EE 418 - Assignment 4

Total Points: 100
Autumn Quarter, 2021
Prof. Radha Poovendran
Department of Electrical and Computer Engineering
University of Washington, Seattle, WA 98195

Due: 11:59 pm (PST) on December 2nd (Thursday), 2021 via Canvas

Notes:

- This homework contains both computation questions (marked as **[Com]**) which are required to do by hand calculations and programming questions (marked as **[Pro]**) which are required to write Python /MATLAB codes. Zero points will be awarded if **[Com]** questions are solved via Python/MATLAB scripts and if **[Pro]** questions are solved by hand calculations.
- **Your answers to this homework must be submitted through canvas as a single zip file containing the following:** *i*) **hand written and scanned or word or pdf answers to all the computational and discussion questions as single pdf file.** *ii*) **Python/MATLB codes for programming questions as in filename.py or filename.m respectively.**
- **Name of your submission zip file should follow the following format. "#_$_*EE418_HW4.zip*", where "#" and "$" should be replaced with your first name and last name, respectively.**
- Show the computation steps and/or justify your answers in all the **[Com]** questions. Failure to show computation steps in **[Com]** questions will result zero points.
- You can use and modify the Python functions provided in the file section of the EE 418 canvas page when answering the **[Pro]** questions below.
- You can discuss with others but you need to write your own computation steps for **[Com]** questions and Python/MATLAB codes for **[Pro]** questions.

1. **[Com]**(Hash Functions, 5pts $\times$ 2 = 10pts) Let $n = pq$ be the product of two distinct large primes. Given an input x, define the hash of the input as $h(x) = x^2 \pmod{n}$.

   (a) Why is h preimage resistant? (Of course, there are some values, such as 1, 4, 9, 16, ... for which it is easy to find a preimage. But usually it is difficult.)

   (b) Why is h not collision resistant?

2. **[Com]**(Hash Functions, 5 pts $\times$ 3 = 15 pts)

   (a) Given a hash function where the hash value is a 256-bits long binary string, how many attempts (queries) are required to find two messages m and m' that are different but have the same hash value (i.e., collision), with the average success probability of 0.75?

   (b) Let $n = pq$ where p and q are large distinct prime numbers. a is an integer that is relatively prime to $\phi(n)$ and b is chosen such that $ab = 1 \bmod \phi(n)$. Consider the following hash function. Given $m_1, m_2 \in \mathbb{Z}_n$, the hash function takes message m as input, which is a concatenation of m_1 and m_2, mathematically written as $m = m_1||m_2$. The hash is computed as

   $$h(m) = m_1^a m_2^b \bmod n$$

   Is this hash function second preimage resistant? Explain why or why not.

   (c) Let $n = pq$ where p and q are large distinct prime numbers. a is an integer that is relatively prime to $\phi(n)$ and b is chosen such that $ab \equiv 1 \bmod \phi(n)$. Consider the following iterated hash function, which takes a message $m = m_1||m_2$ as input, where $m_i \in \mathbb{Z}_n$ and $||$ means concatenation. The hash is computed as

   $$h_1 = h_0^a \cdot m_1^b \bmod n$$
   $$h_2 = h_1^a \cdot m_2^b \bmod n$$

   where h_0 is the initial value (known and fixed), and $h(m) = h_2$ is the final hash value. It this hash function second preimage resistant? Explain why or why not.

3. **[Com]** (Birthday Problem, 10 pts) In a family of four, what is the probability that no two people have birthdays in the same month? (Assume that all months have equal probabilities.)

4. **[Com]** (Hash, Preimage Resistance, 5pts $\times$ 3 = 15pts) Suppose that $h : \mathcal{X} \mapsto \mathcal{Y}$ is an $(|\mathcal{X}| = N, |\mathcal{Y}| = M)$-hash function, let

   $$h^{-1}(y) = \{x : h(x) = y\}$$

   and let $s_y = |h^{-1}(y)|$ for any $y \in \mathcal{Y}$. Suppose that we try to solve **Preimage** for the function h, using Algorithm 4.1 in Handout 7, assuming that we only have oracle access for h. For a given $y \in \mathcal{Y}$, suppose that $\mathcal{X}_0$ is chosen to be a random subset of $\mathcal{X}$ having cardinality q.

   (a) Prove that the success probability of Algorithm 4.1 in Handout 7, given y, is

   $$1 - \frac{\binom{N-s_y}{q}}{\binom{N}{q}}.$$

   (b) Prove that the average success probability of Algorithm 4.1 Handout 7 (over all $y \in \mathcal{Y}$) is

   $$1 - \frac{1}{M} \sum_{y \in \mathcal{Y}} \frac{\binom{N-s_y}{q}}{\binom{N}{q}}.$$

   (c) In the case $q = 1$, show that the success probability in part (b) is $1/M$.

5. **[Com]** (Hash, Collision Resistance, 5pts × 2 = 10pts) Suppose $h_1 : \{0,1\}^{2m} \mapsto \{0,1\}^m$ is a collision resistant hash function. Define $h_2 : \{0,1\}^{4m} \mapsto \{0,1\}^m$ as follows:

   (a) Write $x \in \{0,1\}^{4m}$ as $x = x_1||x_2$, where $x_1, x_2 \in \{0,1\}^{2m}$.

   (b) Define $h_2(x) = h_1(h_1(x_1)||h_1(x_2))$.

   Prove that h_2 is collision resistant.
   Hint: Prove by contradition that h_2 is collision resistant: assume that there exists $x_1, x_2 \in \{0,1\}^{4m}$, such that $x_1 \neq x_2$, but $h_2(x_1) = h_2(x_2)$. Also use the fact that h_1 is collision resistant, i.e., there does not exist $x_1, x_2 \in \{0,1\}^{2m}$ such that $x_1 \neq x_2$ but $h_1(x_1) = h_1(x_2)$; in other words, $h_1(x_1) = h_1(x_2)$ implies $x_1 = x_2$.

6. **[Com]** (CBC-MAC, 10pts × 2 = 20pts) Consider the CBC-MAC as illustration in Fig. 1, where the initial vector (IV) is a block of zeros. The corresponding equations for CBC encryption are given as

$$C_i = E_K\{C_{i-1} \oplus m_i\}, \text{ for } i = 1, 2, ..., n. \tag{1}$$

   Recall that in the original CBC-MAC, C_n represents the generated MAC.

   Now, "A" wants to modify the original CBC-MAC by using the last block of message m_n as the IV, following the original CBC-MAC protocol up to step $(n-1)$, and then releasing C_{n-1} as the MAC. The new equation of the iteration of the modified CBC-MAC is given as follows:

$$C_i = E_K\{C_{i-1} \oplus m_i\}, \text{ for } i = 1, 2, ..., n-1. \tag{2}$$

   (a) Draw a block diagram of the modified CBC-MAC.

   (b) If the same encryption algorithm and the same encryption key are used for both the original and modified CBC-MAC, show that the modified CBC-MAC **does not** give the same output as te original CBC-MAC for a message $m = m_1||m_2||...||m_n$.

   (Hint: Proof by induction.)

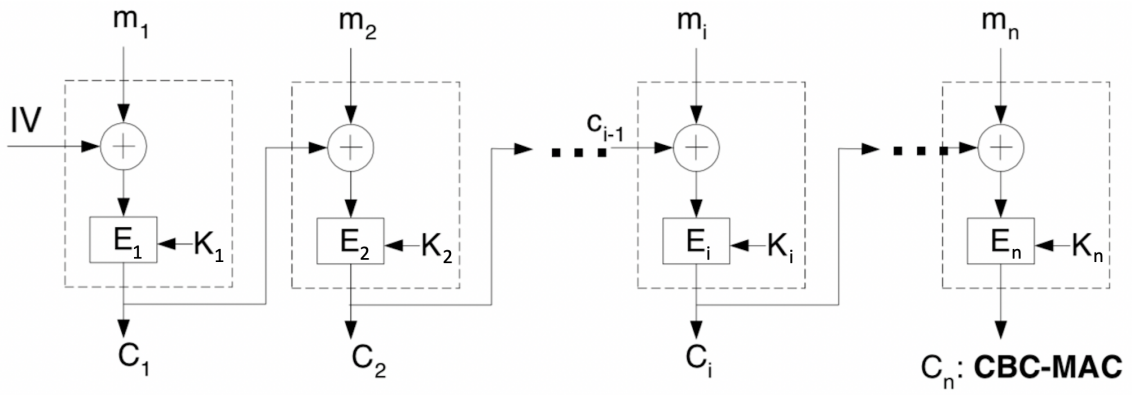

Figure 1: CBC-MAC

7. **[Pro]** (CBC-MAC, 20 pts) Consider a CBC-MAC given in Figure 1 with block size $t = 16$ and $IV = 1010000011111010$. Assume message m is given as a binary string consists of n message blocks of size t (i.e., $m = m_1||m_2||\dots m_n$) and the encryption in CBC-MAC given in Figure 3 is done as follows:

   - If i corresponding to ith encryption box E_i is an even number then encryption is done using Hill Cipher with key $K_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ (Recall in Hill cipher encryption of a plaintext x is given by $E_K(x) = xK$).
   - If i corresponding to ith encryption box E_i is an odd number then encryption is done using Vigenère Cipher with key $K_i = 1001001111001001$.

   Further assume that all the operations in the CBC-MAC given in Figure 3 are done in mod2.

   (a) (15 pts) Implement the CBC-MAC algorithm explained in the text above using Python/MATLAB.

   (b) (5 pts) Use your Python/MATLAB implementation to find the CBC-MAC of the message,
   $m = 1001100100111000110001010001111011001111101010100101101101011000011011100101$
   01111000000010001001