CS 601.442/642 – Modern Cryptography Instructor: Abhishek Jain

Midterm Exam

Deadline: October 20; 2020, 11:59 PM EST

Instructions

- All submissions must be made via Gradescope. No late submissions will be accepted.
- Please add the following declaration on the first page of your submission:

"I have neither given nor received any unauthorized aid on this exam. I understand that this exam must be taken without the aid of any other online resources ***besides*** *the lecture slides/videos, resources posted on the course website and the handouts sent via email. The work contained herein is wholly my own. I understand that violation of these rules, including using an unauthorized aid or collaborating with another person/student, may result in my receiving a 0 on this exam."*

1. **(10 points) One-Way Functions:** Let f be a function such that $f : \{0,1\}^n \to \{0,1\}^{\log n}$, where n is the security parameter. Show that f is *not* a one-way function.

2. **(15 points) Pseudorandom Generators:** Let $\mathbb{G}$ be a cyclic group of prime order q with generator g. Consider the following function $\mathsf{PRG} : \mathbb{Z}_q^3 \mapsto \mathbb{G}^5$.

$$\mathsf{PRG}(x, y_1, y_2) := (g^x, g^{y_1}, g^{xy_1}, g^{y_2}, g^{xy_2}),$$

where $x, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_q$ constitute the seed. Prove that the above construction is a secure Pseudorandom Generator.

3. **Pseudorandom Functions:** Let $\{f_k\}_k$ be a family of PRFs, where $f_k : \{0,1\}^n \mapsto \{0,1\}^n$ and $k \in \{0,1\}^n$. Let us consider the following two ways of increasing the input space of this family of functions from $\{0,1\}^n$ to $\{0,1\}^{2n}$ without increasing the key length:

   (a) **(10 points)** Let $g_k(x_1 \| x_2) = f_k(x_1) \| f_k(x_2)$. Show that the resulting family $\{g_k\}_k$ is *not* a secure family of PRFs.

   (b) **(15 points)** Let $g_k(x_1 \| x_2) = f_{f_k(x_1)}(x_2)$. Show that $\{g_k\}_k$ is a secure family of PRFs.

4. **(15 points) Key Exchange and Encryption:** Let $\mathsf{NIKE} = (\mathsf{Alice}, \mathsf{Bob}, \mathsf{ComputeAliceKey}, \mathsf{ComputeBobKey})$ be the tuple of algorithms associated with a non-interactive key exchange scheme defined as follows:

   - $(\mathsf{msg}_A, \mathsf{st}_A) \leftarrow \mathsf{Alice}(1^n)$: It takes the security parameter as input and outputs a message msg_A that Alice sends to Bob and Alice's private state st_A.
   - $(\mathsf{msg}_B, \mathsf{st}_B) \leftarrow \mathsf{Bob}(1^n)$: It takes the security parameter as input and outputs a message msg_B that Bob sends to Alice and Bob's private state st_B.
   - $\mathsf{key}_A \leftarrow \mathsf{ComputeAliceKey}(\mathsf{msg}_B, \mathsf{st}_A)$: It takes as input the message msg_B sent by Bob along with Alice's private state st_A and outputs a key key_A.

- $\mathsf{key}_B \leftarrow \mathsf{ComputeBobKey}(\mathsf{msg}_A, \mathsf{st}_B)$: It takes as input the message msg_A sent by Alice along with Bob's private state st_B and outputs a key key_B.

These algorithms satisfy the following two properties:

- **Correctness:** Let Alice and Bob's keys be computed as $\mathsf{key}_A \leftarrow \mathsf{ComputeAliceKey}(\mathsf{msg}_B, \mathsf{st}_A)$ and $\mathsf{key}_B \leftarrow \mathsf{ComputeBobKey}(\mathsf{msg}_A, \mathsf{st}_B)$ respectively, where $(\mathsf{msg}_A, \mathsf{st}_A) \leftarrow \mathsf{Alice}(1^n)$ and $(\mathsf{msg}_B, \mathsf{st}_B) \leftarrow \mathsf{Bob}(1^n)$. Then, it holds that

$$\Pr[\mathsf{key}_A = \mathsf{key}_B] = 1$$

- **Security:** Let the transcript of the NIKE scheme be $\mathsf{trans} := (\mathsf{msg}_A, \mathsf{msg}_B)$ where $(\mathsf{msg}_A, \mathsf{st}_A) \leftarrow \mathsf{Alice}(1^n)$ and $(\mathsf{msg}_B, \mathsf{st}_B) \leftarrow \mathsf{Bob}(1^n)$ and $r \xleftarrow{\$} \mathcal{K}$ be a uniformly sampled value from its key-space. Then, it holds that

$$(\mathsf{key}_A, \mathsf{trans}) \equiv (\mathsf{key}_A, \mathsf{trans}) \approx_c (r, \mathsf{trans})$$

Construct an IND-CPA secure public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ from $\mathsf{NIKE} = (\mathsf{Alice}, \mathsf{Bob}, \mathsf{ComputeAliceKey}, \mathsf{ComputeBobKey})$ and prove its security.