CS 409M : Introduction to Cryptography Fall 2024

Assignment 4
MAC and Hash

Instructor: Sruthi Sekar *TAs:* Giriraj Singh, Gyara Pragathi, Nilabha Saha

Problem 1: Lower Bound on Fixed-Length Tag Secure MAC

Say $\Pi = (\texttt{Gen}, \texttt{Mac}, \texttt{Vrfy})$ is a secure MAC, and for $k \in \{0,1\}^n$ the tag-generation algorithm $\texttt{Mac}_k$ always outputs tags of length $t(n)$. Prove that t must be super-logarithmic or, equivalently, that if $t(n) = \mathcal{O}(\log n)$ then Π cannot be a secure MAC.
Hint: Consider the probability of randomly guessing a valid tag.

Problem 2: Am I Secure?

Consider the following MAC for messages of length $\ell(n) = 2n-2$ using a pseudorandom function F: On input a message $m_0||m_1$ (with $|m_0| = |m_1| = n-1$) and key $k \in \{0,1\}^n$, algorithm $\texttt{Mac}$ outputs $t = F_k(0||m_0)||F_k(1||m_1)$. Algorithm $\texttt{Vrfy}$ is defined in the natural way. Is $(\texttt{Gen}, \texttt{Mac}, \texttt{Vrfy})$ secure? Prove your answer.

Problem 3: On Insecurity

Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. (In each case $\texttt{Gen}$ outputs a uniform $k \in \{0,1\}^n$). Let $\langle i \rangle$ denote an $n/2$-bit encoding of the integer i.)

a) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \cdots \oplus F_l(m_\ell)$.

b) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, compute $t := F_k(\langle 1 \rangle, m_1) \oplus \cdots \oplus F_l(\langle \ell \rangle, m_\ell)$.

c) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, choose uniform $r \leftarrow \{0,1\}^n$, compute

$$t := F_k(r) \oplus F_k(\langle 1 \rangle, m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle, m_\ell),$$

and let the tag be $\langle r, t \rangle$.

Problem 4: Could I *Be* More Insecure?

Let F be a pseudorandom function. Show that the following MAC for messages of length $2n$ is insecure: $\texttt{Gen}$ outputs a uniform $k \in \{0,1\}^n$. To authenticate a message $m_1||m_2$ with $|m_1| = |m_2| = n$, compute the tag $F_k(m_1)||F_k(F_k(m_2))$.

Problem 5: The CBC-MAC (Cousins) Forge

We explore what happens when the basic CBC-MAC construction is used with messages of different lengths.

a) Say the sender and receiver do not agree on the message length in advance (and so $\texttt{Vrfyk}(m,t) = 1$ iff $t \stackrel{?}{=} \texttt{Mac}_k(m)$, regardless of the length of m), but the sender is careful to only authenticate messages of length $2n$. Show that an adversary can forge a valid tag on a message of length $4n$.

b) Say the receiver only accepts 3-block messages (so $\texttt{Vrfy}_k(m,t) = 1$ only if m has length $3n$ and $t \stackrel{?}{=} \texttt{Mac}_k(m)$), but the sender authenticates messages of any length a multiple of n. Show that an adversary can forge a valid tag on a new message.

Problem 6: Strength Matters

Show that Construction 4.18 (from the book[1]) might not be CCA-secure if it is instantiated with a secure MAC that is not strongly secure.

Construction 4.18

Let $\Pi_E = (\texttt{Enc}, \texttt{Dec})$ be a private-key encryption scheme and let $\Pi_M = (\texttt{Mac}, \texttt{Vrfy})$ be a message authentication code, where in each case key generation is done by simply choosing a uniform n-bit key. Define a private-key encryption scheme $(\texttt{Gen}', \texttt{Enc}', \texttt{Dec}')$ as follows:

- $\texttt{Gen}'$: on input 1^n, choose independent, uniform $k_E, k_M \in \{0,1\}^n$ and output the key (k_E, k_M).
- $\texttt{Enc}'$: on input a key (k_E, k_M) and a plaintext message m, compute $c \leftarrow \texttt{Enc}_{k_E}(m)$ and $t \leftarrow \texttt{Mac}_{k_M}(c)$. Output the ciphertext $\langle c, t\rangle$.
- $\texttt{Dec}'$: on input a key (k_E, k_M) and a ciphertext $\langle c, t\rangle$, first check whether $\texttt{Vrfy}_{k_M}(c,t) \stackrel{?}{=} 1$. If yes, then output $\texttt{Dec}_{k_E}(c)$; if no, then output $\perp$.

Problem 7: ATE with CPA + EU-CMA

Prove that the authenticate-then-encrypt approach, instantiated with any CPA-secure encryption scheme and any secure MAC, yields a CPA-secure encryption scheme that is unforgeable.

[1] Introduction to Modern Cryptography, Second Edition - Jonathan Katz, Yehuda Lindell

Problem 8: OR Constructions

Let $(\texttt{Gen}_1, H_1)$ and $(\texttt{Gen}_2, H_2)$ be two hash functions. Define $(\texttt{Gen}, H)$ so that $\texttt{Gen}$ runs $\texttt{Gen}_1$ and $\texttt{Gen}_2$ to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x)||H_2^{s_2}(x)$.

a) Prove that if at least one of $(\texttt{Gen}_1, H_1)$ and $(\texttt{Gen}_2, H_2)$ is collision resistant, then $(\texttt{Gen}, H)$ is collision resistant.

b) Determine whether an analogous claim holds for second preimage resistance and preimage resistance, respectively. Prove your answer in each case.

Problem 9: Self-Composition of CRHF

Let $(\texttt{Gen}, H)$ be a collision-resistant hash function. Is $(\texttt{Gen}, \hat{H})$ defined by $\hat{H}^s(x) \stackrel{\text{def}}{=} H^s(H^s(x))$ necessarily collision resistant?

Problem 10: Let's Modify Merkle Damgård!

Recall the Merkle-Damgård transform (Construction 5.3 from the book):

Construction 5.3

Let $(\texttt{Gen}, h)$ be a fixed-length hash function for inputs of length $2n$ and with output length n. Construct the hash function $(\texttt{Gen}, H)$ as follows:

- $\texttt{Gen}$: remains unchanged.
- $\texttt{H}$: om input a key s and a string $x \in \{0,1\}^*$ of length $L < 2^n$, do the following:
  1. Set $B := \lceil \frac{L}{n} \rceil$ (i.e., the number of blocks in x). Pad x with zeroes so its length is a multiple of n. Parse the padded result as the sequence of n-bit blocks $x_1, \ldots, x_B$. Set $x_{B+1} := L$, where L is encoded as an n-bit string.
  2. Set $z_0 := 0^n$. (This is also called the IV.)
  3. For $i = 1, \ldots, B+1$, compute $z_i := h^s(z_{i-1}||x_i)$.
  4. Output z_{B+1}.

For each of the following modifications to the Merkle-Damgård transform, determine whether the result is collision resistant. If yes, provide a proof; if not, demonstrate an attack.

a) Modify the construction so that the input length is not included at all (i.e., output z_B and not $z_{B+1} = h^s(z_B||L)$). (Assume the resulting hash function is only defined for inputs whose length is an integer multiple of the block length.)

b) Modify the construction so that instead of outputting $z = h^s(z_B||L)$, the algorithm outputs $z_B||L$.

c) Instead of using an IV, just start the computation from x_1. That is, define $z_1 := x_1$ and then compute $z_i := h^s(z_{i-1}||x_i)$ for $i = 2, \ldots, B+1$ and output z_{B+1} as before.

d) Instead of using a fixed IV, set $z_0 := L$ and then compute $z_i := h^s(z_{i-1}||x_i)$ for $i = 1, \ldots, B$ and output z_B.