CSE 426: Cryptography
Prof. Stefano Tessaro

University of Washington
Autumn 2023

Homework 1

Posted: Wednesday, October 4, 2023 – 11:59pm
Due: Wednesday, October 11, 2023 – 11:59pm
(Gradescope submission, instructions will be posted.)

Instructions and Rules

- Write your solutions clearly, and ideally, type them up. If they are handwritten, you are responsible to ensure they are readable. Justify *all claims* of your solution. Partially incorrect solutions can still be worth several points, but unjustified incorrect solutions will result in zero points for the corresponding question.
- You are not allowed to copy or transcribe answers to homework assignments from others or other sources. You are not allowed to look up solutions online.
- You need to provide an individual solution. You are allowed to have high-level discussions with other students (for instance, review the definition of a concept, discuss what a homework question mean, and high-level approaches). Please disclose if possible who you discussed with.
- You are given *six* late days with no question asked, but can only use at most *three* per homework. To use late days, send an e-mail to `cse426-staff@cs.washington.edu` *before* the deadline. Other late submissions will be considered on a per-case basis, but expect to provide an explanation.

Task 1 – Encryption Scheme (10 points)

Let $\mathbb{Z}_{10} = \{0, 1, \ldots, 9\}$. We consider a symmetric encryption scheme $\Pi = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$, for which both the message and ciphertext spaces are $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{10}^4$, i.e., both a plaintext M and a ciphertext C consist of four decimal digits, and where:

- Kg outputs a secret key $K = (d, \pi)$, where $d \xleftarrow{\$} \mathbb{Z}_{10}$ and $\pi \xleftarrow{\$} \mathsf{Perms}(\mathbb{Z}_{10})$, i.e., d is a uniformly chosen random decimal digit and π is a uniformly chosen random permutation of the decimal digits.
- The encryption algorithm is defined by the following procedure:

$$
\begin{array}{l}
\underline{\textbf{procedure } \mathsf{Enc}(K = (d, \pi), M = (M[1], \ldots, M[4])) :} \\
x_0 \leftarrow d \\
\textbf{for } i = 1 \text{ to } 4 \textbf{ do} \\
\quad x_i \leftarrow (x_{i-1} + M[i] + 1 - i) \bmod 10 \\
\quad C[i] \leftarrow \pi(x_i) \\
\textbf{return } C = (C[1], \ldots, C[4])
\end{array}
$$

a) [4 points] Complete the description of Π by giving a decryption algorithm Dec that satisfies the correctness requirement discussed in class.

b) [6 points] Show that this encryption scheme is not perfectly secret.

Task 2 – The Shuffle (19 points)

Consider the following symmetric encryption scheme $\Pi' = (\mathsf{Kg}', \mathsf{Enc}', \mathsf{Dec}')$ with plaintext space $\mathcal{M} = \{0,1\}^n$. Moreover:

- Kg' outputs a secret key π, where $\pi \xleftarrow{\$} \mathsf{Perms}(\{1,\ldots,2n\})$, i.e., π is a uniformly chosen random permutation of the set $\{1,\ldots,2n\}$.
- The encryption algorithm is defined by the following procedure:

$$
\begin{array}{l}
\underline{\textbf{procedure } \mathsf{Enc}'(\pi, M = (M[1],\ldots,M[n])):} \\
M' \leftarrow M \| \overline{M} \\
\textbf{for } i = 1 \text{ to } 2n \textbf{ do} \\
\quad C[i] \leftarrow M'[\pi(i)] \\
\textbf{return } C = (C[1],\ldots,C[2n])
\end{array}
$$

  Here, $\overline{M}$ is the bit-wise complement of M, i.e., $\overline{M}[i] = 1 - M[i]$ for all $i = 1,\ldots,n$. Further, $M\|\overline{M}$ is the concatenation of M and $\overline{M}$.

a) [4 points] Describe the ciphertext space $\mathcal{C}$, i.e., the set of all possible valid ciphertexts resulting from encrypting a plaintext $M \in \mathcal{M}$ with some key.

Hint: Find an invariant satisfied by M' for all $M \in \{0,1\}^n$.

b) [4 points] Complete the description of Π' by giving a decryption algorithm Dec' that satisfies the correctness requirement discussed in class.

c) [8 points] Characterize the distribution of $\mathsf{Enc}'(\pi, M)$, for a uniformly chosen π and an arbitrary $M \in \{0,1\}^n$.

d) [3 points] Conclude that Π' satifies perfect secrecy.

Task 3 – Playing with AES (10 points)

We want to develop a better sense of the pseudorandomness of the ciphertexts generated by the AES block cipher. In particular, we will focus on the most commonly used variant with 128-bit keys. Let X be the 16-byte string

$$X = 10\ 04\ 20\ 18\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00$$

in hexadecimal format.

a) [2 points] What is the value of $\mathsf{AES}(X, X)$? Write the result in hexadecimal format. Here, $\mathsf{AES}(K, M)$ is the ciphertext generated by AES on key K and block M.

b) [4 points] Find a 16-byte block M such that the lower half of $C = \mathsf{AES}(X, M)$ is all zero. In other words, C ends with 00 00 00 00 00 00 00 00. Explain how you have found it!

c) [4 points] Find a 16-byte key K with the property that the last byte of $C = \mathsf{AES}(K, X)$ is equal to 00. Explain how you have found it!

You can use the Python code for AES (`hw1.py`) provided on Ed, or any of your favorite programming languages and libraries, to help performing AES evaluations. (Do *not* re-implement AES!)

Task 4 – Distinguishing Advantage (6 points)

The goal of this task is to practice with the notion of distinguishing advantage.

To this end, we are given the following two oracles, O_0 and O_1. They both are initialized by running the (private) procedure $\mathsf{Init}()$, and the adversary can then only call the procedure $\mathsf{Eval}()$.

oracle O_0:	**oracle** O_1:
private **procedure** $\mathsf{Init}()$:	*private* **procedure** $\mathsf{Init}()$:
$b_1 \xleftarrow{\$} \{0,1\}$, $b_2 \xleftarrow{\$} \{0,1\}$	$b_1 \xleftarrow{\$} \{0,1\}$ **if** $b_1 = 0$ **then** $b_2 \xleftarrow{\$} \{0,1\}$ **else** $b_2 \leftarrow 0$
public **procedure** $\mathsf{Eval}()$:	*public* **procedure** $\mathsf{Eval}()$:
return $b_1 \| b_2$	**return** $b_1 \| b_2$

Here, $b_1 \| b_2$ is the concatenation of b_1 and b_2. Consider the following distinguishers D_1 and D_2, which are given access to an oracle O that is either O_0 or O_1:

distinguisher D_1^{O}:	**distinguisher** D_2^{O}:
$b_1 \| b_2 \leftarrow \mathsf{O}.\mathsf{Eval}()$ **return** b_1	$b_1 \| b_2 \leftarrow \mathsf{O}.\mathsf{Eval}()$ **return** $b_1 \oplus b_2$

a) **[3 points]** What is the advantage of D_1 in distinguishing O_0 and O_1?

b) **[3 points]** What is the advantage of D_2 in distinguishing O_0 and O_1?