

Problem Set 4

Your name

Your Collaborators

Total Number of Points: 40.

Instructions: Solutions should be typeset in $\text{\LaTeX}$ and submitted as PDF by 11:59:59 PM ET on the due date. You can collaborate on the psets in small groups (of up to three), but must write your solutions separately, in your own words. This means not copying from a board nor from your collaborator's computer.

We strongly recommend **not** using AI tools to solve your problem sets. You are responsible for understanding everything you submit and you must be able to explain and justify every part of your solution. In particular, the oral problem set review (10% of the course grade) will evaluate your understanding of the solutions you submit. In all cases, you must list your collaborators, human or machine.

Problem 1. Hash It Out. (14 points)

In this problem, we will explore four different definitions of hash functions. We say that a family of functions $\mathcal{H} = \{H_k : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n\}_{k \in \{0, 1\}^m}$ is a X hash function, for $X \in \{\text{"universal"}, \text{"weak collision resistant"}, \text{"universal one-way"}^1, \text{"collision resistant"}\}$, if it has the following properties:

- **Efficiency:** Given $k \in \{0, 1\}^m$ (where $m = \text{poly}(n)$) and $x \in \{0, 1\}^{2n}$, $H_k(x) \in \{0, 1\}^n$ can be computed in $\text{poly}(n)$ time.

- **Security:**

- If $X = \text{"universal"}:$ There exists a negligible function negl such that for all $x_1 \neq x_2 \in \{0, 1\}^{2n}$,

$$\Pr[k \leftarrow \{0, 1\}^m : H_k(x_1) = H_k(x_2) \wedge x_1 \neq x_2] \leq \text{negl}(n).$$

- If $X = \text{"weak collision resistant"}:$ For all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that the following holds:

$$\Pr[(k, x_1) \leftarrow \{0, 1\}^m \times \{0, 1\}^{2n}; \mathcal{A}(k, x_1) \rightarrow x_2 : H_k(x_1) = H_k(x_2) \wedge x_1 \neq x_2] \leq \text{negl}(n).$$

- If $X = \text{"universal one-way"}:$ For all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that for all $x_1 \in \{0, 1\}^{2n}$, the following holds:

$$\Pr[k \leftarrow \{0, 1\}^m; \mathcal{A}(k, x_1) \rightarrow x_2 : H_k(x_1) = H_k(x_2) \wedge x_1 \neq x_2] \leq \text{negl}(n).$$

- If $X = \text{"collision resistant"}:$ For all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that the following holds:

$$\Pr[k \leftarrow \{0, 1\}^m; \mathcal{A}(k) \rightarrow (x_1, x_2) : H_k(x_1) = H_k(x_2) \wedge x_1 \neq x_2] \leq \text{negl}(n).$$

- (2 points) Construct a family of universal hash functions without using any cryptographic assumptions. *Hint: Consider multiplication by a random matrix $A \leftarrow \{0, 1\}^{n \times 2n}$.*
- (3 points) Show how to build a universal one-way hash function family from any weak collision resistant hash function family.
- (4 points) Show that there exists a hash family that is universal one-way but not collision resistant. (You may assume the existence of universal one-way hash functions.)

¹also called "targeted collision resistant"

(d) (2 points) Show that for all of these definitions, for all $k \in \{0, 1\}^m$,

$$\Pr_{x \leftarrow \{0,1\}^{2n}} [|H_k^{-1}(H_k(x))| \leq 1] \leq \frac{1}{2^n}.$$

(e) (3 points) Suppose $\mathcal{H}$ is a weak collision resistant hash function family. Show that the function $F : \{0, 1\}^{m+2n} \rightarrow \{0, 1\}^{m+n}$ given by $F(k, x) = (k, H_k(x))$ is a one-way function.

It turns out that the existence of one-way functions also implies the existence of weak collision resistant hash functions and universal one-way hash functions, even though it is an open question for collision resistant hash functions! You may assume these facts throughout the problem set.

Problem 2. Shrinking Signature Trees. (8 points)

Recall the many-time signature scheme using signature trees from class (see Lecture 12 slides). In the scheme presented in class, Alice computes many (VK, SK) pairs and arranges them in a tree of depth λ . In this problem, we will explore whether Alice can save space by computing only parts of the tree.

- (a) (6 points) Suppose Alice signs messages using the leaves in order from left to right. Instead of generating the entire tree in advance, she proposes generating only the nodes along the path to the current leaf and their siblings. Show that Alice can use this approach to sign up to 2^λ messages while storing only $O(\lambda)$ signing keys, verification keys and signatures at any time. Briefly justify that the resulting signature scheme is EUF-CMA secure, assuming that the underlying one-time signature scheme is EUF-CMA secure.
- (b) (2 points) Now Alice wants to reduce her storage even further. She proposes generating the nodes along the path to the current leaf and their siblings as in part (a), but discarding the signing keys of the sibling nodes after their parent has signed their verification keys. When she later needs to use one of these sibling nodes, she generates a fresh signing and verification key pair for it. Explain why this approach does not work.

Problem 3. Statistically Hiding Commitments (10 points)

Recall the commitment scheme based on an injective one-way function from class (Lecture 13 notes, Section 5).

- (a) (3 points) Prove that no commitment scheme without public parameters can be both statistically hiding and computationally binding.
- (b) (7 points) Suppose that instead of an injective one-way function, we use a family of collision-resistant hash functions

$$H_{pp} : \{0, 1\}^{2\lambda} \rightarrow \{0, 1\}^\lambda,$$

where pp is a publicly sampled hash-function key. Consider the following modification of the commitment scheme from class: to commit to $b \in \{0, 1\}$, Alice samples independent uniform $r, s \in \{0, 1\}^{2\lambda}$ and computes

$$\text{Com}(pp, b; (r, s)) = (H_{pp}(r), s, \langle r, s \rangle \oplus b).$$

To open the commitment, Alice reveals r , and the receiver verifies the opening as in the original construction.

Prove that the resulting commitment scheme is statistically hiding and computationally binding.

Problem 4. Zero Knowledge Proof for Knight's Move Sudoku (8 points)

Knight's Move Sudoku is a variant on the popular logic puzzle Sudoku. For a Knight's Move Sudoku puzzle for positive integer n , you are given as input an $n^2 \times n^2$ grid of cells, where the cells are partitioned into n^2 disjoint $n \times n$ subgrids. Some of the cells begin filled with integers from 1 to n^2 .

A solution is a way to assign an integer from 1 to n^2 to each cell which hasn't already been filled in a way that:

1. Each row contains each of the integers from 1 to n^2 exactly once.
2. Each column contains each of the integers from 1 to n^2 exactly once.
3. Each of the n^2 different $n \times n$ subgrids contains each of the integers from 1 to n^2 exactly once.
4. For all cells that are a "knight's move" away from each other, the entries are distinct. Two cells $(i_1, j_1) \in [n^2] \times [n^2], (i_2, j_2) \in [n^2] \times [n^2]$ are a "knight's move" away if either (a) $|i_1 - i_2| = 1$ and $|j_1 - j_2| = 2$ or (b) $|i_1 - i_2| = 2$ and $|j_1 - j_2| = 1$.

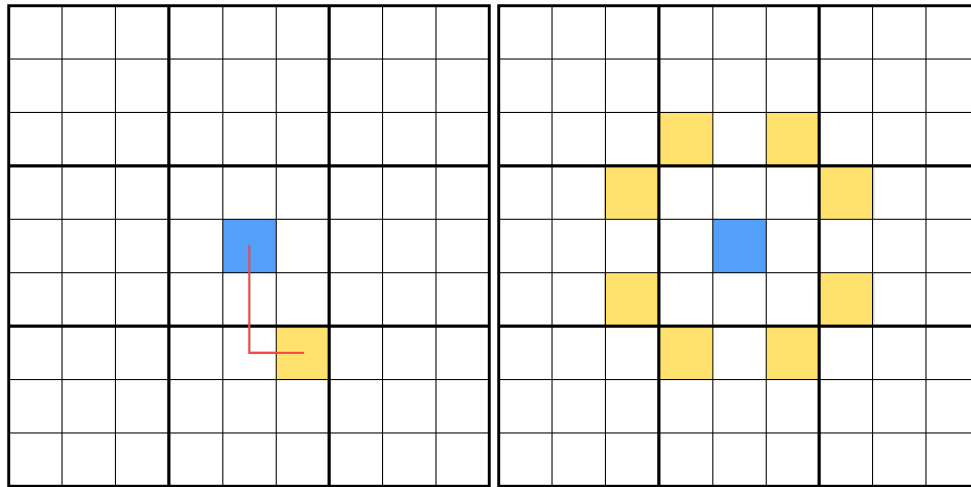


Figure 1: An example demonstrating what a "knight's move" is, for $n = 3$, from <https://masteringsudoku.com/chess-sudoku/>.

Design a zero-knowledge protocol for Knight's Move Sudoku. Your protocol should be computationally zero knowledge, have perfect completeness, and have soundness error $1 - p(n)$ for a non-negligible function p . You must prove that your construction satisfies all of the above properties. You may assume the existence of one-way functions.