6.893: Cryptographic Techniques for Protecting Privacy **Fall 2020**

Problem Set 3

Due: October 16, 2020 at 5pm, Boston time via Gradescope.

Instructions: You **must** typeset your solution in LaTeX using the provided template:

`https://6893.csail.mit.edu/homework.tex`

Submission Instructions: You must submit your problem set via Gradescope. Please use course code **MGWNYV** to sign up. For ease of grading, please begin each *part* of each question on a new page.

Bugs: I make mistakes! If it looks like there might be a mistake in the statement of a problem, please ask a clarifying question on Piazza.

Problem 1: True/False [5 points].

(a) If $\mathsf{P} = \mathsf{NP}$, then two-party distributed point functions on n-bit domains with keys of size $O(n)$ exist.

(b) If $\mathsf{P} = \mathsf{NP}$, then two-party distributed point functions on n-bit domains with keys of size $O(2^n)$ exist.

(c) If secure PRPs exist, then two-party distributed point functions on n-bit domains with keys of size $O(n)$ exist.

(d) In the offline/online PIR scheme we covered in class, the servers need to store the database in preprocessed/encoded form.

(e) In the offline/online PIR scheme we covered in class, the left and right servers have the same running time.

Problem 2: Private Information Retrieval from Fully Homomorphic Encryption [10 points]. A fully homomorphic encryption (FHE) scheme is an encryption scheme (E, D) that—speaking very informally—allows computing arbitrary functions over encrypted data. That is, for any boolean function $f : \{0,1\}^\ell \to \{0,1\}$, there is an operation Eval_f such that if we sample an encryption key k from the keyspace of the FHE scheme then for any messages $m_1, \ldots, m_\ell \in \{0,1\}$, if

$$\mathsf{ct}_1 \leftarrow E(k, m_1), \cdots, \mathsf{ct}_\ell \leftarrow E(k, m_\ell),$$

then

$$D\big(k, \mathsf{Eval}_f(\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)\big) = f(m_1, \ldots, m_\ell).$$

(Formally defining FHE takes quite a bit of work. For example, to make the definition non-trivial, we need that the ciphertext size and decryption time are independent of the complexity of f. But I think that this informal definition should be enough for this problem, but please ask on Piazza if something seems off here.)

(a) Show how to use an FHE scheme to construct a single-server one-round PIR scheme in which the client uploads $O(\log n)$ ciphertexts to the server, for an n-bit database, and the server responds with a single ciphertext to the client.

(b) Say that the database contains n pairs $(\text{name}_i, \text{salary}_i)$. The client holds a string $\sigma \in \{0,1\}^\ell$ and wants to learn the sum of the salaries of the people whose names begin with the string σ. Use FHE to construct a single-server one-round PIR scheme in which the client uploads ℓ FHE ciphertexts and the server responds with $O(\log(n \cdot \texttt{SALARY\_MAX}))$ ciphertexts.

(c) Suppose that you have a *somewhat* homomorphic encryption scheme that supports computing boolean circuits consisting of only OR and AND gates of AND-depth $d+1$ (i.e., there are at most $d+1$ AND gates between any input and the output wire) for some constant $d \geq 2$. Construct a single-server one-round PIR scheme in which the client sends $O(n^{1/d})$ ciphertexts to the server and the server replies with a single ciphertext. Note that the constant in the big-O can depend on d. (It is actually sufficient to have an FHE scheme that supports circuits of AND-depth $O(\log d)$ but it is not required in this problem.)

Problem 3: More PIR servers ⇒ Less communication [15 points]. On the last problem set, you constructed a two-server PIR scheme with communication complexity $O(n^{1/3})$. In this problem, you will construct a $O(\log n)$-server PIR scheme with much much lower communication complexity polylog(n). The catch is that if an attacker can compromise any two servers, out of the $O(\log n)$ total servers, it can break client privacy. But, the stronger non-collusion assumption on the servers gives a much more communication-efficient information-theoretic PIR scheme.

For this problem let $\mathbb{F}$ be a finite field of size $\approx n$. If you are not familiar with finite fields, you can think of $\mathbb{F}$ as $\mathbb{Z}_p$ for a prime $p \approx n$. For this problem, all logarithms are base-two and n is a power of two.

The construction works in three steps:

1. First, we will construct a multivariate polynomial f_D that "encodes" the bits of the database D.
2. Next, we will show that it is possible to recover the value of this polynomial at any point by evaluating the polynomial at $O(\log n)$ points. Furthermore, each of these evaluation points will look random on its own (though the evaluation points will be correlated).
3. Finally, we will construct the PIR protocol.

(a) Show that for any bit $b \in \{0,1\} \subset \mathbb{F}$, there is a degree-one polynomial $f_b \in \mathbb{F}[x]$ (i.e., a polynomial in indeterminate x whose coefficients are in $\mathbb{F}$) such that

- $f_b(b) = 1$ and
- $f_b(1-b) = 0$.

Notice that $f_b(\cdot)$ is defined for all elements in the field $\mathbb{F}$, but that we don't care about its value on any points except 0 and 1.

(b) Show that for any string $b = b_1 b_2 \cdots b_m \in \{0,1\}^m \subset \mathbb{F}^m$, there is a polynomial $g_b \in \mathbb{F}[x_1, \ldots, x_m]$ such that

- $g_b(b_1, \ldots, b_m) = 1$ and

- $g_b(b'_1, \ldots, b'_m) = 0$, for any string $b' = b'_1 \cdots b'_m \in \{0,1\}^m \subset \mathbb{F}^m$ where $b' \neq b$.

Furthermore, g_b is multilinear: if you fix the values of $m-1$ variables, the polynomial is a linear function of the last variable. (In particular, $g_b(x_1, x_2, \ldots x_m)$ has no terms like $x_1^2 x_2$.)

Hint: Use your solution from Part (a) m times.

(c) Show that for any n-bit database $D = D_1 D_2 \cdots D_n \in \{0,1\}^n$, there is a multilinear polynomial $f_D \in \mathbb{F}[x_1, \ldots, x_{\log n}]$ such that for all $i \in [n]$, if $i_1 i_2 \cdots i_m \in \{0,1\}^{\log n}$ is the bit decomposition of i, it holds that $f_D(i_1, \ldots, i_m) = D_i \in \{0,1\} \subset \mathbb{F}$.

(d) Fix any point $\beta = (i_1, \ldots, i_{\log n}) \in \{0,1\}^{\log n} \subset \mathbb{F}^{\log n}$. We want to show that it is possible to interpolate the value $f_D(\beta) \in \mathbb{F}$ by evaluating f_D at $O(\log n)$ other points. (This is the magical part.)

Choose $\alpha \xleftarrow{R} \mathbb{F}^{\log n}$. Now, define the univariate polynomial $h \in \mathbb{F}^{\log n}[t]$ as $h(t) := \alpha t + \beta \in \mathbb{F}^{\log n}$. The notation $h \in \mathbb{F}^{\log n}[t]$ means that h is a polynomial in indeterminate t whose coefficients are $\log n$-tuples of elements of $\mathbb{F}$. So, the value $h(7)$ is a vector of $\log n$ elements of $\mathbb{F}$.

Observe that $h\colon \mathbb{F} \to \mathbb{F}^{\log n}$ and $f_D\colon \mathbb{F}^{\log n} \to \mathbb{F}$. So $(f_D \circ h) : \mathbb{F} \to \mathbb{F}$ is a univariate polynomial that maps $\mathbb{F} \to \mathbb{F}$. Furthermore, since h has degree one and f_D is linear in each of its $\log n$ variables, the composition $(f_D \circ h)$ is a univariate polynomial in indeterminate t of degree at most $\log n$.

Explain why having the values $f_D(h(1)), f_D(h(2), \ldots, f_D(h(1+\log n))$ is enough to recover the value $f_D(h(0)) = f_D(\beta)$. Notice that $f_D(\beta) = D_{i_1 \cdots i_n}$, which is the ith bit of the database.

(e) Use the result of Part (d) to construct a PIR protocol with $O(\log n)$ servers that provides security provided that no two servers collude. The client should send $O(\log n)$ field elements to each server and should receive one field element in return. The total communication complexity of this protocol will be polylog(n).

Problem 4: An anonymous feedback system [15 points]. I always appreciate it when students submit anonymous feedback via the online form. However, I would really like a way to give students extra credit for sending in feedback, without learning which student submitted which piece of feedback. In this problem, we will explore one possible way to do this.

The system works in two phases: a feedback phase and a grading phase. In both phases, the student interacts with the 6.893 server, which holds a secret key sk for a PRF $F(\cdot,\cdot)$.

- **Feedback phase.** The student connects to the 6.893 server (e.g., using Tor) and sends in their feedback. The student and server then run a protocol. At the end of the protocol, the student holds a token $t_{\mathsf{user}} = F(\mathsf{sk}, \mathsf{user})$, where user is the student's Kerberos identity. Over the course of the protocol, the server "learns nothing" about the student's identity.

- **Grading phase.** When the student submits her problem set to the 6.893, she includes her token t_{user}. I will check that the token is "valid:" that $t_{\mathsf{user}} = F(\mathsf{sk}, \mathsf{user}')$, where user' is the username of the student who submitted the problem set. If this check passes, I will assign the student some extra-credit points.

Let $\mathbb{G} = \{g, g^2, \ldots,\}$ be a group of prime order p, where $\mathbb{G}$ is the sort of group that we use for Diffie-Hellman key exchange. Let $H : \{0,1\}^* \to \mathbb{G}$ be a hash function, which we model as a random oracle.

(a) For this sort of system to be useful, it should be *correct*, in that an honest server interacting with an honest student will grant the student some extra-credit points at the end of this interaction. Please explain two other properties that this type of feedback system should satisfy. Your definitions can be informal.

(b) Say that instead of sending $F(\mathsf{sk}, \mathsf{user})$ to the student in the feedback phase, the feedback server sent $(r, t_r = F(\mathsf{sk}, r))$, for some per-student random value r sampled from the domain of the PRF. In the grading phase for this alternate system, the student submits (r, t_r) and I check only that $t_r = F(\mathsf{sk}, r)$. Show that, if we used this scheme, a pair of colluding students could cause a grading headache for the instructor. (In particular, the scheme will not satisfy one of your properties from part (a).)

(c) For the group $\mathbb{G}$ defined earlier in this problem, we can define a PRF $F_{\mathbb{G}}$ with keyspace $\mathbb{Z}_p$ and domain $\{0,1\}^*$ as $F_{\mathbb{G}}(\mathsf{sk}, x) := H(x)^{\mathsf{sk}} \in \mathbb{G}$. Give a two-message protocol that the student and server can run in the feedback phase such that:

- initially the server holds a PRF secret key sk and student holds a string $\mathsf{user} \in \{0,1\}^*$, and
- at the end of the protocol, the student holds $F_{\mathbb{G}}(\mathsf{sk}, \mathsf{user})$ and the server "learns nothing" about the client's string user. In particular, the view of the server in the protocol interaction should be independent of the client's string user.

(d) A different construction uses a PRF $F'_{\mathbb{G}}$ with keyspace $\mathbb{Z}_p$ and domain $\mathbb{G}$, defined as $F'_{\mathbb{G}} := x^{\mathsf{sk}} \in \mathbb{G}$. The only difference between $F_{\mathbb{G}}$ and $F'_{\mathbb{G}}$ is that the latter construction does not use the hash function H. Say that each student's Kerberos username is an element of the group $\mathbb{G}$. Show that if we instantiate the anonymous-feedback protocol with $F'_{\mathbb{G}}$ instead of $F_{\mathbb{G}}$, a student use one valid token for student user and convert it into a valid token for some other student $\mathsf{user}' \neq \mathsf{user}$. (Even if the student does not get to choose the student ID user' for while the other token is valid, this is still problematic.)

(e) **Extra credit [1 point]:** In some groups $\mathbb{G}$, the decision Diffie-Hellman problem is easy, while the discrete-log problem appears hard. (The group $\mathbb{Z}_p^*$, for a large prime p, is one such group.) Show that the scheme above is insecure in such a group. In particular, the scheme should not satisfy one of the properties from part (a).

(f) **Extra credit [5 points]:** Implement the client for this protocol (as a command-line utility) and the server (as a web app). If someone makes a nice implementation, I will use it to give extra-credit points for anonymous feedback for the rest of the semester.

Problem 5: Extra Credit: Current events [2 points]. Post an news article (from the last 14 days) on Piazza relating to privacy and/or cryptography and/or computer security. Put the number of the Piazza post as your answer here.

Problem 6: Extra Credit: Offline/Online PIR with Perfect Correctness [4 points]. In class we saw an offline/online PIR scheme with $O(\sqrt{n})$ communication and online time. The basic protocol we discussed in class fails (in correctness) with probability $O(1/\sqrt{n})$. One way to fix this is to repeat the protocol λ times in parallel to drive the failure probability down to $O(1/\sqrt{n}^{\lambda}) = O(2^{-\lambda})$. Show that you can modify the basic protocol to achieve *perfect correctness* (i.e., failure probability zero) without paying a λ factor in communication or online time.

Problem 7: Extra Credit: $k-1$-out-of-k Offline/Online PIR [6 points]. Construct an offline/online PIR scheme with $o(n)$ communication and online time such that for any $k \geq 2$, (a) the client communicates with k database replicas and (b) client privacy holds against an adversary that controls at most $k-1$ of the replicas.

Problem 8: Extra Credit (Research Problem): Two-server PIR [100 points]. Construct a two-server PIR protocol with information-theoretic security in which the total communication, on a database of n bits is polylog(n) bits.

Problem 9: Feedback [1 points].

(a) Roughly how many hours did you spend on this problem set?

(b) What was your favorite problem on this problem set? In one sentence: why?

(c) What was your least favorite problem on this problem set? In one sentence: why?

(d) **[Optional]** If you have any other feedback on this problem set or on the course, please write it here or submit it using the anonymous feedback form.