Linear Locally Decodable Codes **Fall 2016**

Lecture 7: Private Information Retrieval

Lecturer: Zeev Dvir *Scribe: Kalina Petrova*

A *Private Information Retrival* protocol (PIR protocol) is a protocol that allows a user to extract an item from a database without revealing to the server which item has been retrieved. This is useful in many settings in which the privacy of the user is important.

Let us denote the data saved at the server with $\mathbf{a} \in \{0,1\}^s$. Now suppose the user wants to retrieve item $\mathbf{a}_i$ with high probability for some $i \in [s]$, without revealing i to the server. More formally, we would like a scheme such that the messages sent from a user to the server do not depend on i. We would like to minimize the *communication cost*, which is defined as the total number of bits sent from the user to the server and back.

Claim 7.1. All PIR protocols with one server such that the messages sent has no randomness in it and does not depend on i have communication cost at least s.

Proof. Suppose the communication cost is $N < s$. Then there are at most 2^N possible communication transcripts. However, there are $2^s > 2^N$ options for $\mathbf{a}$, which means that there are $\mathbf{a}' \in \{0,1\}^s$ and $\mathbf{a}'' \in \{0,1\}^s$, $\mathbf{a}' \neq \mathbf{a}''$, with identical communication transcripts. Then if $i \in [s]$ is such that $\mathbf{a}'_i \neq \mathbf{a}''_i$, if the user wants to find the i-th item, he/she would get the same information for $\mathbf{a}'$ as for $\mathbf{a}''$, which would be wrong.

□

Notice that we can trivially get a protocol with cost s if the server simply sends $\mathbf{a}$ to the user.

Exercise 7.1. Extend the proof above so that it holds even if the messages sent can have randomness in them.

There are two ways to overcome this barrier.

- Using cryptography: defining *computational privacy* and requiring not that the communication transcript does not depend on i, but that finding i is at least as hard for the server as some hard computational problem, for instance factoring.
- Using more than one server.

We will only deal with the second method here since it is closely related to Locally Decodable Codes.

Definition 7.1. An *r-server PIR* is a protocol $(U, S_1, \ldots, S_r)(\mathbf{a})$ between a user U and r servers $S_1, \ldots, S_r$ such that $\forall \mathbf{a} \in \{0,1\}^s$, $U(i)$ retrieves $\mathbf{a}_i$ with high probability (say $\geq \frac{3}{4}$) after communicating with $S_1, \ldots, S_r$ and such that $\forall j \in [r]$, the distribution of messages between U and S_j is independent of i.

The following is a more restrictive definition of PIR that captures all known constructions.

Definition 7.2. A *1-round PIR* is a protocol $(U, S_1, \ldots, S_r)(\mathbf{a})$ between a user U and r servers $S_1, \ldots, S_r$, where $S_1, \ldots, S_r$ get $\mathbf{a} \in \{0,1\}^s$ and U gets $i \in [s]$. Then U sends messages $(Q_1(i), \ldots, Q_r(i))$ to $S_1, \ldots, S_r$ respectively. Each $Q_j(i) \in \{0,1\}^L$ is a random variable uniformly distributed on $\{0,1\}^L$, but $(Q_1, \ldots, Q_r)$ is not necessarily uniform on $\{0,1\}^{L \times r}$. After that, each S_j answers deterministically with $A_j(Q_j(i), \mathbf{a}) \in \{0,1\}^R$. U outputs $O(A_1(Q_1(i), \mathbf{a}), \ldots, A_r(Q_r(i), \mathbf{a}), i)$, and with probability at least $\frac{3}{4}$, $O(A_1(Q_1(i), \mathbf{a}), \ldots, A_r(Q_r(i), \mathbf{a}), i) = \mathbf{a}_i$.

We will now reduce an r-query LDC with a special property to an r-server PIR, after which we will reduce an r-server PIR to an r-query LDC. The first reduction requires that each individual query in the LDC is completely uniform. All codes discussed in these lecture notes - Hadamard, Reed-Muller, MV-codes, have this property. We will call codes that have it *perfectly smooth* codes.

Theorem 7.1. Let $E : \{0,1\}^k \to \{0,1\}^n$ be a perfectly smooth r-query LDC. Then there exists an r-server PIR protocol with cost $\log n + 1$ and $s = k$.

Proof. Let U be the user and $S_1, \ldots S_r$ be the servers. Given $i \in [s]$, U will use the local decoder D of E to decode $\mathbf{a}_i$, sending each query location to a different server. Suppose the local decoder $D(i)$ needs to query locations $l_1, \ldots, l_r \in [n]$. $\forall j \in [r], U$ will send l_j to S_j and S_j will respond with $E(\mathbf{a})_{l_j}$. Since E is perfectly smooth, $\forall j \in [r], l_j$ is uniform, which implies privacy. Notice that $\forall j \in [r], l_j$ has length $\log n$ and $E(\mathbf{a})_{l_j}$ is one bit long, so the cost of the PIR protocol is $\log n + 1$.

□

Notice that the protocol we get has the extra property that each server's answer is only 1-bit long. When going from PIR to LDC, this is not necessarily the case, as can be seen from the following theorem.

Theorem 7.2. Suppose $(U, S_1, \ldots, S_r)$ is a 1-round PIR protocol with $|Q_j| = L$ and $|A_j| = R$. Then there exists an r-query LDC $E : \{0,1\}^k \to \Sigma^n$ with $\Sigma = \{0,1\}^R$, $n = r2^L$, and $k = s$.

Proof. Define $E(\mathbf{a})$ to be the string of S_1's answers to all possible questions of U, followed by S_2's answers to all possible questions of U, ..., followed by S_r's answers to all possible questions of U. Thus we have that $E(\mathbf{a}) \in \Sigma^{r \times 2^L}$, where $\Sigma = \{0,1\}^R$, since there are r

servers, 2^L possible questions of U and each answer has length R. Then the local decoder D simulates the behavior of U from the PIR protocol by randomly permuting the servers. Thus, D will get $D(i) = \mathbf{a}_i$ with probability at least $\frac{3}{4}$.

$\square$

Corollary 7.1. There exists a 3-server PIR protocol with cost $s^{o(1)}$.

Proof. By Corollary 5.2 and the comment under it, there exists a 3-query LDC $E : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with $n = 2^{k^{o(1)}}$. As can be seen from the proof of Theorem 5.1, any LDC obtained from a Matching Vector family is perfectly smooth, therefore E is perfectly smooth. Then by Theorem 7.1, there exists a 3-server PIR protocol with $s = k$ and cost $\log n + 1 = s^{o(1)}$.

$\square$

The next question we can ask ourselves is whether we can get a 2-server PIR. If we restrict ourselves to using $\Sigma = \{0, 1\}$ as alphabet for the respective LDC, then we cannot get cost better than $\Omega(s)$, where $\mathbf{a} \in \{0, 1\}^s$. This is for the following reason. Suppose that there is a 1-round 2-server PIR protocol with cost $C < \Omega(s)$. If L is the maximum length of a question asked by the user, and R is the maximum length of an answer given by a server, then $L + R = C$. Now by Theorem 7.2, there exists a 2-query LDC $E : \{0, 1\}^k \to \{0, 1\}^n$, where $n = 2 \times R \times 2^L < 2^{C+1}$, and $k = s$. Since $C < \Omega(s), n < 2^{\Omega(k)}$. But by Theorem 2.1, this is impossible. Therefore, there is no 1-round 2-server PIR protocol with cost better than $\Omega(s)$ and alphabet $\{0, 1\}$.

Notice, however, that there are 2-query LDCs with large Σ and small encoding length.

Theorem 7.3. There exists a 2-server PIR with cost $s^{\frac{1}{3}}$.

Proof. Suppose $p > 3$ is prime. Let ℓ and s be such that $s = \binom{\ell+3}{3}$ and let $S = \{\mathbf{v}_1, \dots, \mathbf{v}_s\} \subseteq \mathbb{F}_p^\ell$ be a Minimal Interpolating Set for degree ≤ 3 polynomials $f \in \mathbb{F}_p[\mathbf{x}_1, \dots, \mathbf{x}_\ell]$. Thus, $\forall \mathbf{a} \in \mathbb{F}_p^s$, $\exists$ a polynomial of degree at most 3 $f_\mathbf{a} \in \mathbb{F}_p[\mathbf{x}_1, \dots, \mathbf{x}_d]$ such that $\forall i \in [s], f_\mathbf{a}(\mathbf{v}_i) = \mathbf{a}_i$. Now we will describe the behavior of U when it gets i as an input. $U(i)$ operates in the following way. It picks a random $\mathbf{b} \in \mathbb{F}_p^\ell$ and considers the line $L = \{\mathbf{v}_i + t \cdot \mathbf{b} | t \in \mathbb{F}_p\}$. Then it sends to the two servers the following questions: $Q_1(i) = \mathbf{v}_i + \mathbf{b}$ and $Q_2(i) = \mathbf{v}_i + 2 \cdot \mathbf{b}$. Next we describe the behavior of the servers. A server takes as input the message it receives from the user $\mathbf{y}$ and the data $\mathbf{a}$. We have that $A_1(\mathbf{y}, \mathbf{a}) = A_2(\mathbf{y}, \mathbf{a}) = \left(f_\mathbf{a}(\mathbf{y}), \frac{\delta f_\mathbf{a}}{\delta \mathbf{x}_1}(\mathbf{y}), \dots, \frac{\delta f_\mathbf{a}}{\delta \mathbf{x}_\ell}(\mathbf{y})\right)$. Upon receiving $A_1(Q_1(i), \mathbf{a})$ and $A_2(Q_2(i), \mathbf{a})$, U aims at finding $\mathbf{a}_i$. The way to do this is as follows. Let $g(t) = f_\mathbf{a}(\mathbf{v}_i + t \cdot \mathbf{b})$. Then g is a degree 3 univariate polynomial. We want to find $g(0) = f_\mathbf{a}(\mathbf{v}_i) = \mathbf{a}_i$. We have that

$$\begin{aligned}
g(1) &= f_{\mathbf{a}}(\mathbf{v}_i + \mathbf{b}) \\
g(2) &= f_{\mathbf{a}}(\mathbf{v}_i + 2\mathbf{b}) \\
g'(1) &= \Big(\frac{df_{\mathbf{a}}(\mathbf{v}_i + t\mathbf{b})}{dt}\Big)\Big|_{t=1} = \sum_{j=1}^{l} \Big(\frac{\delta f_{\mathbf{a}}(\mathbf{v}_i + \mathbf{b})}{\delta \mathbf{x}_j}\Big) \cdot \mathbf{b}_j \\
g'(2) &= \Big(\frac{df_{\mathbf{a}}(\mathbf{v}_i + t\mathbf{b})}{dt}\Big)\Big|_{t=2} = \sum_{j=1}^{l} \Big(\frac{\delta f_{\mathbf{a}}(\mathbf{v}_i + 2\mathbf{b})}{\delta \mathbf{x}_j}\Big) \cdot \mathbf{b}_j
\end{aligned}$$

U has the values of $g(1), g(2), g'(1)$, and $g'(2)$ since they can be calculated from the answers it gets from the two servers. From $g(1), g(2), g'(1)$, and $g'(2)$, U can recover g completely, since it is a polynomial of degree 3 and therefore has 4 coefficients, so 4 values are enough to reconstruct it.

Now we calculate the cost of this protocol. The user sends $l \cdot \log p \approx s^{\frac{1}{3}}$ bits since $p = O(1)$. Servers answer with $\log_e p \cdot (l+1) \approx s^{\frac{1}{3}}$ bits.

□

Corollary 7.2. There exists a 2-query LDC $E : \{0,1\}^k \to \Sigma^n$ with $\Sigma = \{0,1\}^{k^{\frac{1}{3}}}$ and $n = 2^{k^{\frac{1}{3}}}$.

A more recent result [DG15] shows the existence of a 2-server PIR with cost $s^{o(1)}$ using Matching Vector codes and partial derivatives.

References

[DG15] Zeev Dvir and Sivakanth Gopi. 2-server PIR with sub-polynomial communication. *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing (STOC '15)*, pages 577–584, 2015.