Seminar Spring 2020 Notes - Private Information Retrieval

Vijaykrishna

Spring 2020

These are short summaries for the papers I presented in a seminar course guided by Prof. Mrinal Kumar. They are not self contained (due to the length) and must be used in conjunction with the actual papers.

1 Woodruff Paper: A Geometric Approach to Information-Theoretic Private Information Retrieval

1.1 Theorem 4

This is in relation to Theorem 4 and uses notation common to Theorem 3 and 4. Let $z \in \mathbb{F}_q^m$ be the point at which user U wants to evaluate $F(z)$. Let V be as in the paper i.e a random vector in $\mathbb{F}_q^m$. Define $G(\lambda) = \sum_{i=1}^{n} \left(x_i \prod_{E(i)_l=1} (z_l + \lambda V_l) \right)$. This is a single variable polynomial of degree $\leq 2k-1$. Server S_j can evaluate $G(\lambda_j)$ directly and send back one bit. For evaluating $G'(\lambda_j)$, notice that if you differentiate $G(\lambda)$ and substitute λ_j, you get a linear form in $V(i)$'s, so S_j can return these coefficients. Using Lemma 2 in the paper, U can reconstruct $G(\lambda)$ - and finally trivially get $G(0)$.

1.2 Theorem 7

Claim: There exists a t-private k-server PIR protocol with communication complexity $O(\frac{k^2}{t} \log(k) n^{\frac{t}{2k-1}})$.

Proof idea: The idea is to set $d = \frac{2k-1}{t}$ and consider $G(\lambda) = \sum_{i=1}^{n} \left(x_i \prod_{E(i)_l=1} (z_l + \lambda V_l^1 + \lambda^2 V_l^2 + .. + \lambda^t V_l^t) \right)$. Generate t independent random vectors $\{V^i\}_{i=1..t}$ and send $Q_i = (z_l + \lambda V_l^1 + \lambda^2 V_l^2 + .. + \lambda^t V_l^t)$ to the i^{th} database. Since $G(\lambda)$ is a polynomial of degree at most $2k-1$, one can reconstruct it from the k values of G and G' and then trivially output $G(0)$.

Remark: How to prove the privacy of this scheme? The paper says it is based on Shamir's secret sharing scheme. Essentially we need to show that the joint distribution function of any subset of size t is independent of i.

1.3 Lemma 10

Claim: Let F be a homogeneous degree-d polynomial in $\mathbb{F}_p[z_1, ..., z_m]$. Using $poly(m)$ preprocessing time, for all $V \in \mathbb{F}_p^m$, $F(V)$ can be computed with $O(\frac{m^d}{log^d m})$ work.

Proof: Partition $[m]$ into groups G_i of size $log(m)$ each. Call a choice D of groups as a way of choosing d groups in non descending order (repetition allowed). Now find the sum F_D of monomials (algebraically) $z_{i_1} .... z_{i_d}$ in F such that $i_k \in G_{i_k}$ for all $k = 1$ to d. F_D will be a polynomial of at most $d \log m$ variables, so constructing its truth table representation would take space and time of $p^{d \log m} = poly(m)$. The number of choices D possible is upper bounded by the ways of choosing without the constraint of non decreasing order i.e $\frac{m^d}{log^d m}$.

1.4 Theorem 9

Claim: There exists a 2-server PIR protocol with $O(n^{1/3})$ communication, $poly(n)$ preprocessing, and $O(n/log^r n)$ server work for any constant r.

Proof: We use $d = 2r + 1$. Suppose the database is viewed as a polynomial F and the user wants $F(z)$. Consider the derivative polynomials of F upto degree r. If the user wants to evaluate $G^{(k)}(\lambda_i)$ he would require the values of all mixed k^{th} derivatives evaluated at $z+\lambda V$. If we use the scheme in Lemma 9 for each of these derivative polynomials, we will get bad server work since the number of r^{th} derivative polynomials is m^r.
The fix to this is that is to partition the database of size n into t groups and store each group as a polynomial. Use lemma 10 on each of these polynomials. If F_u is the polynomial the user is interested in, he sends $z+V_u$, $z - V_u$ to the servers respectively. For all the other groups he sends V_i, $-V_i$ respectively. The server will substitute these values and return for a particular derivative, the sum (call them s_1, s_2) over all groups since it is not aware of u. Defining $H(\lambda) = G(\lambda) - G(-\lambda)$, it is easy to see that $H^{(k)}(1)$, $k = 1$ to r can be obtained due to the cancellations of the terms (details in the paper). Then defining $Q(\lambda^2) = H(\lambda)$ (H has only even powers since it was a difference), it is easy to inductively get r derivatives of Q from those of H. Since $deg(Q) \leq r$, we can uniquely find it and then we obtain $Q(0) = H(0) = 2G(0)$, from which $G(0)$ is obtained uniquely since we have a prime field with $p \geq 3$.

2 Gopi Paper: 2-Server PIR with sub-polynomial communication

2.1 The rings $\mathcal{R}_{m,r}$

We define the rings $\mathcal{R}_{m,r} = \frac{\mathbb{Z}_m[\gamma]}{\gamma^r - 1}$, which is essentially the set of polynomials of degree less than r with coefficients in $\mathbb{Z}_m$. Addition is the usual modulo addition and multiplication is performed modulo γ^r.

2.2 Matching vector families

Let $\mathcal{U} = \{\mathbf{u}_1, \cdots, \mathbf{u}_n\}$, $\mathcal{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$ where $\mathbf{u}_i$, $\mathbf{v}_i \in \mathbb{Z}_m^k$. $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ is called an S-matching vector family (where $0 \notin S$ and $S \subseteq [m]$) if the following condition holds:

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle = \begin{cases} 0 & \text{if } i = j \\ \in S & \text{if } i \neq j \end{cases}$$

Theorem [Gro99]: There is an explicitly constructible $\{1, 3, 4\}$-matching vector family $\mathcal{F}_6$ over $\mathbb{Z}_6^k$ of size $n = \Omega\left(\frac{\log^2 k}{\log \log k}\right)$.

2.3 The 2 server scheme:

$\{a_i\}_{i=1}^n$ is the binary string of length n denoting the database, present at both the servers.

Definition: Denote (symbolically) $\mathbf{x}^{\mathbf{c}} = x_1^{c_1} \cdots x_k^{c_k}$. Let $\mathcal{R}$ be a commutative ring and let $F(\mathbf{x}) = \Sigma c_{\mathbf{z}} \mathbf{x}^{\mathbf{z}} \in \mathcal{R}[x_1, \cdots, x_k]$. We define $F^{(1)} \in (\mathcal{R}^k)[x_1, \cdots, x_k]$ to be $F^{(1)}(\mathbf{x}) := \Sigma(c_{\mathbf{z}} \cdot \mathbf{z})\mathbf{x}^{\mathbf{z}}$.

Assumptions/notation:

1. Every polynomial or matrix we define will be over the ring $\mathcal{R} = \mathcal{R}_{6,6}$.
2. We use the matching vector family $\mathcal{F}_6$ over $\mathbb{Z}_6^k$ defined in Section 2.2. We want n to be the size of the database, so we will have $k = \exp\left(O\left(\sqrt{\log n \log \log n}\right)\right)$.
3. Define a polynomial $F(\mathbf{x}) = F(x_1, \cdots, x_k) = \Sigma_{i=1}^n a_i \mathbf{x}^{\mathbf{u_i}}$.
4. For $\gamma \in \mathcal{R}$, by the notation $\gamma^{\mathbf{z}}$, we refer to the argument list. $(\gamma^{z_1}, \cdots, \gamma^{z_k})$

Scheme: User $\mathcal{U}$ is interested in a_τ. The databases are $\mathcal{S}_1$, $\mathcal{S}_2$. Note that t_1, t_2 are fixed numbers (we use 0 and 1).

1. $\mathcal{U}$: Picks a uniformly random $\mathbf{z} \in \mathbb{Z}_6^k$
2. $\mathcal{U} \to \mathcal{S}_i : \mathbf{z} + t_i \mathbf{v}_\tau$
3. $\mathcal{S}_i \to \mathcal{U} : F(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau}), F^{(1)}(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$

Reconstruction: Define $G(t) := F(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}) = \sum_{i=1}^{n} a_i \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle + t\langle \mathbf{v}_\tau,\mathbf{u}_i\rangle}$. We can rewrite $G(t)$ as $G(t) = \sum_{l\in\{0,1,3,4\}} c_l \gamma^{tl}$, where $c_l = \sum_{i:\langle \mathbf{v}_\tau,\mathbf{u}_i\rangle = l} a_i \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle}$. In particular $c_0 = a_\tau \gamma^{\langle \mathbf{z},\mathbf{u}_\tau\rangle}$. We next define $g \in \mathcal{R}[T]$ as $g(T) := c_0 + c_1 T + c_3 T^3 + c_4 T^4$, and notice (with a few lines of pen and paper work using definitions) that

$$g(\gamma^t) = F(\gamma^{\mathbf{z}+t\mathbf{v}_\tau})$$
$$g^{(1)}(\gamma^t) = \langle F^{(1)}(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}), \mathbf{v}_\tau\rangle$$

Since $t_1 = 0$, $t_2 = 1$ were the values used, the user can obtain the values of g, $g^{(1)}$ at 1, γ. Thereby we get the matrix equation (in the unknown c_i's) as:

$$\begin{bmatrix} g(1) \\ g^{(1)}(1) \\ g(\gamma) \\ g^{(1)}(\gamma) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \\ 1 & \gamma & \gamma^3 & \gamma^4 \\ 0 & \gamma & 3\gamma^3 & 4\gamma^4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ c_4 \end{bmatrix} = M \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ c_4 \end{bmatrix}$$

With some standard calculations, $\det(M) = 3\gamma^5 + 4\gamma^4 + 3\gamma^3 + 2\gamma \in \mathcal{R}$ is non-zero. Denote the lhs vector by $\tilde{g}$. So, by pre-multiplying both sides with $\mathrm{adj}(M)$, we get $\mathrm{adj}(M) \cdot \tilde{g} = \det(M) \cdot c$. Recall that $c_0 = a_\tau \gamma^{\langle \mathbf{z},\mathbf{u}_\tau\rangle}$, so a_τ is zero $\iff$ $\det(M) \cdot c_0$ is zero $\iff$ the first element of $\mathrm{adj}(M) \cdot \tilde{g}$ is zero. So it suffices to compute $\mathrm{adj}(M) \cdot \tilde{g}$ to determine whether a_τ is 0 or 1.