

Assignment 3

AE, MAC, Hash

Instructor: Sruthi Sekar

Problem 1 (Lower Bound on Fixed-Length-Tag MAC). Let $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ be an EU-CMA secure MAC, and for $k \in \{0, 1\}^n$ the tag-generation algorithm 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 an EU-CMA secure MAC.

Hint: Consider the probability of randomly guessing a valid tag.

Problem 2 (MAC constructions from PRF). Let $F = \{f_k : \{0, 1\}^n \rightarrow \{0, 1\}^n\}_{k \in \{0, 1\}^n}$ be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. In each case $\text{Gen}(1^n)$ outputs a uniform $k \in_R \{0, 1\}^n$ and Vrfy is the canonical one. Let $\langle i \rangle$ denote an $n/2$ -bit encoding of the integer i .

- a) $\text{Mac}_1(k, m)$: To authenticate a message $m = m_1, \dots, m_\ell$, where $m_i \in \{0, 1\}^n$, output $t := f_k(m_1) \oplus \dots \oplus f_k(m_\ell)$.
- b) $\text{Mac}_2(k, m)$: To authenticate a message $m = m_1, \dots, m_\ell$, where $m_i \in \{0, 1\}^{n/2}$, output $t := f_k(\langle 1 \rangle, m_1) \oplus \dots \oplus f_k(\langle \ell \rangle, m_\ell)$.
- c) $\text{Mac}_3(k, m)$: To authenticate a message $m = m_1, \dots, m_\ell$, where $m_i \in \{0, 1\}^{n/2}$, pick $r \in_R \{0, 1\}^n$, compute $t := f_k(r) \oplus f_k(\langle 1 \rangle, m_1) \oplus \dots \oplus f_k(\langle \ell \rangle, m_\ell)$, and output (r, t) .
- d) $\text{Mac}_4(k, m)$: To authenticate a message $m = m_1 || m_2$ with $|m_1| = |m_2| = n$, output the tag $t := f_k(m_1) || f_k(f_k(m_2))$.

Problem 3 (MAC Combiners). We want to build an EU-CMA secure MAC $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$, given two MACs $\Pi_1 = (\text{Gen}_1, \text{Mac}_1, \text{Vrfy}_1)$ and $\Pi_2 = (\text{Gen}_2, \text{Mac}_2, \text{Vrfy}_2)$ such that if at some time one of Π_1 or Π_2 is broken (but not both), then Π is still secure. I.e., we want Π to be secure if either Π_1 or Π_2 is secure.

- a) Show that the following Π is EU-CMA secure if either Π_1 or Π_2 is EU-CMA secure.
 - $\text{Gen}(1^n)$: Generates $k_1 \leftarrow \text{Gen}_1(1^n)$ and $k_2 \leftarrow \text{Gen}_2(1^n)$ and output $k = (k_1, k_2)$.
 - $\text{Mac}(k, m)$: output $(\text{Mac}_1(k_1, m), \text{Mac}_2(k_2, m))$.
 - $\text{Vrfy}(k, m, (t_1, t_2))$: output 1 iff $\text{Vrfy}_1(k_1, m, t_1) = 1$ and $\text{Vrfy}_2(k_2, m, t_2) = 1$.
- b) Suppose Π_1 and Π_2 are both deterministic MACs with canonical verification and both have tag space $\{0, 1\}^n$. Prove that the deterministic MAC Π defined below (with canonical verification) is EU-CMA secure if either Π_1 or Π_2 is EU-CMA secure:
 - $\text{Gen}(1^n)$: Generates $k_1 \leftarrow \text{Gen}_1(1^n)$ and $k_2 \leftarrow \text{Gen}_2(1^n)$ and output $k = (k_1, k_2)$.
 - $\text{Mac}(k, m)$: output $\text{Mac}_1(k_1, m) \oplus \text{Mac}_2(k_2, m)$.

Problem 4 (Domain Extension). Let $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ be a fixed-length deterministic MAC over $\mathcal{M}$ with canonical verification and with tag space $\{0, 1\}^n$. Each of the following schemes over $\mathcal{M}^2$ use Gen to generate the keys (independently, if invoked twice) and are deterministic with canonical verification. Show an EU-CMA attack for each of them:

- a) $\text{Mac}_1(k, (m_1, m_2)) := \text{Mac}(k, m_1) || \text{Mac}(k, m_2)$.
- b) $\text{Mac}_2(k, (m_1, m_2)) := \text{Mac}(k, m_1) \oplus \text{Mac}(k, m_2)$.
- c) $\text{Mac}_3((k_1, k_2), (m_1, m_2)) := \text{Mac}(k_1, m_1) || \text{Mac}(k_2, m_2)$.
- d) $\text{Mac}_4((k_1, k_2), (m_1, m_2)) := \text{Mac}(k_1, m_1) \oplus \text{Mac}(k_2, m_2)$.

Problem 5 (Cousins of CBC MAC). We explore what happens when the basic CBC-MAC construction is modified in different ways.

- a) Say the sender and receiver do not agree on the message length in advance and in CBC-MAC, the length is not pre-pended. Then, $\text{Vrfy}_k(m, t) = 1$ iff $t \stackrel{?}{=} \text{Mac}_k(m)$, regardless of the length of m . However, 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) Let F be a keyed function that is an EU-CMA secure (deterministic) MAC for messages of length n (i.e., $F_k = \text{Mac}_k$, for $k \in \{0, 1\}^n$). Note that F need not be a pseudorandom function. Show that CBC-MAC instantiated with this F is not necessarily EU-CMA secure. (**Hint:** try constructing a secure MAC, modifying the PRF-based construction, that helps with the attack)

Problem 6 (Hash Functions). Let (Gen_1, H_1) and (Gen_2, H_2) be two collision resistant hash functions. Answer the following questions.

- a) (*Hash Combiner.*) Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys k_1 and k_2 , respectively. Then define $H^{k_1, k_2}(x) = H_1^{k_1}(x) || H_2^{k_2}(x)$. Prove that if at least one of (Gen_1, H_1) and (Gen_2, H_2) is collision resistant, then (Gen, H) is collision resistant.
- b) (*Self Composition.*) Define (Gen_1, H) as $H^k(x) := H_1^k(H_1^k(x))$. Is it collision resistant?

Problem 7 (Merkle Damgård Transform). Recall the Merkle-Damgård transform:

Merkle-Damgård Transform

Let (Gen, h) be a fixed-length hash function for inputs of length $2n$ and with output length n . Construct the hash function (Gen, H) , where Gen remains unchanged and H is as follows:
H: on input a key k and a string $x \in \{0, 1\}^*$ of length $|x| < 2^n$, do the following:

1. Set $\ell := \left\lceil \frac{|x|}{n} \right\rceil$ (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 $x_1, \dots, x_\ell$. Set $x_{\ell+1} := |x|$, where $|x|$ is encoded as an n -bit string.
2. Set $z_0 := 0^n$. (This is also called the *IV*.)
3. For $i = 1, \dots, \ell + 1$, compute $z_i := h^k(z_{i-1} || x_i)$ and output $z_{\ell+1}$.

- a) (*MAC from Hash.*) Let (Gen, H) be a collision resistant hash function constructed via the Merkle-Damgård transform. Consider the following MAC for arbitrary-length messages:

- $\text{Gen}_M(1^n)$: $k \leftarrow \text{Gen}(1^n)$, $r \in_R \{0, 1\}^n$. output (k, r) .
- $\text{Mac}_{k,r}(m)$: output $H^k(r||m)$.

Show that this is not an EU-CMA secure MAC. Note here that the hash-key k is not hidden from the MAC adversary, only r is hidden!

- b) (*Modifications to MD transform.*) 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.
- Modify the construction so that the input length is not included at all (i.e., output z_ℓ and not $z_{\ell+1} = h^k(z_\ell||x|)$). (Assume the resulting hash function is only defined for inputs whose length is an integer multiple of the block length.)
 - Modify the construction so that instead of outputting $z = h^k(z_\ell||x|)$, the algorithm outputs $z_\ell||x|$.
 - Instead of using an IV , just start the computation from x_1 . That is, define $z_1 := x_1$ and then compute $z_i := h^k(z_{i-1}||x_i)$ for $i = 2, \dots, \ell + 1$ and output $z_{\ell+1}$ as before.
 - Instead of using a fixed IV , set $z_0 := |x|$ and then compute $z_i := h^k(z_{i-1}||x_i)$ for $i = 1, \dots, \ell$ and output z_ℓ .

Problem 8 (Insecure AE). Let $\Pi^{\text{cpa}} = (\text{Gen}^{\text{cpa}}, \text{Enc}^{\text{cpa}}, \text{Dec}^{\text{cpa}})$ be a CPA-secure encryption over $\mathcal{M}$ and with ciphertext space $\mathcal{C}$. Let $H_1 : \mathcal{M} \rightarrow \mathcal{T}$ and $H_2 : \mathcal{C} \rightarrow \mathcal{T}$ be collision resistant hash functions (for this question, just using a single hash function instead of a family). Prove that the following encryption schemes are not secure AE schemes. Both schemes use Gen^{cpa} to generate the key.

- $\text{Enc}_1(k, m)$: Generate $c \leftarrow \text{Enc}^{\text{cpa}}(k, m)$ and output $(c, H_1(m))$;
 $\text{Dec}_1(k, (c_1, c_2)) := \begin{cases} \text{Dec}^{\text{cpa}}(k, c_1), & \text{if } H_1(\text{Dec}^{\text{cpa}}(k, c_1)) = c_2 \\ \perp, & \text{otherwise} \end{cases}$
- $\text{Enc}_2(k, m)$: Generate $c \leftarrow \text{Enc}^{\text{cpa}}(k, m)$ and output $(c, H_2(c))$;
 $\text{Dec}_2(k, (c_1, c_2)) := \begin{cases} \text{Dec}^{\text{cpa}}(k, c_1), & \text{if } H_2(c_1) = c_2 \\ \perp, & \text{otherwise} \end{cases}$