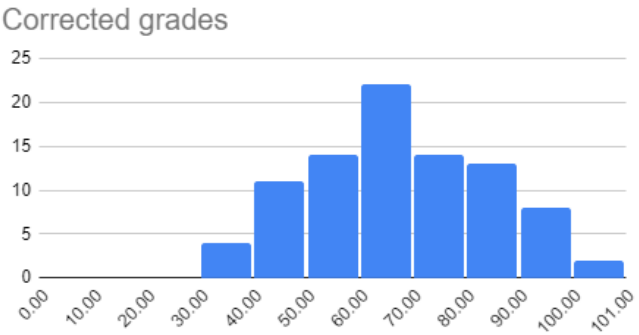
CS-523 Midterm Most Repeated Errors

Spring 2023





The bin 100-101 contains people with 100 points

General advice

1. **Answer (all parts of) the question**. A large number of points are lost because either:
    a. some parts of a question are not answered (e.g., not giving a concern, not giving an alternative solution),
    b. The answer ignores constraints in the question (e.g., answering about a different adversary model than indicated in the question)
    c. the answer completely misses the question (e.g., answering with a method instead of a legal basis or answering with an attack instead of providing a privacy concern).
2. **Do not write an attack when asked for a threat model.** A threat model includes the background knowledge of the adversary, their capabilities, and their goal. Providing an attack not only does not answer the question, but its description may lack some of these details.
3. **Do not paraphrase the slides (or the question).** When you paraphrase the slides without applying the concepts to the problem at hand, there is no way for us to evaluate whether you understand the concepts. Thus we cannot give points. When you paraphrase the question is the same (and typically is also not an answer as the information is already given to you)

Question 1: Privacy-preserving Data Publishing

Part 1 (Carmela)

- **Providing an attack instead of a threat model.** The question asked for a threat model, asking explicitly to describe the “background information, capabilities, and goal”. A large percentage of students described an attack that would materialise the concern but not really the elements that characterise a threat model as required by the question. Answers presenting an attack only received points if the elements of the threat model appeared in some form. If the elements did not appear points were reduced.
- **Answering the question partially**. Full points were not given in cases where the answer did not mention all elements asked in the question, e.g., the privacy concern is not made explicit, or the background information and capabilities of the adversary do not appear in any form.
- **Saying reidentification is a concern without a justification.** Reidentification, as in knowing that a user is in a database, is only a privacy concern if it leads to a privacy loss. Answers that did not link reidentification to a loss of privacy (e.g., learning dietary restrictions) did not receive points.
- **Including the private information to be learned in the threat model.** Many answers argued that an adversary that can observe the user browsing pages, or can link comments in the web and the anonymized table, can use that information to reveal their dietary restrictions. If the adversary already knows which pages you browse, or you commented on, they already know your restrictions.
  Points were given if the answer would explicitly say that the restriction was not learned from the browsed/commented page, but because other entries in the anonymized database linked to the same individual would reveal the dietary restriction.

Part 2 (Klim)

- **Set of quasi-identifers does not correspond to the threat model.** Many answers listed quasi-identifiers which are not accessible for an adversary within the threat model listed in Part 1. For example, writing in the threat model in part 1 that the adversary has access to the name of the recipe accessed by their target (field “recipe” in the table); and then in part 2 the student listed day and comments as quasi-identifiers. This answer is incorrect since the adversary cannot reidentify targets with features hidden from them (day, comments).
  A particular case of this issue were answers that listed all the columns as quasi-identifiers. While this may be true in practice, and it is part of the learning of the lecture, the midterm question specifies “against the adversary you describe there [in part 1].” If the adversary you describe in Part 1 only has access to part of the columns, the other columns cannot be quasi-identifiers.

Part 3 (Klim)

- **Providing an attack in Part 1 setting instead of an alternative threat model.** This error is similar to the error in Part 1. In some works instead of providing a new threat model, an attack for the old threat model is listed. These answers do not match the question and were not given points.

Part 4 (Carmela)

- **Using consent as a legal basis.** Note that BecomeAChef decides to open this API "After seeing the success of their newspaper campaign". Therefore, the data already existed and they cannot have the consent to use the data in this manner.
- **Including assumptions about user behaviour in the threat model.** Assumptions about user behaviour may determine whether an attack succeeds or not. But they do not have a place in a threat model where we indicate the capabilities, background information, and goals of the adversary.
- **Privacy concerns that are not from individuals.** Many answers proposed as concerns inferences about recipes, or groups. If these inferences cannot be linked back to individuals, cannot be classified as privacy concerns. Statistics about recipes, in particular, are at most a business secret for BecomeAChef, but have no privacy implications.
- **Assign sensitivity one to the query**. Many answers said that sensitivity is one because users can visit a page or not. The query, however, is not about visits to a given page but about the number of pages. Thus, a user can affect the result in the number of pages visited. We gave points to answers regardless of the assumptions they made (all recipes in the website, at most 3 recipes per day, etc).
- **Not giving details about the algorithm**. The question asked specifically for the details required to implement the system. Some answers did not provide any detail (nor sensitivity, nor epsilon usage), and many did not provide any detail on how epsilon would be distributed across queries.

Question 2: Privacy-preserving Encryption

Part 1 (Simone)

- **Missing Triserve owns all servers.** Many answers said the system is secure because computation happens in three servers in different jurisdictions. While this is true, the three servers are owned by the same entity: Triserve. Therefore Triserve has access to all the shares in these servers. As we are in the honest but curious threat model, which states that parties try to learn as much as possible from the inputs they have access to, the system is insecure as Triserve can reconstruct the votes from the shares they can access.

- **Relying on data-protection laws to prevent Triserve from learning which professors each student voted for.** The question asked about the capability of actors to learn (“ensure that no actor in this system can know"). Data protection laws

can establish penalties for misusing data, but they cannot eliminate actors' capabilities of learning from those data. As Triserve has access to all the shares and the students that send them (because it controls all servers), data protection laws cannot prevent Triserve from learning which students voted for which professor.

Part 2 (Simone)

- **Wrong SMC protocol.** The question explicitly asks to present a SMC protocol to “learn how many students voted for the same two professors”. A protocol that privately computes the best professor according to the students of the university does not answer the question.
- **Privacy violations.** Disclosing the vote of one or more students is incompatible with the requirements of the question which explicitly states "without anyone having to reveal their full list of three votes in the clear", even if an honest-but-curious threat model is specified. (See the comment about the HbC threat model in Part 1.)
- **Imprecise circuit description.** The question clearly asks to specify the inputs to the circuit(s), how these inputs are computed (if needed) by the different parties, the output of the circuit(s) and the final output of the protocol. Not giving these information resulted in points deduction.
- **Missing threat model.** The question explicitly asks to give the threat model for the protocol. Not providing a threat model resulted in points deduction.

Part 3 (Neelu)

- **Not answering the question**. The question asked whether it is possible to compute the result in a reasonable time.Points were deducted for answers that do not mention a feasible solution or debate about the definition of a reasonable time.
- **Not thinking about whether the suggested solution is needed at all.** Points were deducted for answers that don’t consider whether multiplication (which is the root cause of making the computation time unreasonable) is needed or not to perform the required computation using Hompute.
- **Speaking about SMC in the answer**. The question is explicit that the Hompute service discussed for this part is based on FHE. This is unrelated to the SMC mechanisms discussed in the prior parts. Any answer referring to SMC was not accepted as it is not relevant to the question.
- **Not answering the full question**. Points were deducted for those answers which did not state any assumptions on capabilities of the actors explicitly asked in the question.

Part 4 (Carmela)

- **Paraphrasing slides without specifying the link to the question context**. Many answered generic statements about SMC or FHE that could apply to any system

based on those cryptographic primitives. If the answer does not provide any relation to the question it is not possible to evaluate that the concepts have been understood. These answers have been given 0 points.

- **Not fully answering the questions.** Many students only provided one threat. The question asked for two threats. Not answering part of a question results in points reduction.

Question 3: Anonymous authentication

Part 1 (Mathilde)

- **Repeating the question statement without new information.** For example "*all attributes are revealed hence the guard can track visitors*" is not a justification as it only paraphrases the question statement without explaining why (equivalently how) the guard can track users given these attributes.
- **Considering only the properties of the ABC scheme (i.e., unforgeability) when arguing about (in)security of the scheme and forgetting to take into account the application level details: the choice of attributes and their values. As a consequence, these answers wrongly agreed with the claim of the student that balelec cannot track students.** The scheme is actually leaky compared to the expected functionality. However, this is not because of the "tool" (i.e., using ABC), but how these credentials are built and used. More precisely, the attributes that are revealed act as a unique identifier of each student and can be used to track them.
- **Changing the setup of the question.** Some answers included claims such as "only the *stages_allowed* attribute is revealed". The question explicitly states that all attributes are revealed. Answers to modifications of our questions did not receive full points.

Part 2 (Mathilde)

- **Revealing more than necessary and not minimizing the leakage of the scheme.** Some answers propose to hide the *time_of_purchase* and *faculty* attributes and reveal *stages_allowed*. Revealing the whole set of stages the student is allowed to go to is actually more information than what is minimally required to maintain functionality. It is enough for the student to prove that they have access to the stage they are in front of, through a ZKP (as hinted).
- **Being imprecise when describing the ZKP to be implemented.** For example, "The student needs to show that the scene is in *stages_alowed."* did not receive full points since the zero-knowledge dimension is completely missing. For this question, using the ZKP shorthand notation was a precise and concise way to describe the ZKP you are thinking of, since it contains all the needed details for the implementation.

Part 3 (Neelu)

- **Assuming that the verifier stores the revealed attributes** and they can be used to prevent duplicates i.e. prevent multiple use of same credentials. Though the direction is correct, such answers don't explain under which conditions the concern of ticket reuse materialises. Partial points were deducted for such cases.
- **Not discussing how the verifier performs blacklisting.** If the verifier is checking against a nonce/random ID, the verifier needs to store them. Answers mentioning that the verifier has a good memory without details about how this "memory" could be implemented to remember these were not accepted.
- **Limited amount of tokens or limited number of entries regardless of stage**. With a limited number of entries to stages per credential or getting tokens for each stage the user paid for, if the credential allows multiple entries per stage or the tokens are not specific to stages, multiple people can enter one stage at the same time.
- **Unforgeability - confusing duplicating credentials with forging credentials.** The unforgeability property refers to not being able to produce valid credentials without interacting with an issuer. As a side note, duplicating a credential does not mean it becomes invalid, as the signature would be the same (and valid), not violating unforgeability. To use such a duplicate credential, though, students would need to share their secret key.

Part 4 (Neelu)

- **Implicitly assuming that the verifier is malicious and colluding with the issuer.** There is nothing mentioned in the question about the verifier's honesty. If there is no explicit assumption on this, it cannot be considered that the verifier is honest and will provide the issuer with information to track users. Answers that explicitly stated the assumption that the verifier is malicious were given full points.
- **Solutions that break the functionality of the system.** Solutions like not giving access to students for the stages they paid for in the credential, so they come to the issuer to complain during the event, which break functionality of the system (which is to allow students to go to concerts). Solutions that destroy functionality --or affect it in a significant manner-- cannot be considered solutions to the problem.
- **Using different keys per user**. Trying to verify on the spot using hundreds of keys, for one student to enter is may not be efficient. It would result in very slow checks that would have a strong impact on the experience of the festival attendees, to the point of making it not work (attendees could miss concerts). Thus, it becomes a particular case of a "solution that breaks the functionality of the system". Points were not deducted when the answers made it clear that this solution slows down credential checking and has an impact on utility.
- **Solutions not related to ABCs.** The question explicitly asks about a solution **"**given your proposal in Part 2". Solutions like attaching a physical tracker to every ticket, or the ones that talk about scanners with red/green light or physical ticket checking machines which are not related to ABCs (and thus not to your ABC proposal in part 2) were not accepted. (We also note that even if relation to Part 2 was not explicitly required, some of these solutions are impractical, and we would not accept them without an explanation of how they could be implemented that is plausible)

- **Not providing enough details or justification for the proposed answer.** Stating that the issuer cannot track users without precisely justifying what prevents the issuer from tracking users. For example, stating that the issuer can keep a copy of credentials or certain attributes from the credential that they issued, but not explaining how they use this information to track users.