

## 6.5620 (6.875), Fall 2022

Homework # 2

Due: October 5 2022, 11:59:59pm ET

---

- **Typsetting:** You are encouraged to use L<sup>A</sup>T<sub>E</sub>X to typeset your solutions. You can use the following [template](#).
  - **Submissions:** Solutions should be submitted to Gradescope.
  - **Reference your sources:** If you use material outside the class, please reference your sources (including papers, websites, wikipedia).
  - **Acknowledge your collaborators:** Collaboration is permitted and encouraged in small groups of at most three. You must write up your solutions entirely on your own and acknowledge your collaborators.
- 

### Problems:

1. (4 points) **PRF or not?** Let  $\mathcal{F} = \{F_K : \{0, 1\}^n \rightarrow \{0, 1\}^n\}_{K \in \{0, 1\}^k}$  be a family of pseudorandom functions. For which of the following constructions is  $\mathcal{F}_c$  necessarily a family of pseudorandom functions? If  $\mathcal{F}_c$  is a family of pseudorandom functions, give a proof; otherwise, show a counterexample.

(a) (2 points)  $\mathcal{F}_0 = \{F_K^{(0)} : \{0, 1\}^{n-1} \rightarrow \{0, 1\}^{2n}\}_{K \in \{0, 1\}^k}$ , where

$$F_K^{(0)}(x) := F_K(0||x) || F_K(1||x) .$$

(b) (2 points)  $\mathcal{F}_1 = \{F_K^{(1)} : \{0, 1\}^{n-1} \rightarrow \{0, 1\}^{2n}\}_{K \in \{0, 1\}^k}$ , where

$$F_K^{(1)}(x) := F_K(0||x) || F_K(x||1) .$$

2. (9 points) **Faster GGM.** Let  $\mathcal{F} = \{F_s : \{0, 1\}^m \rightarrow \{0, 1\}^n\}_{s \in \{0, 1\}^n}$  be a family of PRFs (taking  $m$  bits to  $n$  bits, with  $n$ -bit keys) obtained by applying the GGM construction to any family of PRGs. We noted in class that the GGM construction is *highly sequential*: in order to evaluate  $F_s(x)$  on any input  $x$ , it is necessary to do  $m$  sequential evaluations of a PRG taking  $n$  bits to  $n$  bits. In this question, we will explore how to get a PRF family with the same input-output parameters ( $m$ -bit inputs,  $n$ -bit outputs) for which  $F_s(x)$  can be evaluated in only  $\log^2 m$  PRG evaluations, at the expense of some (tolerable) loss in security.

In this question, you may assume that  $m, n$  are both at least linear in some security parameter  $\lambda$ .

(a) (4 points) Let  $\mathcal{H} = \{H_k : \{0, 1\}^m \rightarrow \{0, 1\}^{\log^2 m}\}_{k \in \{0, 1\}^{p(m)}}$  be a family of functions<sup>1</sup>, for some  $p(m) = \text{poly}(m)$ . Note that any given  $H_k$  compresses  $m$  bits into  $\log^2 m$  bits. We say  $\mathcal{H}$  is *collision resistant* if no PPT adversary  $\mathcal{A}$  can win the following game with more than negligible probability:

- i. The challenger samples a key  $k \leftarrow \{0, 1\}^{p(m)}$  uniformly at random, and gives it to  $\mathcal{A}$ . (The assumption, as with PRFs, is that anybody can evaluate  $H_k$  efficiently if they have  $k$ .)
- ii.  $\mathcal{A}$  outputs two distinct strings,  $x_0 \neq x_1$ . It wins if and only if  $H_k(x_0) = H_k(x_1)$ .

Informally, collision resistant functions have the property that it is hard to find two inputs which evaluate to ('hash to') the same output under the function.

Assume that 1) secure length-doubling PRGs exist, and 2) collision resistant function families exist. Construct a PRF family  $\mathcal{F} = \{F_s : \{0, 1\}^m \rightarrow \{0, 1\}^n\}_{s \in \mathcal{S}}$  taking  $m$  bits to  $n$  bits such that, for any  $x$  and any  $s$ ,  $F_s(x)$  can be evaluated in only  $\log^2 m$  evaluations of the PRG and one evaluation of the collision-resistant function. (Your keys can be as long as you like, except that their length should be polynomial in  $m$  and  $n$ .) Show that your candidate construction is a secure PRF family. (Hint: you may find it easier to work with GGM as a black box during the security proof than to think about the paths explicitly.)

(b) (3 points) Unfortunately, it is not known how to construct collision-resistant hash functions from PRGs. We would like to do without the extra assumption—and, fortunately, we can!

As before, let  $\mathcal{H} = \{H_k : \{0, 1\}^m \rightarrow \{0, 1\}^{\log^2 m}\}_{k \in \{0, 1\}^{p(m)}}$  be a family of functions. We use the notation  $x \leftarrow_R \mathcal{S}$  to denote that  $x$  is sampled uniformly from the set  $\mathcal{S}$ . We say that  $\mathcal{H}$  is *pairwise independent* if, for any  $x, x' \in \{0, 1\}^m$ ,  $x \neq x'$ ,  $x \neq 0^m$ ,  $x' \neq 0^m$ , and any  $y, y' \in \{0, 1\}^{\log^2 m}$ ,

$$\Pr_{k \leftarrow_R \{0, 1\}^{p(m)}} [H_k(x) = y \text{ and } H_k(x') = y'] = \left( \frac{1}{2^{\log^2 m}} \right)^2.$$

We could also define the pairwise independence of  $\mathcal{H}$  in terms of a game, if a slightly trivial one:

- i. The adversary submits a tuple  $(x, x', y, y')$  to the challenger such that  $x, x' \in \{0, 1\}^m$ ,  $y, y' \in \{0, 1\}^{\log^2 m}$ ,  $x \neq x'$ ,  $x \neq 0^m$ ,  $x' \neq 0^m$ .
- ii. The challenger samples  $k \leftarrow_R \{0, 1\}^{p(m)}$  uniformly at random.

We say that  $\mathcal{H}$  is pairwise independent if the probability over the choice of  $k$  in step 2 that  $H_k(x) = y$  and  $H_k(x') = y'$  is *exactly*  $\left( \frac{1}{2^{\log^2 m}} \right)^2$ , no matter what  $(x, x', y, y')$  the adversary chose in the first step.

Define the family  $\mathcal{H}$  as follows: the key is a  $(\log^2 m) \times m$  matrix  $M$ , drawn uniformly at random from  $\{0, 1\}^{(\log^2 m) \times m}$ , and we define the hash function as

---

<sup>1</sup>Not necessarily a family of PRFs.

$H_M(x) = Mx$ , where the matrix multiplication is performed over the field  $\mathbb{F}_2$ . Show that this family  $\mathcal{H}$  is pairwise independent.

- (c) (2 points) Assume that secure length-doubling PRGs exist. Define a candidate construction for a PRF family  $\mathcal{F} = \{F_s : \{0, 1\}^m \setminus \{0^m\} \rightarrow \{0, 1\}^n\}_{s \in \mathcal{S}}$  taking  $m$  bits to  $n$  bits such that, for any  $x$  and any  $s$ ,  $F_s(x)$  can be evaluated in only  $\log^2 m$  PRG evaluations. Show that your candidate construction is a secure PRF family.
3. (9 points) **Let's Encrypt and Authenticate!** Let  $(\text{Gen}_{\text{Enc}}, \text{Enc}, \text{Dec})$  be an IND-CPA secure symmetric encryption scheme, and let  $(\text{Gen}_{\text{MAC}}, \text{Mac}, \text{Ver})$  be an EUF-CMA secure message authentication scheme. *You may assume in this problem that  $(\text{Gen}_{\text{Enc}}, \text{Enc}, \text{Dec})$  has perfect correctness.*

Suppose Alice and Bob share keys  $k_1 \leftarrow \text{Gen}_{\text{Enc}}$  and  $k_2 \leftarrow \text{Gen}_{\text{MAC}}$ , and they hope to transmit messages to each other in a *private* and *authenticated* way. Towards this end, they define a new algorithm **Transmit** which takes two keys,  $k_1$  and  $k_2$ , along with a message  $m$ , and purports to output an authenticated encryption of  $m$ . For each of the following definitions of **Transmit**:

- Construct algorithms  $\text{Dec}'$  and  $\text{Ver}'$  so that  $\mathcal{E}_1 = (\text{Gen}', \text{Transmit}, \text{Dec}')$  is a correct encryption scheme, and  $\mathcal{E}_2 = (\text{Gen}', \text{Transmit}, \text{Ver}')$  is a correct authentication scheme.
- Either prove  $\mathcal{E}_1$  is IND-CPA secure and  $\mathcal{E}_2$  is EUF-CMA secure via reductions, or provide an attack on at least one of the two.

Your algorithms  $\text{Dec}'$  and  $\text{Ver}'$  should be the ‘best possible’, in the sense that, if it is *possible* to define  $\text{Dec}'$  such that  $\mathcal{E}_1 = (\text{Gen}', \text{Transmit}, \text{Dec}')$  is IND-CPA secure for all possible secure  $(\text{Gen}_{\text{Enc}}, \text{Enc}, \text{Dec})$  (resp.  $\text{Ver}'$  such that  $\mathcal{E}_2 = (\text{Gen}', \text{Transmit}, \text{Ver}')$  is EUF-CMA secure for all possible secure  $(\text{Gen}_{\text{Enc}}, \text{Enc}, \text{Dec})$ ), then your definition of  $\text{Dec}'$  (resp.  $\text{Ver}'$ ) should achieve IND-CPA security (resp. EUF-CMA security).

For notational convenience, you may assume in this problem that:

- the length of the messages  $m$  accepted by **Transmit** is  $n$ ,
- the length of ciphertexts output by **Enc** on messages of length  $n$  is  $\ell_1$ ,
- the length of MACs output by **Mac** on messages of length  $n$  is  $\ell_2$ ,
- and the length of MACs output by **Mac** on messages of length  $\ell_1$  is  $\ell_3$ .

- (a) (3 points)  $\text{Transmit}(k_1, k_2, m) = (\text{Enc}(k_1, (m, \text{Mac}(k_2, m))))$ .
- (b) (3 points)  $\text{Transmit}(k_1, k_2, m) = (\text{Enc}(k_1, m), \text{Mac}(k_2, m))$ .
- (c) (3 points)  $\text{Transmit}(k_1, k_2, m) = (c := \text{Enc}(k_1, m), \text{Mac}(k_2, c))$ .

4. (9 points) **One-way (function) or another?** Let  $f$  be a length-preserving one-way function. For which of the following is  $f'$  necessarily a one-way function? If  $f'$  is a one-way function, give a proof; otherwise, show a counterexample. Your counterexamples must rely only on the existence of one-way functions.

(a) (2 points)  $f_0(x) = f(f(x))$ .

(b) (2 points)  $f_1(x, y) := f(x) \| f(x \oplus y)$ , where  $|x| = |y|$ .

- (c) (*2 points*)  $f_2(x) := f(x)|x_{[1:\log |x|]}$ , where the notation  $y_{[1:\ell]}$  denotes the string  $y$  restricted to its first  $\ell$  bits.

(d) (3 points)  $f_3(x) := f(x)_{[1:|x|-1]}$ .

5. (7 points) **This is a Bit Hard(core).**

(a) *Universally hardcore (3 points)*. Assume the existence of one-way functions. A polynomial time-computable predicate  $b : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be *universal* if for every one-way function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ ,  $b$  is hardcore. Prove that there is no universal hardcore predicate.

(Note that the Goldreich-Levin hardcore predicate  $\text{GL}(x, r) = \langle x, r \rangle \bmod 2$  from class is not universal since it is randomized. Equivalently, it only shows that for every one-way function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ , there is another one-way function  $f' : \{0, 1\}^{2n} \rightarrow \{0, 1\}^m$  for which  $\text{GL}$  is a hardcore predicate.)

(b) *Not one bit hardcore (4 points)*. Assuming the existence of one-way functions, show that there exists a one-way function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  for which  $b_i(x) = x_i$  is not hardcore for any  $i \in \{1, 2, \dots, n\}$ . Here,  $x_i$  denotes the  $i$ -th bit of the string  $x \in \{0, 1\}^n$ .