CS4501 Cryptographic Protocols, Homework 1

Response by: Your Name, (Computing ID)

Total points: 40 awarded maximum. 50 available. Points are noted after each problem.

Instructions. For each problem, typeset your solution in the `answer` environment, and if there are sub-problems, mark them clearly. Feel free to use as much space as you require, and be sure to update your name and computing ID above, and the acknowledgments box at the end.

Policies. In short, you are encouraged to think about the problems on your own, and then discuss them and work toward solutions with your classmates. You must write and submit your own solutions. You may also read any published material that helps you come to an understanding of the problems, but you must acknowledge and/or reference any discussion or published material, with the exception of lecture notes and other official class materials, in-class and in-office-hours discussions, and basic LaTeX help or dictionary lookups. It is a violation of the honor code if any of the following occur:

- You copy text directly from any source.
- You use any material or discussion without acknowledgment or citation, excluding the above special cases.
- You are unable to explain your work orally.

See https://jackdoerner.net/teaching/2026/Spring/CS4501/#policies for more details.

Notation. Recall that we use λ to represent the empty string and $\varnothing$ to represent the empty set.

Problem 1 (Low-Temperature Civics (5pts or 10pts))**.** For 5 points, choose an electoral system system (e.g. instant runoff, single transferrable vote, Dodgson's method, the electoral system used by some real-world polity, etc.) and design an ideal functionality that models this electoral system in an interaction with a set of *voters*. Try to ensure that you have no edge cases and that a leader is always elected.

If you think elections are boring when the winner is determined using a succinct and comprehensible set of rules, then you can receive 10 points by designing a functionality that specifically models the presidential election procedure of the United States, as specified in Appendix A.

Hint: For some electoral systems it might be sufficient to combine $\mathcal{F}_{\mathsf{SFE}}$ with some specific election function f_{election}. More complicated electoral systems (particularly those involving multiple rounds of voting) might require you to specify a *reactive* functionality that produces outputs or takes inputs at several different points during its operation.

Problem 2 (Three Easy Protocols (5pts each))**.**

1. Consider the 2-ary XOR function for bits, that is, $f_{\mathsf{xor}} : \{0,1\} \times \{0,1\} \to \{0,1\} \times \{0,1\}$ such that $f_{\mathsf{xor}}(x_1, x_2) = (x_1 \oplus x_2, x_1 \oplus x_2)$. Design a two-party protocol that perfectly securely computes f_{xor} in the presence of a semi-honest adversary that statically corrupts at most one party. Prove that your protocol is secure.

2. Consider the 2-ary coin tossing function $f_{\mathsf{ct}}(\lambda, \lambda) = (b, b)$, where $b \leftarrow \{0,1\}$. Design a two-party protocol that perfectly securely computes f_{ct} in the presence of a semi-honest adversary that statically corrupts at most one party. Prove that your protocol is secure.
3. Consider a function $f : \{0,1\}^* \times \varnothing \to \{0,1\}^* \times \{0,1\}^*$ such that $f(x, \lambda) = (g(x), h(x, g(x)))$ where $g(x)$ is some randomized process and h is deterministic. Prove that there exists a *one-message* protocol that perfectly securely computes any such f in the presence of a semi-honest adversary that statically corrupts at most one party.

Hint: All of these sub-problems specify a semi-honest adversary, so we can use the simplified definitions from Lecture 3. f_{xor} is deterministic, so the "even simpler" definition for deterministic functionalities will do, but the other two functions are randomized, so it will be necessary to consider the joint distribution of outputs and corrupt party views.

Problem 3 (How to Share a Coke (5pts))**.** Consider a multi-national soft-drink manufacturer in which the CEO, two VPs, and five executives are responsible for a secret recipe of immense monetary value. We require that none of these individuals can access the information on their own, nor can a VP and an executive access it together, nor can any four executives access it. However,

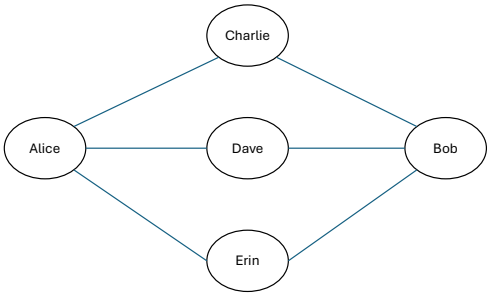
- The CEO should be able to access the information by working together with anyone else
- A VP should be able to access the information by working together with the CEO, the second VP, or any two of the other executives
- All five executives together should be able to access the information

Design a secret sharing scheme for the company that is based upon Shamir's scheme. That is, describe the Share and Recon algorithms, and prove that they have the properties of correctness and privacy.

Hint: If you wish, you may invoke Shamir's scheme as a subroutine and show that any attack on your scheme implies an attack on Shamir's (that is, write a *security reduction*). You may also use the principles of Shamir's scheme without invoking it as a subroutine, and prove the security of your scheme directly.

Problem 4 (Lagrange Interpolation (5pts))**.** Consider the field $\mathbb{F}_7$. Suppose that a secret $s \in \mathbb{F}_7$ is shared among three parties using Shamir's scheme for $t = 2$. Alice receives the point $(1, 4)$, Bob receives the point $(2, 1)$ and Charlie receives the point $(3, 5)$. All of them have sent their points to you. Use Lagrange interpolation to reconstruct the secret s.

Problem 5 (The Messengers (5pts each))**.** Consider the following communication graph, where each of Alice and Bob (P_1 and P_2, respectively) can communicate with Charlie, Dave, and Erin, (P_3, P_4, and P_5, respectively) but no other communication lines are present.



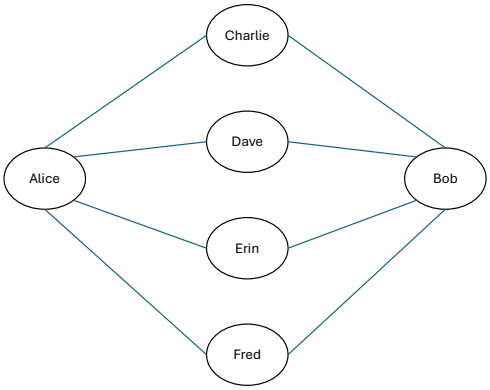
1. Alice would like to send a private message of ℓ bits to Bob, without Charlie, Dave, and Erin learning anything but its length. Define a 5-ary *secure message transmission* (SMT) function for this task.

2. Alice knows that Charlie, Dave, and Erin will behave honestly and follow the protocol, but she suspects that two of them might collude and try to learn her message. She is quite certain though that at most two will collude. Design a protocol that is perfectly secure against any two semi-honest corruptions (out of the five parties) and prove that it securely computes the SMT function you described above.

   Hint: Since your SMT function should be deterministic, and the adversary is semi-honest, you can use the "even simpler" security definition.

3. Fred (P_6) is also willing to help Alice and Bob by communicating with the two of them, but not with Charlie, Dave, and Erin. Alice still suspects that two out of the four might collude. Furthermore, she's afraid that any one of the four might crash and fail to communicate with Bob.

   Adjust the SMT function to account for Fred, and then design a protocol that securely computes the new function in the presence of two semi-honest corruptions. Your protocol should *also* be able to resist a single crash of Charlie, Dave, Erin, or Fred. Prove that your protocol is secure.



Acknowledgments

In this box, you should acknowledge your collaborators and the resources you used, if any. For example:

Problem 1: I discussed this problem with Alice and Bob. In addition, I asked Carol for help understanding Conditional Probability, but we did not discuss the problem further.

Problem 2: I asked ChatGPT "What is a turing machine?", and it gave me the following transcript: https://chatgpt.com/share/68b4dba7-e19c-8013-a137-e8db901493b7.

Problem 3: It helped me to read the proof of [Vad12, Theorem X, Page Y].

Instructor's Acknowledgments

Problems in this homework have been borrowed from or inspired by a number of sources. Citations can be provided on request, after the homework is completed.

References

[Vad12] Salil P. Vadhan. *Pseudorandomness*. 2012. https://people.seas.harvard.edu/~salil/pseudorandomness/pseudorandomness-published-Dec12.pdf.

A The American Way to Choose a President

In the United States, the president is decided using the following procedure:[1]

1. Members of the *electoral college* each cast a single vote for the candidate of their choice.
2. The *vice president* opens the ballots, and announces the contents of each individual ballot to the *joint session of congress* (that is, the combined *house of representatives* and *senate*). If at least one representative *and* at least one senator object, then both houses must vote on whether to count the ballot. If *either* a majority of representatives *or* a majority of senators vote that the ballot should be counted, then it is. Importantly, the vice president may *not* choose whether or not the ballots are opened and announced.[2]
3. After the ballots are counted, if any candidate receives a majority of the possible electoral votes,[3] then that candidate is declared to be the president.
4. If no candidate receives a majority of the possible electoral votes, a contingent election occurs:
   (a) In a contingent election, only the three presidential candidates who received the greatest numbers of electoral college votes in step 2 are eligible. Each representative casts a single ballot for the eligible candidate of their choice.

[1] Which has been simplified considerably to remove less-important participants, such as the *citizens*.
[2] That is, the VP's input is an empty string, but he or she must be present all the same.
[3] That is, a number of votes that is greater than or equal to half of the number of electors, as distinct from a majority of the votes that were actually counted.

(b) The ballots of the representatives are then grouped by *state*. If the majority of representatives from a particular state name any single candidate on their ballots, then one vote is added to the tally for that candidate in the contingent election. If no candidate is named on a majority of ballots from a particular state, then no vote is counted for that state.[4]

(c) If any single candidate receives a majority of possible state votes,[5] then that candidate is declared president.

(d) If no single candidate receives a majority of possible state votes, another contingent election begins, using the same procedure. The number of contingent elections is not necessarily bounded.

There are always at least two candidates. Each state has exactly two senators and at least one representative. For every representative there is one exactly elector,[6] and for every senator there is exactly one elector. To model this process, your ideal functionality must interact with the representatives, senators, electors, and VP. Make sure that it knows unambiguously with whom each interaction occurs, and that it works for any number of participants and candidates that satisfy the above criteria.

Optional (no points): Suppose an adversary wishes to prevent a president from being chosen *forever*. How many participants of each type does the adversary need to control in order to ensure that this occurs?

[4] In other words, each state gets (at most) one vote, which is determined by a vote among the representatives from that state.

[5] Again, a number of votes no less than half the number of states, as distinct from a majority of state votes actually counted.

[6] That is, one member of the electoral college.