MIT 6.875J/18.425J and Berkeley CS276
Foundations of Cryptography (Fall 2020)

Problem Set 6: Released November 19, Due December 8

The problem set is due on **Tuesday, December 8 at 10pm ET/7pm PT**. Please make sure to upload to the Gradescope course webpage by the deadline (all registered students should have access to this webpage on Gradescope). Be sure to mark on Gradescope where each problem's solution starts. Typed solutions using LaTeX are strongly encouraged (template provided on the course webpage).
Collaboration is permitted; however, you must write up your own solutions individually and acknowledge all of your collaborators.

Problem 1. Voting System from Shamir's Secret-Sharing

In this problem, we will consider a voting system for an election with two candidates.

The proposed system will use Shamir's secret-sharing. The system consists of a committee of size n which collects and aggregates the votes. In particular, each voter chooses a number $s^{(v)}$ which is 0 if she wants to vote for candidate 0, and 1 if she wants to vote for candidate 1. Voter v submits its vote $s^{(v)} \in \{0,1\}$ by picking a random t-degree polynomial f_v over a finite field $\mathbb{F}$ with free coefficient $s^{(v)}$, and gives committee member i the value $s_i^{(v)} = f_v(i)$.

After the voting deadline is past, each committee member i computes

$$s_i^* = \sum_v s_i^{(v)}$$

Observe that $f^*(\cdot) = \sum_v f_v(\cdot)$ is a t-degree polynomial over $\mathbb{F}$ with free coefficient $s^* = \sum_v s^{(v)}$. Now, to reconstruct the secret, i.e. to compute s^* which is the total number of voters for party 1, any set of $t+1$ or more committee members can use their shares to interpolate the polynomial $f^*(\cdot)$, thus obtaining s^*.

For this problem, you may assume that there exists a public-ledger, and all parties have read/write access to this ledger. You can use any cryptographic tool learned in the class such as: a public-private-key encryption scheme, a digital signature scheme, a message authentication code (MAC), a (interactive or non-interactive) zero knowledge proof system, a homomorphic encryption scheme, and so on.

1.1 In the voting system described above, *cheating committee members* can prevent the reconstruction of the secret by contributing bad shares $\hat{s}_i^* \neq s_i^*$. Suggest a modification to this voting system which overcomes this attack. Argue why in your modified voting system this attack is not possible.

1.2 Additionally, in the voting system described above, *cheating voters* can submit shares which do not correspond to any t-degree polynomial $f_v(\cdot)$, thus causing different groups in the committee to reconstruct different "voting results," i.e. the value of s^* constructed is different for different groups. Moreover, cheating voters can submit shares of a polynomial with free coefficient $s^{(v)} \notin \{0,1\}$, thus influencing the "voting result" more than they should. Suggest how to further modify the scheme to overcome these problems as well. Argue why your modified voting system does not suffer from these problems.

Problem 2. Oblivious Transfer Schemes

Recall that in class, we learned about 1-out-of-2 Oblivious Transfer (OT) schemes. In the scheme, a sender has two messages $m_0, m_1 \in \{0,1\}$, and a receiver has an index bit b, and the sender wants to send m_b to the receiver while satisfying correctness (the receiver gets m_b), sender's privacy (the receiver gains no knowledge about m_{1-b}), and receiver's privacy (the sender gains no knowledge about b). For this problem (as in class), we focus on achieving security against *semi-honest* (or "honest-but-curious") senders and receivers.

2.1 Oblivious Transfer for long messages. You want to design a 1-out-of-2 OT scheme where the messages m_0 and m_1 have length $\ell = \ell(\lambda)$. You should think of ℓ as being a very large polynomial function of the security parameter λ. How can this be done, if you are only allowed to use the 1-out-of-2 OT scheme for single bit messages $\lambda \ll \ell$ times?

2.2 1-out-of-n Oblivious Transfer. Assume (e.g. from the previous part) that you have a 1-out-of-2 OT scheme that works for any message length ℓ and security parameter λ (where λ and ℓ are polynomially related). Show how to construct a 1-out-of-n OT scheme for any integer $n \geq 2$. In other words, the sender now has n messages $m_0, \ldots, m_{n-1} \in \{0,1\}^\ell$, and the receiver has an index $b \in \{0, 1, \ldots, n-1\}$, and the sender wants to send m_b to the receiver while satisfying correctness, sender's privacy, and receiver's privacy. How many times does your scheme call the 1-out-of-2 OT scheme?

Full credit will be given to solutions which use an asymptotically minimal number of calls (in terms of n). Security of your system should rely *only on* that of the underlying 1-out-of-2 OT scheme.

Problem 3. Private Information Retrieval

In this problem, we will consider *private information retrieval*. In this model, an array $x \in \{0,1\}^n$ is stored by one or more servers, and a user wants to a single bit of x, without the servers knowing which bit he wants.

We will consider a *two-server* model of PIR. Each server has a copy of x and will answer the user's queries, but the servers do not communicate with each other.

Consider the following two-server PIR protocols between user U and servers S_0 and S_1:

- The user chooses a random subset $T \subseteq \{1, 2, \ldots, n\}$, including each value independently with probability 1/2. (Note that T can be represented as an n-bit string w_T by letting the bit in position j represent whether or not $j \in T$.)
- The user calculates the set $T \oplus i$, which is $T \cup \{i\}$ if T does not contain i, and $T \setminus \{i\}$ if T does contain i.
- The user sends w_T to the server S_0 and $w_{T \oplus i}$ to the server S_1.
- Given a set, each server sends back the xor of all of the bits of x in that set.
- The user calculates the xor of the two responses from S_0 and S_1.

3.1 Prove that this scheme is *information-theoretically private*; that is, for any two different values i and i', the view of each server S_0, S_1 is perfectly indistinguishable, even if the servers are computationally unbounded. (It will be the case that the views of S_0 and S_1 together reveal i, but that is OK. Our assumption is that of non-collusion, that is S_0 and S_1 do not ever collude and put their views together.)

3.2 Unfortunately, the above protocol has a total *communication complexity* of $2(n+1)$ bits, because the user U sends an n-bit message and receives a one-bit response from each of the two servers. Note that there is a trivial protocol with communication complexity n that is also private: one of the servers simply sends the entire string x to the user U. We would like to reduce the amount of communication to significantly below n.

Alter the scheme from part 3.1 in such a way as to reduce the amount of communication to $o(n)$.

3.3 Unfortunately, in the above protocol each server needs to look at the entire database to answer every query. That's too computationally expensive.

Your goal is to alter the scheme from part 3.2 in such a way as to reduce the computation *per query* to $O(n/\log(n))$, while keeping the communication complexity the same. In order to do that, you are allowed to preprocess the database into a polynomially larger string in an *offline* phase, that is before receiving any

queries. This preprocessing could be computationally expensive, that is, it could take time polynomial in n. However, the computation in the *online* phase, that is, after receiving a query from the client, should take time $o(n)$. In particular, a solution that achieves online computational complexity $O(n/\log n)$ will receive full points.

Problem 4. Linear Regression MPC

In order to better understand the side-effects of cryptomania, m hospitals in the Boston and Berkeley areas want to compute a joint linear model on their combined patient data. That is, the i'th hospital has a $N \times d$ matrix $\mathbf{X}_i$, where each row corresponds to a patient and each column corresponds to an attribute, as well as a length N vector $\mathbf{y}_i$ of labels. Furthermore, $N \gg d$ (there have been many recent cryptomania diagnoses!) The hospitals' goal is to perform linear regression on their combined mN samples to obtain a model $\mathbf{z} \in \mathbb{R}^d$, without leaking additional information about their patients' data.

Because each hospital wishes to keep their own patient data private, simply collating all the data and using usual linear regression methods is not an option. Instead, the hospitals iteratively solve for m personal models $\mathbf{w}_i \in \mathbb{R}^d$ that are all very close to a global model $\mathbf{z} \in \mathbb{R}^d$, as follows:

1. For $i \in [m]$, let $\mathbf{A}_i = (\mathbf{X}_i^T\mathbf{X}_i + \rho\mathbf{I})^{-1}$ and $\mathbf{b}_i = \mathbf{X}_i^T\mathbf{y}_i$. Also initialize $\mathbf{u}_1^0, \ldots, \mathbf{u}_m^0, \mathbf{w}_1^0, \ldots, \mathbf{w}_m^0, \mathbf{z}^0 \leftarrow 0$.
2. For $k = 0 \ldots \texttt{numiters} - 1$, the following are computed for each $i \in [m]$:
   (a) $\mathbf{w}_i^{k+1} \leftarrow \mathbf{A}_i(\mathbf{b}_i + \rho(\mathbf{z}^k - \mathbf{u}_i^k))$
   (b) $\mathbf{z}^{k+1} \leftarrow S_{\lambda/m\rho}\left(\frac{1}{m}\sum_{i=1}^m(\mathbf{w}_i^{k+1} + \mathbf{u}_i^k)\right)$
   (c) $\mathbf{u}_i^{k+1} \leftarrow \mathbf{u}_i^k + \mathbf{w}_i^{k+1} - \mathbf{z}^{k+1}$
3. Output $\mathbf{z} \leftarrow \mathbf{z}^{\texttt{numiters}}$.

In the above, S_c is the soft thresholding operator, defined by

$$S_c(\alpha) = \begin{cases} \alpha - c & \alpha > c \\ 0 & -c \le \alpha \le c \\ \alpha + c & \alpha < -c, \end{cases}$$

and ρ is a constant that determines the size of the penalization for $\mathbf{w}_i$ deviating from z.

Using garbled circuits, design (and prove the security of) a secure protocol for the m hospitals to compute the model $\mathbf{z}$. Since cryptographic computation is often much slower than regular computation, your cryptographic protocol should run in time polynomial in d, m, `numiters`, and your security parameter, but it should not depend on N, except for computations done with unencrypted data. You may assume that all the hospitals are semi-honest, meaning that they will follow your protocol correctly but may collude and share information to learn information about another hospital, if possible. You may also assume that two of the hospitals, MGH and Mount Auburn, do not collude with anyone at all.

The parameters λ, ρ are both public and fixed.