Problem 1 (Message Authentication, Bug Fix) 5 points

Let $\mathbb{F}$ be a finite field. In class, I defined the message authentication code

$$MAC : \mathbb{F}^2 \times \mathbb{F}^d \rightarrow \mathbb{F} : MAC(k, m) = \sum_{i=0}^{d-1} m_i x^i + y$$

with key $k = (x, y)$ and message $m = (m_0, \ldots, m_{d-1})$. I claimed that this is a statistically secure one-time with security $\varepsilon = \frac{d-1}{|\mathbb{F}|}$. Show that, this is not true. In fact, show that there exists messages $m \neq m' \in \mathbb{F}^d$ such that, given $MAC(K, m)$ for a uniformly random $K$ in $\mathbb{F}^2$, it’s possible to come up with $MAC(K, m')$ with probability 1.

The correct construction (it has now been corrected in the slides, notes) should have been:

$$MAC : \mathbb{F}^2 \times \mathbb{F}^d \rightarrow \mathbb{F} : MAC(k, m) = \sum_{i=1}^{d} m_i x^i + y$$

where $k = (x, y)$ and $m = (m_1, \ldots, m_d)$. The index $i$ should go from 1 to $d$ not 0 to $d-1$. This is a statistically secure one-time with security $\varepsilon = \frac{d}{q}$.

Where does the proof of security for the second construction fail with the first construction?

Problem 2 ($t$-wise independent hash) 10 pts

A hash function $h : \mathcal{K} \times \mathcal{U} \rightarrow \mathcal{V}$ is $t$-wise independent if for all $t$ distinct values $x_1, \ldots, x_t \in \mathcal{U}$ and any $y_1, \ldots, y_t \in \mathcal{V}$ we have

$$Pr[h(K, x_1) = y_1, \ldots, h(K, x_t) = y_t] = \prod_{i=1}^{t} Pr[h(K, x_i) = y_i] = \frac{1}{|\mathcal{V}|^t}$$

where $K$ is a random variable that’s uniform over $\mathcal{K}$.

Use the ideas we saw in class about polynomials over a finite field $\mathbb{F}$ (e.g., in the construction of one-time MACs and Shamir secret sharing) to construct such a scheme for any $t$ with $\mathcal{K} = \mathbb{F}^t$ and $\mathcal{U} = \mathcal{V} = \mathbb{F}$.

A $t$-wise independent hash function can be used as a statistically secure MAC which can be used to authenticate up to $t - 1$ messages. Explain why.
Problem 3 (Two-time Security?) 15 pts

We showed that the one-time pad is a perfectly secure “one-time” encryption scheme that allows us to encrypt a single message. In this problem, we want to define “two-time” encryption that can be used twice to encrypt two messages.

**Part A:** Here is a natural way to define two-time perfect secrecy for encryption. For any two pairs of messages \((m_0, m_1) \in \mathcal{M} \times \mathcal{M}\) and \((m'_0, m'_1) \in \mathcal{M} \times \mathcal{M}\) and for any ciphertexts \(c_0, c_1\) we have
\[
\Pr[\text{Enc}(K, m_0) = c_0, \text{Enc}(K, m_1) = c_1] = \Pr[\text{Enc}(K, m'_0) = c_0, \text{Enc}(K, m'_1) = c_1]
\]
Show that no encryption scheme can satisfy this definition.

**Part B:** To overcome the limitation in part A, we first relax the problem by considering statistical security where we require that for all \((m_0, m_1)\), \((m'_0, m'_1)\) \(\in \mathcal{M} \times \mathcal{M}\)
\[
\text{SD}(\text{Enc}(K, m_0), \text{Enc}(K, m_1)) \leq \varepsilon
\]
Show that, even with this relaxation, no encryption scheme with a deterministic encryption procedure can satisfy the above with \(\varepsilon < 1\).

We relax the problem further by considering randomized encryption schemes where, for a fixed \(k, m\) the encryption procedure \(\text{Enc}(k, m)\) can additional randomness to create the ciphertext. We require perfect correctness so that for all \(m \in \mathcal{M}, k \in \mathcal{K}\) : \(\Pr[\text{Dec}(k, \text{Enc}(k, m)) = m] = 1\) where the probability is over the randomness of the encryption procedure. Show that there exists a randomized encryption scheme that achieves the above for arbitrarily small \(\varepsilon\).

(Hint: Use \(t\)-wise independent hash functions from the previous problem with \(t = 2\). Let the encryption procedure call the hash function on a random input to derive a new “one-time pad” key on each invocation.)

Problem 4 (OWFs with Short Output Don’t Exist) 5 pts

Let \(f : \{0, 1\}^* \rightarrow \{0, 1\}^*\) be a function such that \(|f(x)| \leq c \log |x|\) for all \(x \in \{0, 1\}^*\) and for some fixed constant \(c > 0\). Show that \(f\) is not a one-way function.

Problem 5 (OWF or Not?) 20 pts

Assume that \(f : \{0, 1\}^* \rightarrow \{0, 1\}^*\) is a one-way function (OWF). For each of the following candidate constructions \(f'\) argue whether it is also necessarily a OWF or not. If yes, give a proof else give a counter-example (assuming one-way functions exist, show that there is a one-way function \(f\) such that \(f'\) is not a one-way function).

- \(f'(x) = (f(x), x[1])\) where \(x[1]\) is the first bit of \(x\).
- \(f'(x) = (f(x), x[1], \ldots, x[\lfloor n/2 \rfloor])\) where \(n = |x|\) and \(x[i]\) denotes the \(i\)th bit of \(x\).
- \(f'(x) = f(x) || 0\) where \(||\) denotes string concatenation.
- \(f'(x) = f(x) || f(x + 1)\) where \(||\) denotes string concatenation and \(x\) is interpreted as an integer in binary with addition performed modulo \(2^n\) for \(|x| = n\).
\[ f'(x) = f(G(x)) \] where \( G \) is a pseudorandom generator (with some polynomial stretch).

**Problem 6 (Pseudorandom Generators) 10 pts**

Let \( G \) be any candidate pseudorandom generator (PRG) with 1-bit stretch (i.e., when \(|x| = n, |G(x)| = n + 1\)). For any algorithm \( D \), we define the distinguishing advantage of \( D \) as

\[
\left| \Pr[D(G(U_n)) = 1] - \Pr[D(U_{n+1}) = 1] \right|
\]

where \( U_m \) denotes a uniformly random \( m \)-bit string.

- Construct an *inefficient* distinguisher \( D \) that has advantage \( 1/2 \).
- Construct an *efficient* (PPT) distinguisher \( D \) that has advantage \( 2^{-(n+1)} \).
- Generalize the above to show that for any time bound \( t(n) \leq 2^n \), there is a distinguisher \( D \) that runs in time \( t(n)\text{poly}(n) \) and has advantage \( t(n)2^{-(n+1)} \).

**Problem 7 (PRGs imply OWFs) 10 pts**

Show that if \( G : \{0,1\}^* \rightarrow \{0,1\}^* \) is a pseudorandom generator (PRG) with \( n \)-bit stretch, where \( n \) is the security parameter, then \( G \) is a one-way function.

**Problem 8 (PRG or Not?) 20 pts**

Assume that \( G : \{0,1\}^* \rightarrow \{0,1\}^* \) is a pseudorandom generator (PRG) with \( n \)-bit stretch. For each of the following candidate constructions argue whether it is also necessarily a PRG or not. If yes, give a proof else give a counter-example.

- \( G'(x) = G(x + 1) \) where addition is performed modulo \( 2^n \) for \( x \in \{0,1\}^n \).
- \( G'(x) = G(x||0) \) where \( || \) denotes string concatenation.
- \( G'(x) = G(x||G(x)) \).
- \( G'(x) = G(x) + x \) where we interpret \( x \) and \( G(x) \) as integers in binary and addition is performed modulo \( 2^{G(x)} \).
- \( G'(x) = G(f(x)) \) where \( f \) is a one-way function.