Cryptography Meets Algorithms (15893) Lecture Notes

Lecture 7: Preprocessing PIR Lower Bounds

Scribe: William Seo

February 27, 2024

In this lecture, we will prove a lower bound about the client space and server computation tradeoff for preprocessing PIR schemes. We will borrow techniques for proving time-space tradeoff from the complexity theory literature.

Specifically, we consider a 1-server preprocessing PIR scheme with the following syntax:

- *Preprocessing algorithm.* Suppose there is a (possibly randomized and unbounded) preprocessing function denoted $\mathsf{Prep} : \{0,1\}^n \to \{0,1\}^S$ that takes in an n-bit database $\mathsf{DB} \in \{0,1\}^n$ as input, and outputs an S-bit hint string denoted h.
- *A single query.* The client and the server perform a (possibly randomized) query protocol. The client takes in h and some index $i \in [n]$ as input, and the server takes the database $\mathsf{DB} \in \{0,1\}^n$ as input. To answer the query, the server is allowed to read at most T locations of the database.

In other words, we do not place any restriction on the amount of work performed during preprocessing. The only constraints are that at the end of the preprocessing: 1) the client is allowed to store only the hint h; and 2) the server is allowed to store only the original database DB and no extra information.

We assume perfect correctness, i.e., for any $\mathsf{DB} \in \{0,1\}^n$, any query $i \in [n]$, correctness holds with probability 1. Let $\mathsf{view}_S(\mathsf{DB}, i)$ denote the server's view during the query phase when we run the PIR scheme (i.e., preprocessing followed by the query protocol) over inputs DB and i. For security, we require that for any $\mathsf{DB} \in \{0,1\}^n$, any $i, j \in [n]$, $\mathsf{view}_S(\mathsf{DB}, i) \approx \mathsf{view}_S(\mathsf{DB}, j)$ where $\approx$ means computational indistinguishability.

For such a preprocessing PIR scheme, we will prove a tradeoff between S and T as stated in the following theorem:

Theorem 1 (Time space tradeoff for preprocessing PIR [CGK20])**.** *Given a 1-server preprocessing PIR, let S be the client space, and let T be the server compuation per query. Then, $(S+1)(T+1) \geq N$. [elaine: TODO: edit the theorem based on what we can prove later]*

Again, note that the lower bound holds even when we allow the preprocessing to be unbounded, even when there is only a single query after the preprocessing, and even when the query phase may have arbitrarily many rounds of interaction.

Piano. Recall that in an earlier lecture, we covered Piano [ZPSZ23], a preprocessing 1-server PIR scheme. Piano enjoys the following performance bounds:

- Client space: $\widetilde{O}(\sqrt{n})$ where $\widetilde{O}(\cdot)$ hides a(n arbitrarily small) superlogarithmic function.

- Communication per query: $O(\sqrt{n})$
- Server computation per query: $O(\sqrt{n})$

We can see that Piano achieves optimal client space and server computation tradeoff (up to polylogarithmic factors) in light of Theorem 1.

1 Warmup: Yao's Box Problem

Before proving Theorem 1, we first prove a time-space tradeoff for a classical problem called the Yao's box problem [Yao90], and we shall see why Yao's box problem is closely related to preprocessing PIR.

We have a server with n boxes, each covering a bit. Henceforth we use $\mathsf{DB} := (\mathsf{DB}[1], \ldots, \mathsf{DB}[n])$ to denote the n bits. Consider the following game:

- *Preprocessing.* During a preprocessing phase, the server and client can perform an unbounded amount of computation. At the end of preprocessing, the client obtains an S-bit hint h; the server does not store any extra information besides DB itself where each bit is covered by some box.
- *Query.* The client wants to find out $\mathsf{DB}[i]$ for some index $i \in [n]$. The client and server now engage in some protocol at the end of which the client outputs an answer β. The query protocol must satisfy two requirements: 1) the server is *not allowed to open box i* during the protocol; and 2) the server can open at most T boxes.

We allow the preprocessing algorithm and query protocol to be possibly randomized. We require that the protocol to have *perfect correctness*, i.e., for any $\mathsf{DB} \in \{0,1\}^n$, any index $i \in [n]$, after performing the preprocessing and the query protocol for index i, the client's output $\beta = \mathsf{DB}[i]$ with probability 1.

Note that unlike PIR, Yao's box problem does not have any privacy requirement. In particular, it is perfectly ok for the query index i to be leaked to the server.

In the above formulation, S denotes the client space at the end of preprocessing, and T can be viewed as a lower bound on the server's running time during the query phase. We care about characterizing the tradeoff between S and T, that is, the client-space and server-time tradeoff.

Yao [Yao90] proved the following theorem:

Theorem 2 (Yao's box problem)**.** *For any protocol that solves Yao's box problem, it must be that $S \cdot (T+1) \geq n$, where S is the client space and T is the maximum number of boxes opened during the query.*

1.1 Upper Bound

Before proving Theorem 2, let us first see a simple upper bound that can match the $S \cdot (T+1) = n$ tradeoff. For simplicity, assume n is a perfect square.

1. Divide the n boxes into $\sqrt{n}$ segments each of size $\sqrt{n}$.
2. Preprocessing: the client stores the parity of each segment, denoted $p_1, \ldots, p_{\sqrt{n}}$ respectively.
3. Query for index $i \in [n]$: suppose i belongs to the j-th segment. The server opens every other box except i in the j-th segment, and responds with the parity (denoted p^*) of all opened bits. The client reconstructs the answer as $p^* \oplus p_j$.

In this construction, $S = \sqrt{n}$ and $T = \sqrt{n} - 1$.

You might have observed that this upper bound for Yao's box problem is reminiscient of the Piano PIR scheme [ZPSZ23].

1.2 Lower Bound

We now prove Theorem 2. The intuition of the proof is if we can obtain a too-good-to-be-true tradeoff between S and T, then we can construct an encoding of $\mathsf{DB} \in \{0,1\}^n$ whose length is less than n, which violates Shannon's fundamental theorem of information theory. Specifically, we can obtain some compression in the encoding by leveraging the following fact: every time we open T boxes not including i, we learn not just the values under the T opened boxes but also an additional bit, namely, the i-th bit — this effectively gains us one bit of advantage.

The formal proof works as below.

Proof of Theorem 2. Suppose we have some protocol for solving Yao's box problem with parameters S and T. We fix all the coins in the protocol denoted coins, and we will give coins as input to both the encoder and decoder. As long as DB is randomly sampled independently of coins, we can use Shannon's theorem to argue that the encoding length must be at least n.

Encoding. The encoding algorithm is given coins and DB as input, and constructs an encoding of DB as follows.

1. Seed the protocol (for solving Yao's box problem) with coins, and perform preprocessing such that the client obtains an S-bit hint h.
2. Initially, let $\mathsf{known} = \{\}$. Repeat the following: in each time step i, the client
   - finds the smallest index $q_i \notin \mathsf{known}$, and runs the query protocol for index q_i;
   - let $\mathsf{known} \leftarrow \mathsf{known} \cup \{q_i\} \cup \{\text{all boxes opened during this query}\}$
   - if $\mathsf{known} = [n]$, break.

Define the *newly* opened boxes in time step i as follows: all boxes opened in time step i that are not in the known set yet. Output an encoding containing the following terms:

1. The client's hint h;
2. Let V_i be the *newly* opened values (in the order they are opened) for time step i. Include $V_1, V_2, \ldots, V_k$ in the encoding, where k is the total number of time steps.

Decoding. The decoding algorithm is given coins and an encoding C as input, and outputs a decoded string as follows.

1. Seed the protocol (for solving Yao's box problem) with coins. Treat the first S bits of C as the client's hint, and in the steps below, consume the rest of C bit by bit in a streaming manner.
2. Initially, let $\mathsf{known} = \emptyset$. Every time step i,
   - Let q_i be the smallest index not in known;

- The client performs a query for q_i with the server. Whenever the server needs to open some box: if the box was already opened in some earlier time step, use the same opened value as previously learned; else if the box has not been opened, treat the next bit in C as the opened value.
- For all newly opened boxes, record the opened values. Additionally, reconstruct the value at queried index q_i.
- Let $\mathsf{known} \leftarrow \mathsf{known} \cup \{q_i\} \cup \{\text{all boxes opened during this query}\}$; if $\mathsf{known} = [n]$, break.

3. At the end of the protocol, $\mathsf{known} = [n]$, i.e., the values for n indices have been discovered. Output this reconstructed string.

□

Correctness of the decoding is easy to verify: the decoding algorithm opens the same sequence of boxes as the encoding algorithm, and for every newly opened box, decoding observes the correct value from C.

Let $t_1, \ldots, t_k$ be the number of *newly* opened boxes in each of the k time steps. Thus, the number of elements added to known in time step i is $t_i + 1$. We also know that $t_i \le T$ for all $i \in [k]$, and regardless of the choice of DB. Henceforth, let $t = \frac{1}{k}\sum_{i\in[k]} t_i$. Since the the encoding/decoding algorithm stops as soon as $\mathsf{known} = [n]$, it means that $\sum_{i\in[k]}(t_i + 1) = (t+1)\cdot k = n$. Thus, $k = n/(t+1)$.

The length of the encoding is $S + \sum_{i\in[k]} t_i = S + t \cdot k$. Note that t and k are random variables that depend on the choice of DB. By Shannon's theorem, we know that

$$\mathop{\mathbf{E}}_{\mathsf{DB} \xleftarrow{\$} \{0,1\}^n}[S + t \cdot k] = \mathop{\mathbf{E}}_{\mathsf{DB} \xleftarrow{\$} \{0,1\}^n}\left[S + t \cdot \frac{n}{t+1}\right] \ge n$$

Observe that for any DB, $\frac{t}{t+1} \le \frac{T}{T+1}$. Thus, we have that

$$S + n \cdot \frac{T}{T+1} \ge n$$

which directly implies that $S(T+1) \ge n$.

2 Space-Time Tradeoff for Preprocessing PIR

We will next prove a space-time tradeoff for preprocessing PIR. Intuitively, the privacy requirement of PIR implies that during a query for some index $i \in [n]$, the probability that the server actually visits position i is rather small. This allows us to rely on the space-time tradeoff for Yao's box problem to prove a PIR lower bound.

Theorem 3 (Space-time tradeoff for preprocessing PIR [CGK20])**.** *Suppose we have a 1-server prepossessing PIR with perfect correctness and* $\mathsf{negl}(n)$ *privacy loss where* $\mathsf{negl}(\cdot)$ *denotes a negligible function. Let S be the client space and let T be the server computation per query. Then,*

$$(S+1)(T+1) \ge \Omega(N)$$

[elaine: TODO: change the statement based on the proof]

This lower bound holds even for computationally private schemes. It holds even for a single query, regardless of bandwidth, number of rounds, and preprocessing cost. However, we require that the server store only the original database during the query phase and no extra information.

To prove Theorem 3, we actually need a version of Yao's box problem that allows probalistic correctness, as stated below.

Yao's box problem with probabilistic correctness. We say that some protocol Π solves Yao's box problem with probabilistic correctness $\delta(n)$, iff for any $\mathsf{DB} \in \{0,1\}^n$, it holds that

$$\Pr\left[i \xleftarrow{\$} [n], \text{run } \Pi \text{ with } \mathsf{DB} \text{ and } i : \text{output is correct}\right] \geq \delta(n)$$

We can extend the proof to Yao's box problem to allow probabilistic correctness, as stated in the following theorem:

Theorem 4 (Yao's box problem with probabilistic correctness)**.** *[elaine: FILL]*

Below, we will first assume that Theorem 4 is true, and prove a lower bound for preprocessing PIR in Section 2.1. Next, in Section 2.2, we show how to modify the earlier proof for Yao's box problem to get the probalistic version (i.e., Theorem 4) that we need.

2.1 Preprocessing PIR $\Longrightarrow$ Yao

Given a preprocessing PIR scheme, we will construct a solution to Yao's box problem (with probablistic correctness) as follows:

- Client's hint: PIR's hint.
- Query for $i \in [n]$: Run PIR for query i. If server is about to look at $\mathsf{DB}[i]$, then stop and simply output "error"; else, output the PIR's reconstructed answer.

Clearly, if the PIR scheme has client space S and server time T, then the resulting protocol (for solving Yao's box problem) also enjoys client space S and server time as most T, where server time is measured in terms of the number of locations visited by the server. Moreover, we claim that the protocol satisfies the following probabilistic correctness:

Claim 5. *Suppose the PIR scheme has perfect correctness and negligible privacy loss. Then, the above protocol solves Yao's box problem with correctness probability* $1 - \frac{T}{n} - \mathsf{negl}(n)$ *for a random query.*

Proof. Henceforth, fix an arbitrary DB. Let $\mathsf{PIRExpt}^j$ denote the random experiment that runs the PIR scheme over database DB and query j. let

$$p_{j,i} := \Pr\left[\text{ server visits } i \text{ in } \mathsf{PIRExpt}^j\right]$$

It suffices to prove the following:

$$\Pr\left[i \xleftarrow{\$} [n],\ \text{server visits } i \text{ in } \mathsf{PIRExpt}^i\right] = \frac{1}{n} \cdot \sum_{i \in [n]} p_{i,i} \leq \frac{T}{n} + \mathsf{negl}(n) \tag{1}$$

We now prove the above inequality. The privacy requirement of PIR implies that for any $j, i \in [n]$, $|p_{j,i} - p_{i,i}| \leq \mathsf{negl}'(n)$. Since the server can perform at most T amount of work, we have that for any $j \in [n]$, $\sum_{i \in [n]} p_{j,i} \leq T$. Combining the above facts, we have that

$$T \geq \sum_{i \in [n]} p_{j,i} \geq \sum_{i \in [n]} (p_{i,i} - \mathsf{negl}'(n))$$

which gives us Equation (1). □

Combining the above reduction and Theorem 4, we immediately get the claimed space-time tradeoff for 1-server preprocessing PIR.

2.2 Proof of Theorem 4

We now prove Theorem 4. We present a variant of the proof[1] described by De, Trevisan, and Tulsiani [DTT10].

Fact 1 (Extension of Shannon's theorem for codes with probabilistic correctness.)**.** *Suppose there is a randomized encoding procedure* $\mathsf{Enc} : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ *and decoding procedure* $\mathsf{Dec} : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^n$ *such that*

$$\Pr\left[r \xleftarrow{\$} \{0,1\}^r, \mathsf{DB} \xleftarrow{\$} \{0,1\}^n : \mathsf{Dec}\left(\mathsf{Enc}(\mathsf{DB}, r), r\right) = \mathsf{DB}\right] \geq \delta$$

Then, $m \geq n - \log(\frac{1}{\delta})$.

Proof. There must exist some r such that $\Pr\left[\mathsf{DB} \xleftarrow{\$} \{0,1\}^n : \mathsf{Dec}\left(\mathsf{Enc}(\mathsf{DB}, r), r\right) = \mathsf{DB}\right] \geq \delta$. In other words, there is an r such that at least δ fraction of the databases can decrypt correctly. This means that $\mathsf{Enc}(\mathsf{DB}, r)$ must attain at least $\delta \cdot 2^n$ values as DB varies over $\{0,1\}^n$. As the total number of values $\mathsf{Enc}(\cdot, r)$ can attain is at most 2^m, we have that $2^m \geq \delta \cdot 2^n$ which gives us the desired inequality. □

Fact 2. *Let* $\epsilon \in (0, 1/2)$*. Given a set of* M *items, pick a subset of size at most* $(1/2 - \epsilon)M$*. The subset can be encoded using* $M \cdot (1 - 2.87\epsilon^2)$ *bits.*

Proof. The number of ways to choose a subset of size exactly $(1/2 - \epsilon)M$ among M elements is

$$\begin{aligned} f = &\binom{M}{(\frac{1}{2} - \epsilon)M} = \frac{M!}{(\frac{1}{2} - \epsilon)M)! \cdot ((\frac{1}{2} + \epsilon)M)!} \\ \leq &\frac{\left(\frac{M}{e}\right)^M}{\left(\frac{(\frac{1}{2}-\epsilon)M}{e}\right)^{(\frac{1}{2}-\epsilon)M} \cdot \left(\frac{(\frac{1}{2}+\epsilon)M}{e}\right)^{(\frac{1}{2}+\epsilon)M}} \qquad \text{(By Stirling)} \\ = &\left(\frac{1}{\frac{1}{4} - \epsilon^2}\right)^{(\frac{1}{2}-\epsilon)M} \cdot \left(\frac{1}{\frac{1}{2} + \epsilon}\right)^{2\epsilon M} \end{aligned}$$

Taking $\log_2$ of the above expression, and then using Taylor expansion, we get that for sufficiently large M, $\log_2(f) \leq \log_2\left(\left(\frac{1}{\frac{1}{4}-\epsilon^2}\right)^{(\frac{1}{2}-\epsilon)M} \cdot \left(\frac{1}{\frac{1}{2}+\epsilon}\right)^{2\epsilon M}\right) \leq M \cdot (1 - 2.87\epsilon^2)$. □

Lemma 6. *Consider a* **deterministic** *protocol* Π *for solving Yao's box problem with client space* S *and server time* T*, and suppose* $T \in [2, o(n)]$*. Let* $\mathsf{DB} \in \{0,1\}^n$ *be some fixed string such that* Π *gives a correct answer on at least* $\frac{3}{4}$ *[elaine: note param] fraction of the indices. [elaine: may need some assumption T much smallller than n] Then,* $\mathsf{Enc} : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ *and decoding procedure* $\mathsf{Dec} : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^n$ *such that*

$$\Pr\left[r \xleftarrow{\$} \{0,1\}^r : \mathsf{Dec}\left(\mathsf{Enc}(\mathsf{DB}, r), r\right)\right] \geq \Omega(1)$$

[elaine: FILL some concrete number] Further, the encoding length $m \leq n - \Theta(\frac{n}{10T}) + S$.

[1] The proof we present below fixes a bug in their proof. Specifically, the way they take Chernoff bound in the proof of Lemma 8.4 is incorrect since the underlying random variables are positively correlated. Thanks to Ashrujit Ghoshal for helping me fix the bug.

Proof. Suppose we sample each index in $[n]$ with probability $1/10T$ into some random set R. For some $i \in R$, if running the protocol Π on i does not end up visiting any position in R, then we say that i is *good*. Let $G \subseteq R$ be the subset of R that are good. We partition $G = G_0 \cup G_1$ into two disjoint subsets, where G_0 includes every index $i \in G$ such that running Π on i gives a correct answer, and G_1 includes every index $i \in G$ such that running Π on i gives an incorrect answer.

Fact 3. $\mathbf{E}[|R|] = n/10T$. *Further, by Chernoff bound,* $\Pr[|R| \leq 0.09n/T] \geq \exp(-\Omega(n/T))$.

Claim 7. $\Pr[|G| > 0.042n/T] \geq 0.65$ *for sufficiently large* n.

Proof. For a fixed $i \in [n]$, the probability that it lands in R and is bad is

$$\frac{1}{10T} \cdot \left(1 - \left(1 - \frac{1}{10T}\right)^T\right) \leq \frac{1}{10T} \cdot \left(1 - \frac{1}{(2e)^{0.1}}\right) \leq 0.016/T$$

Therefore, the expected number of indices that land in R and are bad is at most $0.016n/T$. By Markov Inequality, with probability at most $1/3$, the number of indices that land in R and are bad exceeds $0.048n/T$.

Due to Fact 3, and taking a union bound, with probability at least $1 - \exp(-\Omega(n/T)) - 1/3$, $|G| \geq 0.09n/T - 0.048n/T = 0.042n/T$. □

Claim 8. *With probability at least* 0.37, $|G_0| - |G_1| \geq 0.01 \cdot n/T$.

Proof. For a fixed $i \in [n]$, the probability that i lands in R and is good is

$$\frac{1}{10T} \cdot \left(1 - \frac{1}{10T}\right)^T \in [0.09/T, 0.091/T]$$

Therefore, we have that

$$\mathbf{E}[|G_0| - |G_1|] \geq 0.09/T \cdot \frac{3n}{4} - 0.091/T \cdot \frac{n}{4} \geq 0.044n/T$$

Therefore, it must be that with probability $p \geq 0.37$, $|G_0| - |G_1| \geq 0.01 \cdot n/T$, since otherwise,

$$\mathbf{E}[|G_0| - |G_1|] \leq (p - \exp(-\Omega(n/T))) \cdot 0.09n/T + (1-p) \cdot 0.01n/T + \exp(-\Omega(n/T)) \cdot n < 0.044n/T$$

□

Combining Claim 7 and Claim 8 and taking union bound, we know that with probability at least $1 - (1 - 0.65) - (1 - 0.37) = 0.02$ over the choice of R, it holds that $|G_0| - |G_1| \geq 0.01n/T$ and moreover $|G| \geq 0.042n/T$.

Encoding.

Decoding. □

Old TEXT below

For the proof, we will show that a solution to the PIR problem can be used to construct an algorithm to solve a probabilistic version of Yao's Box Problem.

Probabilistic Yao's Box Problem Suppose we have a working PIR scheme. Now, we will construct a solution to probabilistic Yao's Box Problem as follows:

- Client's Hint: PIR's hint
- Query for $i \in [N]$: Run PIR for query i. If server looks at $\mathsf{DB}[i]$, then output "error".

We want to show the following. Given that PIRExpt: $i \xleftarrow{\$} [N]$, PIR preprocessing, PIR query on i,

$$p = \mathbb{P}[\text{PIRExpt opens } i] \leq \frac{T}{N} + \mathsf{negl}(N)$$

That is, we want to show that if we run a PIR query with the a random index i, the probability we open that index is small.

For a fixed i, define the following probability $p_i = \mathbb{P}[\text{PIR on } i \text{ looks at } i]$. Then,

$$p = \frac{1}{N} \sum_i p^i$$

Assume for the sake of contradiction that $p > \frac{T}{N} + \mu$, where μ is non-negligible.

Let $p_{ji} = \mathbb{P}[\text{PIR on } j \text{ opens } i]$. Since our scheme is private, the different indices should be computationally indistinguishable. Thus, this probability should be equally distributed. As a result,

$$p_{ji} = \mathbb{P}[\text{PIR on } j \text{ opens } i] \geq p_i - \mathsf{negl}(N)$$

$$\begin{aligned}
E[\text{server work for PIR on } j] &\geq \sum_{i=1}^{N} p_{ji} \\
&\geq \sum_{i=1}^{N} (p_i - \mathsf{negl}(N)) \\
&= Np - \mathsf{negl}(N) && \text{(By def. of } p\text{)} \\
&> T + \mu N - \mathsf{negl}(N) && \text{(As } p > \tfrac{T}{N} + \mu\text{)}
\end{aligned}$$

Thus, we have a contradiction because the expected number of locations the server needs to look at is strictly greater than T. Thus, we have shown that

$$\mathbb{P}[\text{PIRExpt opens } i] \leq \frac{T}{N} + \mathsf{negl}(N)$$

Shifting our focus back to Yao's Box problem with probabilistic correctness on random index, the probabilistic correctness is

$$\mathbb{P}[i \xleftarrow{\$} \text{correct for } i] \geq 1 - \frac{T}{N} - \mathsf{negl}(N)$$

Encoding Argument Randomness comes from two parts: the preprocessing part (the client's hint), and the query part. With this in mind, we will be using an augmented version of the encoding type argument in **??**.

Consider the following experiment.

1. Run preprocessing, and choose a "reasonably good hint." Initially, let the encoding be just the hint. We will add to this encoding as we go forward.
2. Define an empty set called $\mathsf{known} = \{\}$
3. In each step i, find the smallest $q_i \notin \mathsf{known}$.
   - If $\exists$ online coins such that query q_i will give the correct answer, choose the lexicographically smallest coin. Execute query.

     $$\mathsf{known} \Longleftarrow \cup\{q_i\} \cup \{\text{all newly opened}\}$$

     Add "newly opened" to encoding.
   - Else add q_i-th bit to the encoding.
4. Repeat until $\mathsf{known} = [N]$.

A hint is bad for $i \in [N]$ if $\mathbb{P}[\text{query } i \text{ correct}|\text{hint}] = 0$. That is, there does not exist an online coin such that the query is correct.

Claim 9. *$\exists$ hint that's bad for at most $T+1$ location.*

Proof. If all hints are bad for more than $T+1$ locations, then

$$\mathbb{P}[i \in [N], \text{correct on } i] < 1 - \frac{T+1}{N}$$

We have a contradiction, because it disagrees with our probabilistic correctness result above. □

Finally, we can reason about the encoding length.

Suppose the worst case where the hint is bad for exactly $T+1$ locations. Let $\mathsf{b}_{\mathsf{bad}}$ be the encoding of the "newly opened" boxes in the bad queries, and $\mathsf{b}_{\mathsf{good}}$ be the encoding of the "newly opened boxes in the good queries.

- $|\mathsf{b}_{\mathsf{bad}}| = T+1$, as each bad iteration adds one bit, and we have $T+1$ iterations.
- To find $|\mathsf{b}_{\mathsf{good}}|$, we repeat the argument from **??**. Let $t_1, t_2, \ldots t_k$ be the number of newly opened boxes at each good step $i \in [k]$.

  By the "plus one" argument, we have that $\sum_{i=1}^{k}(t_i+1) \geq N - (T+1)$. We subtract $T+1$ here because those indices have been handled by the bad iterations.

  Let $t = \frac{\sum_{i=1}^{k} t_i}{k}$. Then as before, we have $k \geq \frac{N}{t+1}$

  $$\begin{aligned} \sum_{i=1}^{k}(t_i+1) &\geq N - (T+1) \\ &\Longrightarrow tk + k \geq N - T - 1 \\ &\Longrightarrow k \geq \frac{N-T-1}{t+1} \end{aligned}$$

  Thus, $|\mathsf{b}_{\mathsf{good}}| = tk \geq t\frac{N-T-1}{t+1}$

With the information above, we can mek the following conclusion.

$$
\begin{aligned}
|\mathsf{enc}| = S + |\mathsf{b}_{\mathsf{good}}| + |\mathsf{b}_{\mathsf{bad}}| &\geq N \\
\implies S + t\frac{N-T-1}{t+1} + T + 1 &\geq N \\
\implies S(t+1) + T + 1 &\geq N && \text{(By simplification)} \\
\implies (S+1)(T+1) &\geq N && \text{(As } t \text{ upper bounded by } T\text{)}
\end{aligned}
$$

□

References

[CGK20] Henry Corrigan-Gibbs and Dmitry Kogan. Private information retrieval with sublinear online time. In *Advances in Cryptology – EUROCRYPT 2020*, pages 44–75, Cham, 2020. Springer International Publishing.

[DTT10] Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and prgs. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 649–665. Springer, 2010.

[Yao90] A. C.-C. Yao. Coherent functions and program checkers. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 84–94, New York, NY, USA, 1990. Association for Computing Machinery.

[ZPSZ23] Mingxun Zhou, Andrew Park, Elaine Shi, and Wenting Zheng. Piano: Extremely simple, single-server pir with sublinear server computation. Cryptology ePrint Archive, Paper 2023/452, 2023. `https://eprint.iacr.org/2023/452`.