CS 276: Homework 2

Due Date: Monday, Mar 9, 2026 at 8:59pm via Gradescope

Usage of LLMs/Generative AI tools is prohibited. Other online resources (textbooks/lecture notes) are permissible.

1. **(Insecurity of Same-Seed UOWHF Composition.)**

   Let $\mathcal{H}^{2*,*} = \{h_s : \{0,1\}^{2*} \to \{0,1\}^*\}_{s\in\{0,1\}^*}$ be a $(2*,*)$-UOWHF. Recall that in Step IV of the UOWHF construction (construction 5.4 in the lecture notes), we use independent seeds $s_1, s_2, \ldots$ at each application. Consider instead the following simpler variant where we reuse the *same* seed and apply h_s twice (this can be extended to t applications):

   $$H_s(x) \;=\; h_s\big(h_s(x)\big),$$

   where $x \in \{0,1\}^{4n}$ and $H_s : \{0,1\}^{4n} \to \{0,1\}^n$. (Here h_s first compresses $4n$ bits to $2n$ bits, then $2n$ bits to n bits.)

   Show that even if $\mathcal{H}^{2*,*}$ is a secure UOWHF, the family $\mathcal{H} = \{H_s\}_{s\in\{0,1\}^*}$ is **not** necessarily a secure UOWHF. That is, describe a family $\{h_s\}$ that is a UOWHF but for which the above construction is insecure.

   Solution **Construction.** Let $h_s : \{0,1\}^{2*} \to \{0,1\}^*$ be a UOWHF with $s \in \{0,1\}^n$. Define $\hat{h}_s : \{0,1\}^{2m} \to \{0,1\}^m$ (for any m) by parsing the input as $(x_1\|x_2)$ with $|x_1| = |x_2| = m$:

   $$\hat{h}_s(x_1\|x_2) \;=\; \begin{cases} h_s(0^m\|x_2) & \text{if } x_1 = s\|0^{m-n}, \\ h_s(x_1\|x_2) & \text{otherwise.} \end{cases}$$

   Claim 1: $\hat{h}_s$ is a UOWHF.

   We reduce to the UOWHF security of h_s. In the UOWHF game, the adversary $\mathcal{A}$ commits to some $x^* = (x_1^*\|x_2^*) \in \{0,1\}^{2m}$ *before* seeing s. Since s is uniform over $\{0,1\}^n$, we have $\Pr[x_1^* = s\|0^{m-n}] = 2^{-n}$, which is negligible. Condition on $x_1^* \neq s\|0^{m-n}$ (this holds with overwhelming probability). Then $\hat{h}_s(x^*) = h_s(x^*)$.

   Now suppose $\mathcal{A}$ produces $x' \neq x^*$ with $\hat{h}_s(x') = \hat{h}_s(x^*)$:

   - **Case $x_1' \neq s\|0^{m-n}$:** Both inputs are evaluated under h_s, so $h_s(x') = h_s(x^*)$ with $x' \neq x^*$—a collision on h_s.
   - **Case $x_1' = s\|0^{m-n}$:** Then $h_s(0^m\|x_2') = h_s(x^*)$, and $(0^m\|x_2')$ is a valid second preimage for h_s (it differs from x^* since $x_1^* \neq 0^m$ w.h.p.)—again a collision on h_s.

   In either case we break h_s, contradicting its UOWHF security.

   Claim 2: $\hat{H}_s(x) = \hat{h}_s(\hat{h}_s(x))$ is *not* a UOWHF.

   We exhibit an efficient adversary $\mathcal{B}$:

   (a) **Commit phase** (before seeing s): $\mathcal{B}$ outputs $x^* = 0^{4n}$.

   (b) **Find phase** (after receiving s): $\mathcal{B}$ sets $x' = (s\|0^n)\|0^{2n} \in \{0,1\}^{4n}$.

Verification that $\hat{H}_s(x^*) = \hat{H}_s(x')$:

Since $s \neq 0^n$ with overwhelming probability, $x' \neq x^*$. For the inner application, $\hat{h}_s$ maps $\{0,1\}^{4n} \to \{0,1\}^{2n}$, parsing inputs into two halves of length $2n$:

- **Inner hash of x^*:** $\hat{h}_s(0^{2n} \| 0^{2n})$. Since $0^{2n} \neq s\|0^n$ (w.h.p.), the "otherwise" branch applies: output is $h_s(0^{4n})$. Let $v = h_s(0^{4n}) \in \{0,1\}^{2n}$.
- **Inner hash of x':** $\hat{h}_s\big((s\|0^n)\|0^{2n}\big)$. The first half is $s\|0^n = s\|0^{2n-n}$, so the backdoor fires: output is $h_s(0^{2n}\|0^{2n}) = h_s(0^{4n}) = v$.

The inner hash values are identical: $\hat{h}_s(x^*) = \hat{h}_s(x') = v$. Therefore:

$$\hat{H}_s(x^*) = \hat{h}_s(v) = \hat{H}_s(x').$$

Thus $\mathcal{B}$ produces a valid collision with overwhelming probability, breaking the UOWHF security of $\hat{H}_s$. ■

2. **(MAC Security over Vector Spaces.)**

Let q be a prime with $q \geq 2^n$ and let $\mathbb{F}_q$ denote the field of integers modulo q. Consider a MAC scheme where messages are *nonzero* vectors $\mathbf{m} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}$ and tags are elements $t \in \mathbb{F}_q$. We define the following security notion.

Definition 0.1 *A* vector MAC scheme $\Pi = (\mathsf{Gen}, \mathsf{MAC}, \mathsf{Verify})$ *consists of:*

(a) $k \leftarrow \mathsf{Gen}(1^n)$*: outputs a key* k*.*

(b) $t \leftarrow \mathsf{MAC}(k, \mathbf{m})$*: on input key* k *and message* $\mathbf{m} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}$*, outputs a tag* $t \in \mathbb{F}_q$*.*

(c) $0/1 \leftarrow \mathsf{Verify}(k, \mathbf{m}, t)$*: outputs 1 iff* t *is a valid tag for* $\mathbf{m}$*.*

Definition 0.2 (Span-Forge Game: $\mathsf{SpanForge}_{\mathcal{A},\Pi}(n)$)

(a) ***Setup:*** *The challenger samples* $k \leftarrow \mathsf{Gen}(1^n)$*.* $\mathcal{A}$ *is given* 1^n*.*

(b) ***Queries:*** $\mathcal{A}$ *adaptively submits messages* $\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(q')} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}$*. For each query* $\mathbf{m}^{(i)}$*, the challenger returns* $t^{(i)} \leftarrow \mathsf{MAC}(k, \mathbf{m}^{(i)})$*.*

(c) ***Forgery:*** $\mathcal{A}$ *outputs* $(\mathbf{m}^*, t^*)$*. The output of the game is 1 if:*

$$\mathbf{m}^* \notin \mathrm{span}\{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(q')}\} \quad \textit{and} \quad \mathsf{Verify}(k, \mathbf{m}^*, t^*) = 1.$$

Definition 0.3 (Span-CMA Security) Π *is* Span-CMA-secure *if for all nu-PPT* $\mathcal{A}$*:* $\big|\Pr[\mathsf{SpanForge}_{\mathcal{A},\Pi}(n) = 1]\big| = \mathsf{negl}(n)$*.*

Construction. Let $F : \mathcal{K} \times [n] \to \mathbb{F}_q$ be a PRF. Define the vector MAC scheme Π as follows:

- $\mathsf{Gen}(1^n)$: Sample $k \xleftarrow{\$} \mathcal{K}$. Output k.
- $\mathsf{MAC}(k, \mathbf{m})$: Output $t = \sum_{i=1}^{n} m_i \cdot F(k, i) \in \mathbb{F}_q$.

- $\mathsf{Verify}(k, \mathbf{m}, t)$: Output 1 iff $t = \mathsf{MAC}(k, \mathbf{m})$.

(a) Prove that Π is Span-CMA-secure.

(b) Show that Π is **not** EUF-CMA-secure. That is, describe an efficient adversary that makes a single MAC query and produces a valid existential forgery.

(c) A natural attempt to fix the scheme is to add a constant term to break linearity. Consider the modified MAC:

$$\mathsf{MAC}'(k, \mathbf{m}) = F(k, 0) + \sum_{i=1}^{n} m_i \cdot F(k, i),$$

where $F(k, 0)$ is an additional PRF evaluation at a fixed input $0 \notin [n]$. Is MAC' EUF-CMA-secure? Justify your answer.

Solution

(a) We proceed via a hybrid argument.

Game 0 (Real game): The challenger uses key k and answers MAC queries with $t = \sum_i m_i \cdot F(k, i)$.
Game 1: The challenger replaces the PRF evaluations $F(k, 1), \ldots, F(k, n)$ with truly random values $r_1, \ldots, r_n \xleftarrow{\$} \mathbb{F}_q$. Tags become $t = \sum_i m_i \cdot r_i = \langle \mathbf{r}, \mathbf{m} \rangle$.
Claim: $|\Pr[\mathrm{Win}_0] - \Pr[\mathrm{Win}_1]| \leq \mathsf{negl}(n)$.
Proof: We build a reduction $\mathcal{B}$ to the PRF security of F. $\mathcal{B}$ queries its oracle $\mathcal{O}$ on inputs $1, \ldots, n$ to obtain values $v_1, \ldots, v_n$, and simulates the MAC game using tags $t = \sum_i m_i \cdot v_i$. When $\mathcal{O} = F(k, \cdot)$, this is Game 0; when $\mathcal{O}$ is a random function, this is Game 1.

Now, we show that $\Pr[\mathrm{Win}_1] \leq 1/q$. In Game 1, the vector $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{F}_q^n$ is uniformly random, and the tag for any message $\mathbf{m}$ is $t = \langle \mathbf{r}, \mathbf{m} \rangle$.
After querying messages $\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(q')}$ and receiving tags $t^{(j)} = \langle \mathbf{r}, \mathbf{m}^{(j)} \rangle$, the adversary learns the projection of $\mathbf{r}$ onto $V = \mathrm{span}\{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(q')}\}$.
More precisely, let $d = \dim(V)$. The tags impose d independent linear constraints on $\mathbf{r}$, so $\mathbf{r}$ remains uniformly distributed over an affine subspace of dimension $n - d$.
If the adversary outputs $\mathbf{m}^* \notin V$, then $\langle \mathbf{r}, \mathbf{m}^* \rangle$ is uniform over $\mathbb{F}_q$ conditioned on the adversary's view (since $\mathbf{m}^*$ has a nonzero component in $V^\perp$, over which $\mathbf{r}$ is uniform). Therefore:

$$\Pr[\mathrm{Win}_1] = \frac{1}{q} \leq \frac{1}{2^n} = \mathsf{negl}(n).$$

Hence, $\Pr[\mathsf{SpanForge}_{\mathcal{A},\Pi}(n) = 1] \leq \Pr[\mathrm{Win}_1] + \mathsf{negl}(n) \leq \frac{1}{2^n} + \mathsf{negl}(n) = \mathsf{negl}(n)$.

(b) The scheme is *not* EUF-CMA-secure. Consider the adversary $\mathcal{A}$ that:

i. Queries $\mathbf{m} = (1, 0, 0, \ldots, 0)$ and receives tag $t = F(k, 1)$.
ii. Outputs the forgery $(\mathbf{m}^*, t^*)$ where $\mathbf{m}^* = (2, 0, 0, \ldots, 0)$ and $t^* = 2t$.

Then $\mathsf{MAC}(k, \mathbf{m}^*) = 2 \cdot F(k, 1) = 2t = t^*$, so the forgery is valid. Note that $\mathbf{m}^* \neq \mathbf{m}$ (since $q \geq 2^n \geq 4$), so this is a legitimate existential forgery.

(c) MAC' is *not* EUF-CMA-secure. While the constant term $F(k,0)$ breaks homogeneity (so the scalar attack from part (b) no longer works), the scheme is still *affine* in $\mathbf{m}$, and affine combinations of queries that preserve the constant coefficient yield forgeries. Concretely, let $\mathbf{f} = (F(k,1),\ldots,F(k,n))$. Consider the adversary $\mathcal{A}$ that:

i. Queries $\mathbf{m}^{(1)} = \mathbf{e}_1 = (1,0,\ldots,0)$, receiving $t_1 = F(k,0) + F(k,1)$.

ii. Queries $\mathbf{m}^{(2)} = \mathbf{e}_2 = (0,1,0,\ldots,0)$, receiving $t_2 = F(k,0) + F(k,2)$.

iii. Outputs forgery $(\mathbf{m}^*, t^*)$ with $\mathbf{m}^* = 2\mathbf{e}_2 - \mathbf{e}_1 = (-1,2,0,\ldots,0)$ and $t^* = 2t_2 - t_1$.

This works because:

$$2t_2 - t_1 = 2\big(F(k,0)+\langle \mathbf{f}, \mathbf{m}^{(2)}\rangle\big) - \big(F(k,0)+\langle \mathbf{f}, \mathbf{m}^{(1)}\rangle\big) = F(k,0)+\langle \mathbf{f}, 2\mathbf{m}^{(2)} - \mathbf{m}^{(1)}\rangle = \mathsf{MAC}'(k,\mathbf{m}^*),$$

where $\mathbf{f} = (F(k,1),\ldots,F(k,n))$. Since $\mathbf{m}^*$ is distinct from both $\mathbf{m}^{(1)}$ and $\mathbf{m}^{(2)}$ for a generic choice (e.g., $\mathbf{m}^{(1)} = \mathbf{e}_1, \mathbf{m}^{(2)} = \mathbf{e}_2$ gives $\mathbf{m}^* = 2\mathbf{e}_2 - \mathbf{e}_1$), this is a valid existential forgery.

■