CS4236 Assignment 3 feedback

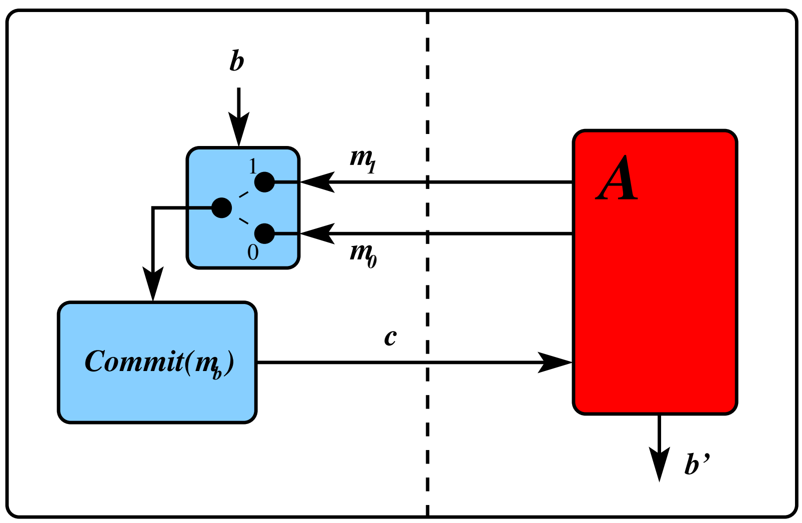
November 9, 2022

Assignment questions:

1. On page 188 is a description of the $\text{Hiding}_{\mathcal{A},\Pi}(n)$ experiment/game for a commitment scheme. Use the definition (from class) of the scheme $\Pi(n) = (\text{Setup}(1^n), \text{Commit}(a), \text{Open}(c_a))$, where $c_a \leftarrow \text{Commit}(a)$ creates the commitment, and $a \leftarrow \text{Open}(c)$ opens it. Draw the experiment/game using the same shapes and ideas as used in the game descriptions in class. Provide a formal definition of the hiding property. (4 marks)

Feedback: *In this question I expected you to follow the consistent diagramming technique that has been used in class to show the game.*

Possible Answer: *A possible diagram representing the game is:*



In this game, the adversary wins (i.e. result is 1) if $b = b'$. The definition should be something like this:
Definition: *A commitment system Π is "hidden" iff for any (PPT) adversary $\mathcal{A}$ there is a negl, s.t.*

$$\Pr[\text{Hiding}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \text{negl}(n)$$

Marking schedule: *The answer should*

(a) *have an accurate diagram.* *(2 marks)*

(b) *give a clear definition for Hiding.* *(2 marks)*

2. Assume that $h_1 : \{0,1\}^{2\times n} \rightarrow \{0,1\}^n$ is a collision resistant compression function. This is used to define a new compression function with an extra bit b concatenated to x:

$$h_2(x + b) \stackrel{\text{def}}{=} \begin{cases} b = 1 & \rightarrow \quad b + h_1(x) \\ b = 0 & \rightarrow \quad b^{n+1} \end{cases}$$

Is $h_2 : \{0,1\}^{2\times n+1} \rightarrow \{0,1\}^{n+1}$ also collision resistant? Show your reasoning. (2 marks)

Feedback: *In this question I expected you to understand the framework and show if the construction was or was not also a collision resistant function.*

Possible Answer: *The resultant function is **not** a collision resistant function. We can make up a simple attack: Pick any two values x, y, where $x \neq y$, and then*

$$h_2(x + 0) = 0^n = h_2(y + 0)$$

We can construct collisions easily, so this version is NOT collision resistant. ■

Marking schedule: *The answer should*

- ***(a)*** *be correct.* *(1 mark)*
- ***(b)*** *give a clear justification.* *(1 mark)*

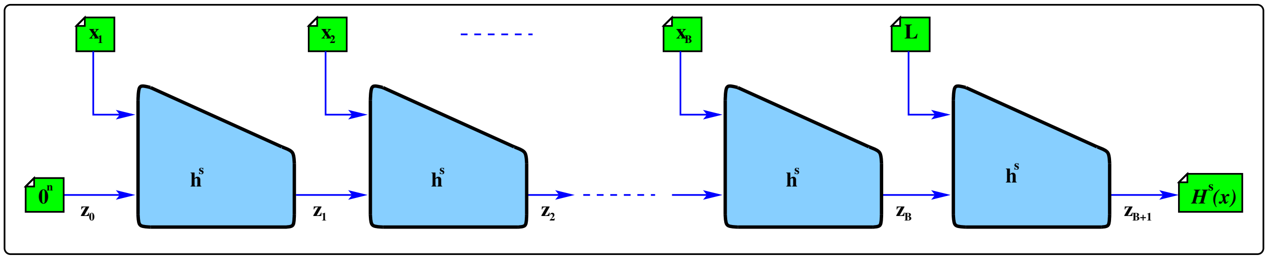
3. Assume we have a collision resistant hash function $\mathcal{H}(x) \stackrel{\text{def}}{=} \mathcal{H}_1(\mathcal{H}_1(x))$. Prove that $\mathcal{H}_1$ is collision resistant. (4 marks)

Feedback: *Many of you used "suppose not".*

Possible Answer: *If we suppose not, then we have a pair $\langle x, y\rangle$, where $x \neq y$ and $\mathcal{H}_1(x) = \mathcal{H}_1(y)$. This would means that $\mathcal{H}_1(\mathcal{H}_1(x)) = \mathcal{H}_1(\mathcal{H}_1(y))$, where $x \neq y$, leading to $\mathcal{H}(x)$ not being collision resistant. However we know that $\mathcal{H}(x)$ is collision resistant. As a result, $\mathcal{H}_1$* is collision resistant. ■

Marking schedule: *The proof should*

- ***(a)*** *have a clear approach and understanding.* *(1 mark)*
- ***(b)*** *be clear, and justify the steps.* *(3 marks)*



4. The above diagram shows the Merkle Damgård construction to construct collision resistant hashes over longer messages out of compression functions. We write the final hash as $\mathcal{H}^s(x) = Z_{B+1} = h^s(Z_B + L)$. Consider the alternative final hash $\mathcal{H}^s_1(x) = Z_B + L$. Is this still collision resistant? (4 marks)

Feedback: *The diagram is showing construction 5.3 in the textbook, which is directly followed by Theorem 5.4, which has a proof (on pages 157,158). We can prove this by appealing to this proof, and identifying the differences if the last step is changed.*

Possible Answer: *The variation in the last step leads to a new construction which is still collision resistant. The only difference in the proof of Theorem 5.4 is in Case 1. In the above construction it is clear that when $L \neq L'$ the result cannot be a collision (note that $Z_B + L$ cannot equal $Z'_{B'} + L'$ when $L \neq L'$). As a result, this version is collision resistant.* ■

Marking schedule: *The proof should*

(a) *have a clear understanding, and be correct (i.e. still collision resistant).* *(1 mark)*

(b) *be clear, and justify the steps.* *(3 marks)*

5. (Similar to the situation described in the first paragraphs of 5.6.2, but without a Merkle tree). In a scheme/system, clients upload files to a server. Later, when a client retrieves a file, it wants a "fingerprint" δ-guarantee that it is the original, unmodified file. The signature is $\Pi(n) = (\mathrm{Put}(x_i), \mathrm{Get}(i), \mathrm{Vrfy}(i, x_i, \delta))$, where $\langle x_i, \delta \rangle \leftarrow \mathrm{Get}(i)$ returns the file and a fingerprint, and $\mathrm{ok} \leftarrow \mathrm{Vrfy}(i, x_i, \delta)$ returns 1/0 if the fingerprint matches/does-not-match the file.

(a) Describe an experiment/game which could be used to define the security of this system - i.e. that an adversary cannot verify $\mathrm{Vrfy}(i, x, \delta)$ unless $x = x_i$. (2 marks)

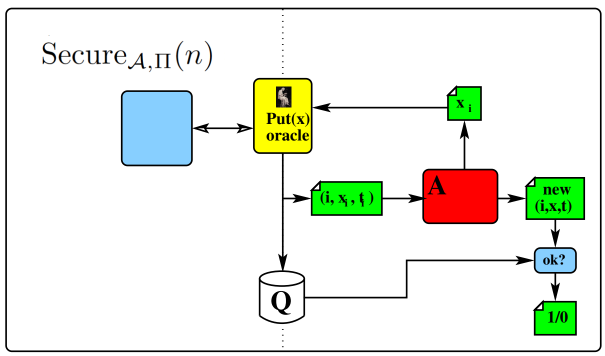
(b) Formally define the property exposed in the above game. (Π is *secure* if ... for all ...) (2 marks)

(c) Construct a "fingerprint" server, and explain why you think it has the "*secure*" property. (2 marks)

Feedback: *I did not expect a detailed answer for each part of this (given it was only worth 2 marks for each part). In class I clarified that the* $\langle i, \delta \rangle \leftarrow \mathrm{Put}(x)$ *algorithm uploads a file, returning an index and a fingerprint* δ.

Possible Answer: *Perhaps:*

(a) *A possible game is called* $\mathrm{Secure}_{\mathcal{A},\Pi}$. *An adversary has a* $\mathrm{Put}(x)$ *oracle with storage as needed. The adversary wins* $\mathrm{Secure}_{\mathcal{A},\Pi}(n) = 1$ *iff it can produce a triple* $\langle i, x, \delta \rangle$ *such that* $\mathrm{Vrfy}(i, x, \delta)$ *and* $x \neq x_i$.



(b) *The definition is:*
Definition: *A fingerprinted storage system* Π *is "secure" iff for any (PPT) adversary* $\mathcal{A}$ *there is a* negl, *s.t.*

$$\Pr[\mathrm{Secure}_{\mathcal{A},\Pi}(n) = 1] \leq \mathrm{negl}(n)$$

(c) *We can construct a system based on a collision resistant hash function. Assume that each time we upload a file* x_i, *we associate it with a hash* $\delta \leftarrow \mathcal{H}^s(x_i)$, *stored on the server along with the index. Later, if someone attempts to verify a file, the server can just check.*

Marking schedule: *The sections should include*

(a) *a clear description of a game.* *(2 marks)*

(b) *a formal definition of the property.* *(2 marks)*

(c) *a clear description of a construction.* *(2 marks)*