6.5620 (6.875), Fall 2022

Homework # 6 **Due: Dec 7 2022, 11:59:59pm ET**

- **Typsetting:** You are encouraged to use LaTeX to typeset your solutions. You can use the following template.
- **Submissions:** Solutions should be submitted to Gradescope.
- **Reference your sources:** If you use material outside the class, please reference your sources (including papers, websites, wikipedia).
- **Acknowledge your collaborators:** Collaboration is permitted and encouraged in small groups of at most three. You must write up your solutions entirely on your own and acknowledge your collaborators.

Problems:

1. *(8 points)* **Solution-checking systems.**

   Please do *either* problem 1 *or* problem 5, and indicate which one you're doing on your problem set.

   Clint and Mallory are pset partners for 6.875. They enjoy collaborating on psets a lot, but they often work alone a bit before discussion, to test their individual problem-solving skills. Also, they are usually offline on Messenger, so they communicate in the non-interactive setting.

   Lest they spoil the fun of problem-solving, they agree to use *zero-knowledge solution-checking systems* when they try to convince each other that they have already discovered some solution w (i.e. a witness) to a problem $x \in L$ (L is an NP language) in the problem set.

   A *solution-checking system* is a tuple of p.p.t. algorithms $\Pi = (\mathsf{Setup}, P, V)$ with the following usage.

   (i) The trusted TAs compute $(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Setup}(1^n)$, send pp to Mallory, and send $(\mathsf{pp}, \mathsf{sp})$ to the Clint.

   (ii) Mallory discovers a witness w certifying that $x \in L$, and wants to prove to Clint that she knows w.

   (iii) Mallory computes $\pi \leftarrow P(1^n, \mathsf{pp}, x, w)$ and sends π to Clint.

   (iv) Clint accepts iff $V(1^n, \mathsf{sp}, x, \pi) = 1$. (We require that V computes a deterministic predicate.)

Note that the output of Setup does not depend on the statement $x \in L$ (nor w), as Setup was run by the trusted TAs before the problem sets were even written up. Also note that the system is designed to be used on a per-verifier basis, as sp is only sent to Clint but not Mallory (nor anyone else in the class). If Clint wanted to prove to Mallory that he found a solution, they would have to switch roles.

Assume we already have a tuple of p.p.t algorithms $\Pi_0 = (P_0^1, P_0^2, V_0)$ with the following usage.

(i) An honest prover has an input x, and a witness w certifying that $x \in L$.

(ii) The prover computes $\alpha \leftarrow P_0^1(1^n, x, w)$ and sends α to the verifier.

(iii) The verifier samples a random bit $\beta \leftarrow_R \{0, 1\}$ and sends β to the prover.

(iv) The prover computes $\gamma \leftarrow P_0^2(1^n, x, w, \alpha, \beta)$ and sends it to the verifier.

(v) The verifier accepts iff $V_0(1^n, \alpha, \beta, \gamma) = 1$, where V_0 is a deterministic predicate.

As a reminder, $|\alpha|, |\gamma| \in \text{poly}(n)$. Π_0 satisfies **perfect completeness**, **soundness with error** $1/2$, and **computational honest-verifier zero-knowledge** with simulator $\mathcal{S}$ (i.e., there exists PPT $\mathcal{S}$ such that $\{(\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}) : (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}) \leftarrow \mathcal{S}(1^n, x)\}$ is computationally indistinguishable from a transcript of a real execution between the honest verifier V and the honest prover P).

Such a protocol for any NP language can be constructed if we try hard enough to construct a zero-knowledge proof system for the NP-complete problem Hamiltonian Path, using ideas similar to the GMW protocol for 3-coloring. Fortunately, our friend Manuel open-sourced and provided for free such a protocol Π_0, so we can just use it for free.

Your task: Transform Π_0 into a solution-checking system Π, using any IND-CPA secure public-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, so that Clint and Mallory can show off to each other that they solved problems without spoiling the fun of problem-solving for the other person.

In particular, prove that your solution-checking system $\Pi = (\mathsf{Setup}, P, V)$ satisfies *perfect completeness*, *soundness with negligibly small soundness error*, and *per-verifier zero-knowledge* with simulator $\mathcal{S}$:

(i) *Perfect completeness.* For every $x \in L$ with witness w,

$$\mathbf{Pr}[(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Setup}(1^n); \pi \leftarrow P(1^n, \mathsf{pp}, x, w) : V(1^n, \mathsf{pp}, \mathsf{sp}, x, \pi) = 1] = 1.$$

(ii) *Soundness.* For every $x \notin L$ and any purported proof π,

$$\mathbf{Pr}[(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Setup}(1^n) : V(1^n, \mathsf{pp}, \mathsf{sp}, x, \pi) = 1] < \text{negl}(n).$$

(iii) *Per-verifier zero-knowledge.* For every $x \in L$ and any witness w certifying that $x \in L$,

$$\mathcal{D}_1 \leftarrow \{(\widetilde{\mathsf{pp}}, \widetilde{\mathsf{sp}}, x, \tilde{\pi}) : (\widetilde{\mathsf{pp}}, \widetilde{\mathsf{sp}}, x, \tilde{\pi}) \leftarrow \mathcal{S}(1^n, x)\},$$

and

$$\mathcal{D}_2 \leftarrow \{(\mathsf{pp}, \mathsf{sp}, x, \pi) : (\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Setup}(1^n, x), \pi \leftarrow P(1^n, \mathsf{pp}, x, w)\}$$

are computationally indistinguishable.

You may assume for simplicity that $|x| = n$.

Remark. (Note this is not a supposed to be a hint.) To simplify presentation, all computational security notions are quantified over non-uniform p.p.t. adversaries in this problem.

Remark. (Note this is not a supposed to be a hint.) It might be interesting to ponder if your system is *reusable*, in the sense that when Clint and Mallory use the same setup parameters more than once, soundness and (per-verifier) zero-knowledge still hold. Or more generally, if you want to use the system in a bigger system, what might become problematic? (Generally such considerations are called *composability.*)

Remark. (Note this is not a supposed to be a hint.) For simplicity, we have only asked you to prove ordinary soundness in this problem, but in order to properly capture their application, Clint and Mallory should actually design a zero-knowledge proof system that is a *proof of knowledge.* Note that basic soundness only guarantees that false statements cannot be proven—and, if the TAs are to be trusted, then x is always in L anyway, so there is no point to having Mallory prove that $x \in L$ to Clint. On the other hand, the stronger property of *proof of knowledge soundness* (or *knowledge soundness*) requires that there exists a p.p.t. algorithm, called an *extractor*, that with rewinding oracle access to any potentially cheating prover P^*, can output a valid witness w certifying that $x \in L$. The existence of the extractor can then be taken as proof that, if Mallory is accepted by Clint, she must have actually 'known' the witness w, and not merely the fact that $x \in L$.

In practice, many zero-knowledge proof systems are also proofs of knowledge (or can be fairly easily modified to capture the stronger soundness requirement). For simplicity, we do not ask you to prove the stronger soundness property here.

2. *(11 points)* **Two-Server PIR with Pre-Processing**

In this problem, we will consider private information retrieval. In this model, an array $x \in \{0,1\}^n$ is stored by one or more servers, and a user wants to a single bit of x, without the servers knowing which bit he wants. We will consider a two-server model of PIR. Each server has a copy of x and will answer the users queries, but the servers **do not communicate** with each other. Consider the following two-server PIR protocols between user U and servers S_0 and S_1:

- The user chooses a random subset $T \subseteq \{1, 2, \ldots, n\}$, including each value independently with probability 1/2. (Note that T can be represented as an n-bit string w_T by letting the bit in position j represent whether or not $j \in T$.)
- The user calculates the set $T \oplus i$, which is $T \cup \{i\}$ if T does not contain i, and $T \setminus \{i\}$ if T does not contain i.
- The user sends w_T to the server S_0 and $w_{T \oplus i}$ to the server S_1.
- Given a set, each server sends back the XOR of all the bits of x in that set.
- The user calculates the XOR of the two responses from S_0 and S_1.

(a) *(2 points)* Prove that the scheme is *information-theoretically private*; that is, for any two different values i and i', the view of each server S_0 and S_1 is perfectly indistinguishable, even if the servers are computationally unbounded. (It will be the case that the views of S_0 and S_1 together reveal i, but that is OK. Our assumption is that of non-collusion, that is S_0 and S_1 do not ever collude and put their views together.)

(b) *(4 points)* Unfortunately, the above protocol has a total communication complexity of $2(n+1)$ bits, because the user U sends an n$n$bit message and receives a one-bit response from each of the two servers. Note that there is a trivial protocol with communication complexity n that is also private: one of the servers simply sends the entire string x to the user U. We would like to reduce the amount of communication to significantly below n. Alter the scheme from part 2a in such a way as to reduce the amount of communication to $o(n)$.

(Hint: It might be useful to reorganise the database into a $\sqrt{n} \times \sqrt{n}$ table.)

(c) *(5 points)* Unfortunately, in the above protocol each server needs to look at the entire database to answer every query. That's too computationally expensive.

Your goal is to alter the scheme from 2b in such a way as to reduce the computation *per query* to $O(n/\log n)$, while keeping the communication complexity the same. In order to do that, you are allowed to preprocess the database into a polynomially larger string in an *offline* phase, that is before receiving any queries. This preprocessing could be computationally expensive, that is, it could take polynomial time in n. However, the computation in the *online* phase, after receiving a query from the client, should take time $o(n)$. In particular, a solution that achieves online computational complexity $O(n/\log n)$ will receive full points.

3. *(6 points)* **Graphic Secret Sharing.**

In this problem, we consider interesting variants of secret-sharing.

(a) *(3 points)* (Path-out-of-graph secret sharing.) You have been approached by the king of Pessiland, who wishes to establish a series of qualifying trials by which his squires may be elevated to the knighthood. He has built a labyrinth of bridges and islands in the moat around his castle such that each bridge is manned by a guard. The king wants to design the maze such that, in order to succeed, the squire being tried must recover a secret piece of information from each guard that he passes on the way from the starting island (node s) to the king's castle (node t), such that all the secrets that the squire gathers along his journey can be put together after he reaches the king's castle into a password which will cause the guards at the castle to unlock the front gate for him. However, the king wishes the squire to be able to succeed regardless of which path he takes through the maze from the starting node s to the finishing node t.

Fix some prime p, and consider some secret x which is chosen uniformly at random from $\mathbb{Z}_p$. Model the labyrinth as an undirected, connected graph $G = (V, E)$ where $|V| = n$ and $|E| = m$. Suppose there is a value in $\mathbb{Z}_p$ associated with each distinct edge in the graph, which is the secret held by the guard responsible for that edge. Fix two distinct vertices $s, t \in V$ corresponding to the starting node s and the ending vertex t.

Now, the king wants to assign values in $\mathbb{Z}_p$ to each of the m distinct edges so that the following properties hold.

- For any subset $E' \subseteq E$ so that the corresponding edges contain a path from vertex s to vertex t, the values assigned to the edges $e \in E'$ can be put together to recover the secret x.
- For any subset $E' \subseteq E$ so that the corresponding edges contain no path from vertex s and t (i.e. the corresponding subgraph contains s and t in different components, or not at all), the values assigned to all parties $m \in M'$ information-theoretically reveal nothing about x. Formally speaking, the distribution of x conditioned on the values of all the shares assigned to parties $m \in M'$ should remain uniform.

Describe an assignment protocol that the king can run such that the above properties hold, and prove that they hold.

(Hint: It might be useful to associate each vertex with a random value in $\mathbb{Z}_p$.)

(b) *(3e points)* (Democracy of democracies) The king of Pessiland has organised his barons and knights so that each of the $2k+1$ barons have $2k+1$ loyal knights that only answer to their respective barons, where $k \geq 1$. One day, the king decides that he wants to go on an adventure across the seas to find Cryptomania—a magical land which is said to keep many secrets that the king longs to learn.

Completing the journey to Cryptomania requires an adventurer from Pessiland to surmount many dangerous obstacles, such as crossing the tempestuous Channel of Publishing in order to reach the comparative safety of the Tenure Isles, before traversing the warm tropical forest in which the Plague of Crackpottery and False Hope is endemic. Since everybody else who has attempted to find Cryptomania has expired in the attempt, the king decides to leave a will in his safe, which can be unlocked with a code s that he chooses uniformly at random from $\mathbb{Z}_p$.

However, he only wants the code to his safe to be revealed if a sufficient number of his loyal barons and knights decide that he is indeed gone never to return. In particular, the king wants to assign values in $\mathbb{Z}_p$ to each of his barons and knights so that the following properties hold for every set $B' \subseteq B$, where B is the set of all the king's barons and knights,

- If for at least $k+1$ of the barons in B', $k+1$ of their knights are also in B', they can together recover the code s.
- Otherwise, the parties in B' information-theoretically learn nothing about the code s. Formally speaking, the distribution of s conditioned on the values of all the shares assigned to parties $b \in B'$ should remain uniform.

Describe an assignment protocol that the king can use such that the above properties hold, and prove that they hold.

4. *(8 points)* **Subset-revealing encryption.**

Professor Moriarty runs a crime ring in 1891, and a large number of criminals report to him. He would like to communicate with all of his subordinates by posting encrypted bulletins in the agony column of *The Times*. However, he would like to keep his subordinates from knowing about operations in which they have no part, in order to minimise the risk of information leakage if any one of them gets caught and confesses or rats out his cronies. For this purpose, he asks you, a time traveller from 2022, to design a *subset-revealing encryption scheme* consisting of the following algorithms:

- $\mathsf{MasterGen}(1^\lambda)$ generates a secret master key msk.
- $\mathsf{Enc}(msk, m)$ takes in a message $m \in \{0,1\}^n$ and produces a ciphertext c. (For simplicity, we will assume that m is always of length $n(\lambda)$ for some large polynomial $n(\cdot)$.)
- $\mathsf{KeyGen}(msk, T)$ takes in the master key msk and a subset $T \subseteq [n]$, and outputs a secret key sk_T. We require that sk_T is *succinct*, in that its length is always $O(\lambda)$ regardless of the size of T.
- $\mathsf{DecMaster}(msk, c)$ outputs a message m. We require that Enc and $\mathsf{DecMaster}$ satisfy perfect correctness.
- $\mathsf{Dec}(sk_T, c)$ outputs $\{m_i\}_{i \in T}$, where $m = \mathsf{DecMaster}(msk, c)$.

We say that this scheme satisfies *subset-revealing correctness* if, for any T and any $m \in \{0,1\}^n$, $\mathsf{Dec}(sk_T, \mathsf{Enc}(msk, m)) = m_T$, where m_T denotes the restriction of $m = \mathsf{DecMaster}(msk, \mathsf{Enc}(msk, m))$ to the coordinates in T.

The security requirement for this scheme is as follows. Consider the following security game between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$:

1. The challenger generates $msk \leftarrow \mathsf{Gen}(1^\lambda)$.

2. $\mathcal{A}$ generates a subset $T \subseteq [n]$ and sends T to the challenger. The challenger responds by providing $sk_T \leftarrow \mathsf{KeyGen}(msk, T)$ to $\mathcal{A}$.
3. $\mathcal{A}$ submits any message $m_1 \in \{0,1\}^n$ to the challenger, and the challenger responds with $\mathsf{Enc}(msk, m_1)$.
4. Step 3 is repeated an arbitrary polynomial in λ many times.
5. $\mathcal{A}$ outputs two messages m, m' such that $m_T = m'_T$, where where m_T denotes the restriction of a message m to the coordinates in T.
6. The challenger picks $m^* \leftarrow \{m, m'\}$ uniformly at random and sends $\mathcal{A}$ the encryption $\mathsf{Enc}(msk, m^*)$.

We say the scheme is *non-adaptively subset-revealing secure* if no PPT $\mathcal{A}$ can guess which message the challenger chose in step 6 with better than $\frac{1}{2} + \mathsf{negl}(\lambda)$ probability.

To use this encryption scheme, Moriarty, who holds the master key, can encrypt any message using the master key and send the resulting ciphertext to the agony column of *The Times*. At the same time that he generates his master key, Moriarty can also produce secret keys corresponding to different subsets T for each of his subordinates, in accordance with the operations that they are part of.

Let $\mathbb{G}$ be a group of prime order $p(\lambda) = 2^{O(\lambda)}$, and let g be a generator of $\mathbb{G}$. Consider a subset-revealing encryption scheme whose $\mathsf{MasterGen}, \mathsf{Enc}, \mathsf{DecMaster}$ algorithms are defined as follows:

- $\mathsf{MasterGen}(1^\lambda)$ samples $y_1, \ldots, y_n \leftarrow \mathbb{Z}_p$ independently and uniformly at random and outputs these.
- $\mathsf{Enc}(msk, m)$ samples $r_1, \ldots, r_n$ independently and uniformly at random from $\mathbb{Z}_p$ and outputs $g^{r_1}, \ldots, g^{r_n}$ in addition to the $n \times n$ matrix E whose (i,j)th entry is

$$E_{ij} = \begin{cases} g^{r_i y_j + m_i} & i = j \\ g^{r_i y_j} & \text{else} \end{cases}.$$

- $\mathsf{DecMaster}(msk, c)$ computes the matrix M whose (i,j)th entry is $g^{r_i y_j}$ using msk and $g^{r_1}, \ldots, g^{r_n}$, and for all i computes $E_{ii} \cdot M_{ii}^{-1} = \begin{cases} g & m_i = 1 \\ 1 & m_i = 0 \end{cases}$.

Define the algorithms KeyGen and Dec for this encryption scheme such that the scheme satisfies subset-revealing correctness, and prove its non-adaptive subset-revealing security assuming the DDH assumption in $\mathbb{G}$. Your KeyGen algorithm should be such that $\mathsf{KeyGen}(msk, T)$ always outputs a *single element* of $\mathbb{Z}_p$.

Hint: You may cite past homework problems, but we recommend that you do the security reductions in this problem carefully. A good sanity check is whether or not you formally used all the relevant assumptions (e.g.: where do you use that $m_T = m'_T$?

where do you use that $y_1, \ldots, y_n$ are sampled independently?). You should also check carefully that your reduction's answers to all of $\mathcal{A}$'s queries are distributed how $\mathcal{A}$ expects them to be distributed. If you handwave, Professor Moriarty will not be happy, and he is 'not a man who lets the grass grow under his feet'.

Remark. The definition of security which we gave is non-adaptive in the sense that $\mathcal{A}$ is not allowed to submit encryption queries before it provides the subset T to the challenger. Can you come up with an *adaptively* secure scheme, in which $\mathcal{A}$ is allowed to submit encryption queries both before and after it provides the subset T to the challenger?

5. *(8 points)* **Commitments from LWE.**

Please do *either* problem 1 *or* problem 5, and indicate which one you're doing on your problem set.

In this problem, we will work with the Learning With Errors assumption in the two forms stated below.

Definition 1 (decisional LWE)**.** Let $m(\lambda), n(\lambda), q(\lambda)$ be functions of λ. We say that the (m, n, q, χ) decisional LWE problem *with secrets drawn from* $S \subseteq \mathbb{Z}_q^n$ is computationally intractable if, for every (nonuniform) PPT distinguisher D and for $x \leftarrow_R S$,

$$\Big|\Pr[D(A, Ax + e) = 1] - \Pr[D(A, r) = 1]\Big| \leq \mathsf{negl}(\lambda),$$

where $A \leftarrow_R \mathbb{Z}_q^{m \times n}, r \leftarrow_R \mathbb{Z}_q^m$, and where e is a vector in $\mathbb{Z}_q^m$ sampled from the distribution χ. All linear algebraic manipulations are done over $\mathbb{Z}_q$.

Definition 2 (search LWE)**.** Let $m(\lambda), n(\lambda), q(\lambda)$ be functions of λ. We say that the (m, n, q, χ) search LWE problem is computationally intractable if, for every (nonuniform) PPT adversary D and for $x \leftarrow_R \mathbb{Z}_q^n$,

$$\Big|\Pr[D(A, Ax + e) = x]\Big| \leq \mathsf{negl}(\lambda),$$

where $A \leftarrow_R \mathbb{Z}_q^{m \times n}, r \leftarrow_R \mathbb{Z}_q^m$, and where e is a vector in $\mathbb{Z}_q^m$ sampled from the distribution χ. (For this version of the assumption, we assume that we are in a parameter regime where x is uniquely recoverable from $Ax+e$.) All linear algebraic manipulations are done over $\mathbb{Z}_q$.

As a reminder, a *bit commitment scheme* is a protocol between two PPT parties, S (the sender) and R (the receiver). S receives as input a bit $b \in \{0, 1\}$ and a security parameter 1^λ; R receives just the security parameter. A commitment scheme has two phases, a *commit phase* and a *reveal phase*, as follows:

- During the commit phase, S and R interact for some number of rounds. At the end of the commit phase, both parties agree on a commitment c, and S also retains some private decommitment information d.

- During the reveal phase, S sends (b, d) to R, and R computes a function $\mathsf{Verify}(b, c, d)$.

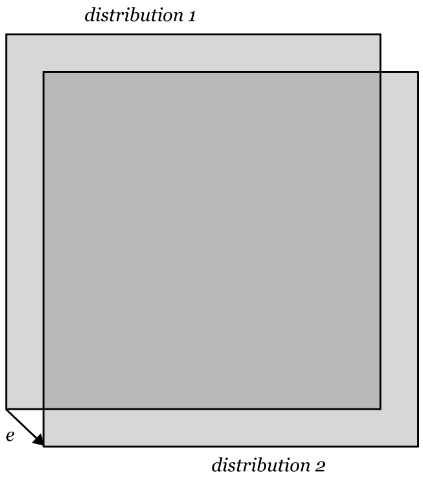
For a reminder on the definitions of *completeness*, *binding*, and *hiding*, see Appendix A. As we learned in class, a commitment scheme that is both statistically hiding and statistically binding is impossible to achieve. In this problem, we will attempt to construct both a *perfectly binding, computationally hiding* and a *computationally binding, statistically hiding* commitment scheme from LWE. (For simplicity, when we construct the statistically hiding scheme, we will only consider honest-but-curious receivers.)

(a) *(2 points)* Assume that the **decisional** LWE assumption holds with secrets drawn from $\{0,1\}^n$ for $n(\lambda) = \lambda, q(\lambda) = 2^{O(\sqrt{\lambda})}$, $m(\lambda)$ such that $m(\lambda) = \omega(n \log q)$ and $m(\lambda) = \text{poly}(\lambda)$, and χ any distribution in which every coordinate in e is sampled uniformly and independently at random from the interval $[-B, B]$, for $B(\lambda) = \Omega(\sqrt{\lambda})$ such that $B < \sqrt{q}/2$ for all λ.[1] Construct a **perfectly binding, computationally hiding** bit commitment scheme and prove that it is complete, perfectly binding, and computationally hiding. (You may cite results from class, but make sure you write down the commitment scheme formally, choose parameters appropriately, and show that your parameter choices are valid.)

(b) *(6 points)* Assume that the **search** LWE assumption holds for any $n(\lambda) = \lambda, q(\lambda) = 2^{O(\sqrt{\lambda})}$, $m(\lambda)$ such that $m(\lambda) = \omega(n \log q)$ and $m(\lambda) = \text{poly}(\lambda)$, and χ any distribution in which every coordinate in e is sampled uniformly and independently at random from the interval $[-B, B]$, for $B(\lambda) = \Omega(\sqrt{\lambda})$ such that $B < \sqrt{q}/2$ for all λ. Construct a **computationally binding** bit commitment scheme that is **statistically hiding against honest-but-curious receivers**, and prove that it is complete, computationally binding, and statistically hiding against honest-but-curious receivers.

This subproblem is trickier than the last one—but don't despair! It may help you to gain intuition to try to construct such a bit commitment scheme from DLOG first (it's easier there). Then, it may help you to stare at the following picture for a while:

[1] The weird setting of the upper bound on B is just to make sure that, within the range of m that we allow, we are always in the regime (for sufficiently large λ) where x can be uniquely recovered from $Ax + e$.



(c) *(Bonus: 1 point)* Briefly indicate how you would lift your scheme from part (b) (using LWE) into a scheme which is still computationally binding and also statistically hiding against malicious unbounded receivers.

A Commitment schemes: properties

We generally require commitment schemes to be *complete*, *binding*, and *hiding*.

- *Complete.* For any $b \in \{0, 1\}$ and any (c, d) resulting from the commit phase, $\mathsf{Verify}(b, c, d) = 1$.
- *Binding.* There are three flavours of this: perfect, statistical, and computational.
  - We say the scheme is *perfectly binding* if, for any c resulting from the commit phase, there does not exist $(d, d'), d \neq d'$ such that $\mathsf{Verify}(c, d) = \mathsf{Verify}(c, d') = 1$.
  - We say the scheme is *statistically binding* if the probability over c resulting from the commit phase between potentially malicious unbounded S^* and honest R that there exists $(d, d'), d \neq d'$ such that $\mathsf{Verify}(c, d) = \mathsf{Verify}(c, d') = 1$ is less than $\mathsf{negl}(\lambda)$.
  - We say the scheme is *computationally binding* if the probability over c resulting from the commit phase between potentially malicious PPT S^* and honest R that PPT S^* can find $(d, d'), d \neq d'$ such that $\mathsf{Verify}(c, d) = \mathsf{Verify}(c, d') = 1$ is less than $\mathsf{negl}(\lambda)$.
- *Hiding.* There are again three flavours of this: perfect, statistical, and computational.
  - We say the scheme is *perfectly hiding* if a potentially malicious unbounded R^*'s view in the commit phase when (honest) S receives input 0 is identical to R^*'s view in the commit phase when S receives input 1.

- We say the scheme is *statistically hiding* if a potentially malicious unbounded R^*'s view in the commit phase when (honest) S receives input 0 is statistically indistinguishable from (i.e. close in statistical distance to) R^*'s view in the commit phase when S receives input 1.
- We say the scheme is *computationally hiding* if a potentially malicious PPT R^*'s view in the commit phase when (honest) S receives input 0 is computationally indistinguishable from R^*'s view in the commit phase when S receives input 1.