Точное значение коммуникационной сложности для одного класса PIR-протоколов

Г. А. Майлыбаева

Протоколы извлечения информации без раскрытия запроса (PIR-протоколы) позволяют пользователю получить желаемый бит информации из базы данных, копия которой хранится на нескольких несообщающихся серверах таким образом, что администраторы базы данных ничего не узнают о номере бита который запрашивал пользователь. Коммуникационная сложность протокола определяется как суммарное количество бит, которыми обмениваются пользователь и сервера во время протокола. В работе найдено точное значение коммуникационной сложности для одного класса PIR-протоколов.

1. Введение

Рассмотрим протокол с $k+1$ участником: пользователем и k несообщающимися серверами ($k \geqslant 1$), причем каждый из серверов хранит один и тот же булев вектор $x = (x_0, \ldots, x_{n-1})$ длины n — базу данных. Пользователь желает узнать значение i-го бита x_i этого вектора так, чтобы номер бита i не стал известен ни одному из серверов. Протокол, который позволяет это делать, называется протоколом доступа к данным без раскрытия запроса и состоит из следующих шагов.

1) Пользователь имеет номер бита i и вырабатывает случайное число r. По числам i и r пользователь вычисляет с помощью специальной функции, называемой функцией запросов, k чисел q^j и посылает j-му серверу запрос q^j.

2) Каждый из k серверов по полученному запросу q^j и базе данных x с помощью специальной функции ответов вычисляет вектор a^j и посылает его пользователю.
3) Пользователь по числам i, r и k ответам серверов a^j вычисляет с помощью реконструирующей функции нужный бит x_i.

Первое требование к протоколу состоит в том, что ни один из серверов по своему запросу q^j не может понять, с помощью какого бита i этот запрос был порожден. Это требование называется требованием защищенности. Второе требование к протоколу, называемое требованием корректности, заключается в том, что пользователь по ответам серверов правильно восстанавливает бит x_i. Предполагается, что всем участникам протокола — и пользователю и серверам — известны функции запросов, ответов и реконструирующая. Но серверам не известно случайное число r и, разумеется, не известен номер бита i.

Понятие такого протокола впервые было введено в [1, B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, 1995] под названием Private Information Retrieval, поэтому мы в дальнейшем будем называть такие протоколы PIR-протоколами.

Суммарная длина в битах запросов к серверам и ответов серверов называется коммуникационной сложностью PIR-протокола.

На настоящий момент известны следующие результаты. В [8, Sergey Yekhanin, 2006] был получен PIR-протокол для $k = 3$ серверов с коммуникационной сложностью $O(n^{10^{-7}})$. На данный момент, коммуникационная сложность этого протокола является наилучшей для $k = 3$.

В [2, A. Beimel, Y. Ishai, E. Kushilevitz, J.-F. Raymond, 2002] для каждого натурального $k \geqslant 2$ получен PIR-протокол с коммуникационной сложностью $n^{O(\lg\lg k/k \lg k)}$. Для $k = 2$ сложность построенного PIR-протокола равна $O(n^{1/3})$. На данный момент, коммуникационная сложность этого протокола является наилучшей для $k = 2$.

В [3, O. Goldreich, H. Karloff, L. Schulman, L. Trevisan, 2002] было показано, что если функции ответов всех серверов линейные, и длина в битах ответа любого сервера равна a, то длина запроса должна быть не меньше $O(\frac{n}{2^a})$. Также было показано, что сложность любого

PIR-протокола для 2-х серверов, функции ответов которого являются линейными, и пользователь использует ровно a бит из ответа каждого сервера, не превышает $O(n^{1/a+1})$. Для доказательства нижних оценок авторы преобразовывают произвольный PIR-протокол в LDC (Locally Decodable Code) протокол и приводят доказательство оценки для LDC-протокола.

В [4, B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, 1998] была получена следующая нижняя оценка. Если функция запросов линейна и ответ любого сервера это ровно один бит, тогда длина запроса к каждому серверу должна быть не меньше $n-1$ бит. Эта оценка является точной. В той же работе построен PIR-протокол с такой же коммуникационной сложностью.

В [5, I. Kerendis и R. de Wolf, 2003] было показано, что если пользователь использует из ответа каждого сервера не более a бит, то длина запроса должна быть не меньше $O(\frac{n}{2^6 a})$. Для доказательства нижней оценки авторы преобразовывают произвольный PIR-протокол в квантовый PIR-протокол и приводят доказательство оценки для квантового PIR-протокола.

В [6, R. Beigel, L. Fortnow, W. Gasarch, 2003] было показано, что если каждый сервер посылает в ответ пользователю ровно один бит, то длина запроса должна быть не менее $n-2$. При этом существует PIR-протокол ([4]), где пользователь посылает каждому серверу $n-1$ бит и получает в ответ ровно 1 бит от каждого сервера.

Для того, чтобы сформулировать следующий результат, дадим некоторые определения.

Определение 1. Пусть k, n, s, m, a — произвольные натуральные числа. Пусть $I = \langle Q, A^0, A^1, \ldots, A^{k-1}, R\rangle$ — PIR-протокол для k серверов, для базы данных длины n, с датчиком случайных чисел длины s, с длиной запроса m и длиной ответа a, тогда

1) PIR-протокол I называется линейным, если для любого $0 \leqslant j \leqslant k-1$ функция $A^j : E_2^m \to E_2^a$ линейна по каждой переменной. То есть для произвольных $b_1, \ldots, b_{p-1}, b, c, b_{p+1}, \ldots, b_m \in \{0,1\}$, верно

$$A^j(b_1, \ldots, b_{p-1}, b+c, b_{p+1}, \ldots, b_m) =$$

$$= A^j(b_1, \ldots, b_{p-1}, b, b_{p+1}, \ldots, b_m) + \\ + A^j(b_1, \ldots, b_{p-1}, c, b_{p+1}, \ldots, b_m).$$

2) Пусть l — произвольное натуральное число. PIR-протокол I называется l-мультилинейным, если для любого $0 \leqslant j \leqslant k-1$ функция $A^j : (E_2^l)^{m/l} \to E_2^a$ линейна по каждой переменной. То есть для произвольных $b_1, \ldots, b_{p-1}, b, c, b_{p+1}, \ldots, b_{m/l} \in \{0,1\}^l$, верно

$$A^j(b_1, \ldots, b_{p-1}, b + c, b_{p+1}, \ldots, b_{m/l}) = \\ = A^j(b_1, \ldots, b_{p-1}, b, b_{p+1}, \ldots, b_{m/l}) + \\ + A^j(b_1, \ldots, b_{p-1}, c, b_{p+1}, \ldots, b_{m/l}).$$

3) Пусть l — произвольное натуральное число. PIR-протокол I называется l-аффинным (l-*affine*), если для любого $0 \leqslant j \leqslant k-1$ функция $A^j : (E_2^l)^{m/l} \to E_2^a$ линейна с параметром $0^l = (0, \ldots, 0) \in \{0,1\}^l$ по каждой переменной. То есть для произвольных $b_1, \ldots, b_{p-1}, b, c, b_{p+1}, \ldots, b_{m/l} \in \{0,1\}^l$, верно

$$A^j(b_1, \ldots, b_{p-1}, b + c, b_{p+1}, \ldots, b_{m/l}) = \\ = A^j(b_1, \ldots, b_{p-1}, b, b_{p+1}, \ldots, b_{m/l}) + \\ + A^j(b_1, \ldots, b_{p-1}, c, b_{p+1}, \ldots, b_{m/l}) + \\ + A^j(b_1, \ldots, b_{p-1}, 0^l, b_{p+1}, \ldots, b_{m/l}).$$

В [7, T. Itoh, 2001] было показано, что любой линейный PIR-протокол для k серверов имеет сложность не менее $\sqrt{\frac{n}{2k}}$. Пусть k, l — произвольные натуральные числа и ε — произвольное положительное число. Тогда если I произвольный l-мультилинейный или l-аффинный PIR-протокол для k серверов, то почти для всех n верно: $C(I) \geqslant (1/(k-1)^{1/(l+1)} - 1)n^{1/(l+1)}$.

В [9, Alexander Razborov, Sergey Yekhanin, 2006] была получена нижняя оценка для билинейных *group-based* PIR-протоколов с 2-мя серверами. Реконструирующая функция билинейных PIR-протоколов является суммой бит ответов серверов, *group-based* означает, что сервера представляют базу данных в виде функции, действующей в

конечном поле, и позволяют пользователю запросить значение этой функции на некотором элементе группы, используя схему разделения секрета в этой группе. А именно, показано, что для данного класса PIR-протоколов $C(I) \geqslant \Omega(n^{1/3})$.

В данной работе был получен результат для более широкого класса PIR-протоколов. А именно, во-первых, в отличие от рассмотренных выше результатов, мы не предполагали, что длины ответов серверов равны между собой, во-вторых, мы не налагали никаких ограничений на количество бит, которые пользователь использует из ответов серверов. В-третьих, получена нижняя оценка, которая не налагает ограничений на линейность функции ответов. Также заметим, что для доказательства нижней оценки, авторы [3, O. Goldreich, H. Karloff, L. Schulman, L. Trevisan, 2002] используют сведение PIR-протоколов к LDC-протоколам, а авторы [5, I. Kerendis и R. de Wolf, 2003] используют сведение PIR-протоколов к квантовым PIR-протоколам. В данной статье мы доказываем нижнюю оценку напрямую для PIR-протоколов.

Ответ каждого сервера это вектор, каждый бит которого является некоторой функцией от бит базы данных. Степенью существенности функции ответов назовем максимальную степень существенности всех этих функций.

Тогда в классе PIR-протоколов с 2-мя серверами, степень существенности функции ответов которых не превышает 2, и длина датчика случайных чисел меньше длины базы данных, построен PIR-протокол и получена нижняя оценка коммуникационной сложности, которая по порядку совпадает с коммуникационной сложностью построенного PIR-протокола для 2-х серверов.

Автор выражает благодарность Э. Э. Гасанову за постановку задачи и помощь в работе.

2. Основные понятия

Строго формальное определение PIR-протокола было введено ранее в [10, 11]. Поскольку в работе рассматривается только случай с 2-мя серверами, то здесь мы приведем несколько упрощенную версию понятия PIR-протокола.

Для любого натурального n обозначим $E_n = \{0, \ldots, n-1\}$.

Пусть n, s, p^0, p^1 — натуральные числа, $p = p^0 + p^1$. Пусть на множестве $B = \{(i, r),\ i \in E_n,\ r \in E_s\}$ задано вероятностное пространство $<B, 2^B, \mathbf{P}>$, где $\mathbf{P}(i, r) = \frac{1}{n \cdot s}$, для любых $i \in E_n$, $r \in E_s$. Тогда $(2, n, s, p)$ *PIR-протоколом* называется набор из 4 функций $I = \langle Q, A^0, A^1, R\rangle$, где Q, A^0, A^1, R некоторые отображения, $Q : E_2 \times E_n \times E_s \to E_s$, $A^j : E_s \times \{0,1\}^n \to \{0,1\}^{p^j}$, $j \in E_2$, $R : E_n \times E_s \times \{0,1\}^p \to \{0,1\}$, такие, что выполнено 2 условия:

корректности: для любых $i \in E_n$, $r \in E_s$ выполнено

$$R(i, r, A^0(Q(0, i, r), x), A^1(Q(1, i, r), x)) = x_i;$$

защищенности: для любых $q \in E_s$, $t \in E_2$, $i, j \in E_n$ выполнено

$$\mathbf{P}(Q(t, i, r) = q) = \mathbf{P}(Q(t, j, r) = q).$$

Здесь и везде далее в статье n — длина базы данных $x = (x_0, x_1, \ldots, x_{n-1})$; s — параметр датчика случайных чисел, точнее, датчик случайных чисел дает равновероятно числа из множества E_s; p^j — количество бит в ответе j-го сервера, A^j — функция ответа j-го сервера ($j \in E_2$); R — реконструирующая функция.

Содержательно протокол $I = \langle Q, A^0, A^1, R\rangle$ состоит из следующих шагов:

- Пользователь U, имея запрос i, вырабатывает случайное число $r \in E_s$, для каждого $j \in E_2$ вычисляет $q^j = Q(j, i, r)$ и посылает q^j серверу S_j.
- Каждый сервер S_j, $j \in E_2$, вычисляет $a^j = (a^j_0, \ldots, a^j_{p^j-1}) = A^j(q^j, x)$ и посылает вектор a^j пользователю.
- U вычисляет $x_i = R(i, r, a^0, a^1)$.

Если d — вещественное число, то через $]d[$ обозначим наименьшее целое не меньшее, чем d, а через $[d]$ — наибольшее целое не большее, чем d.

Величина $C(I) = 2\,]\log_2 s[+ p$ называется *коммуникационной сложностью протокола* I, и представляет собой число бит, переданных в процессе протокола.

Условие корректности гарантирует, что пользователь получит нужный бит базы данных, а условие защищенности — что ни один из серверов по числу q, которое он получил, не сможет понять, какой бит интересует пользователя.

Основной целью исследований в этой области является построение для заданной длины базы данных n и максимального значения датчика случайных чисел s PIR-протокола с минимальной коммуникационной сложностью.

Степенью существенности булевой функции $f(x_1, \ldots, x_l)$ назовем число переменных, от которых она существенно зависит, и обозначим его через $S(f)$.

Степенью существенности булевой вектор-функции

$$F(x_1, \ldots, x_l) = (f_1(x_1, \ldots, x_l), \ldots, f_t(x_1, \ldots, x_l))$$

назовем число

$$S(F) = \max_{1 \leqslant j \leqslant t} S(f_j).$$

Пусть $A^j(q, x) = (A^j_0(q, x), \ldots, A^j_{p^j-1}(q, x))$. Для функций $A^j(q, x)$, $A^j_l(q, x), l \in E_{p^j}, j \in E_2, q \in E_s$ также будем использовать следующую запись: $A^j(q, x) = A^j(q)(x)$, $A^j_l(q, x) = A^j_l(q)(x)$, $l \in E_{p^j}$, где $A^j(q) : \{0,1\}^n \to \{0,1\}^{p^j}$ — булева вектор-функция n переменных, $A^j_l(q) : \{0,1\}^n \to \{0,1\}$ — булева функция n переменных.

Степенью существенности функции ответов j-го сервера $A^j : E_s \times \{0,1\}^n \to \{0,1\}^{p^j}$, $j \in E_2$, назовем число

$$S(A^j) = \max_{q \in E_s} S(A^j(q)).$$

Обозначим через $\mathcal{I}(2, n, s)$ класс всех $(2, n, s, p)$ PIR-протоколов, где $p > 0$. Пусть $\mathcal{A}$ — некоторое множество PIR-протоколов. Тогда обозначим

$$C(2, n, s, \mathcal{A}) = \min\{C(I) : I \in \mathcal{A} \cap \mathcal{I}(2, n, s)\}.$$

Обозначим через $\mathcal{A}_2$ множество всех PIR-протоколов для 2-х серверов таких, что степень существенности функции ответов каждого сервера не превосходит 2.

Теорема 1 (Верхняя оценка). *Для любых натуральных $n \geqslant s$ верно*

$$C(2,n,s,\mathcal{A}_2) \leqslant 2\,]\log_2 s[\,+\frac{s+1}{2s}n+(n \mod s^2).$$

Пусть функция $\mathrm{sign}:\mathbb{N}\cup\{0\}\to\{0,1\}$ такая что

$$\mathrm{sign}(n)=\begin{cases}0, & \text{если } n=0,\\ 1, & \text{если } n\geqslant 1.\end{cases}$$

Теорема 2 (Нижняя оценка). *Для любых натуральных n, s верно*

$$C(2,n,s,\mathcal{A}_2) \geqslant 2\,]\log_2 s[\,+\frac{s+1}{2s}n-\frac{s+1}{2}\left(\mathrm{sign}(d)-\frac{d}{s}\right),$$

где $d = n \mod s$.

Следствие 1. *Для любых натуральных n, s таких что n кратно $2s^2$, верно*

$$C(2,n,s,\mathcal{A}_2) = 2\,]\log_2 s[\,+\frac{s+1}{2s}n.$$

3. Верхняя оценка

Лемма 1. *Для любого $(2,n,s,p)$ PIR-протокола $I=\langle Q,A^0,A^1,R\rangle$, для любых $i\in E_n, j\in E_2$ не существует таких $r_1,r_2\in E_s, r_1\neq r_2$, что верно $Q(j,i,r_1)=Q(j,i,r_2)$.*

Доказательство. Пусть $I=\langle Q,A^0,A^1,R\rangle$ — произвольный $(2,n,s,p)$ PIR-протокол. Для произвольных $j\in E_2, i\in E_n, r\in E_s$ рассмотрим функцию $Q_i^j(r)=Q(j,i,r)$. По свойству защищенности $Q_i^j(r)$ должна принимать все s значений. Допустим, существуют такие $r_1,r_2\in E_s, r_1\neq r_2$ что $Q_i^j(r_1)=Q_i^j(r_2)$. Но тогда функция $Q_i^j(r)$ принимает одно значение в двух точках. Это означает, что существует такой запрос $q\in E_s$, что для любого $r\in E_s$ верно $Q_i^j(r)=Q(j,i,r)\neq q$, требование защищенности нарушается. Получаем противоречие. Тем самым лемма 1 доказана.

Лемма 2. *Для любого* $(2, n, s, p)$ *PIR-протокола* $I = \langle Q, A^0, A^1, R\rangle$ *существует* $(2, n, s, p)$ *PIR-протокол* $I' = \langle Q', A^0, A^1, R\rangle$ *такой что*

$$Q'(j, i, r) = \begin{cases} r, & \text{если } j = 0, \\ Q(1, i, p_i(r)), & \text{если } j = 1, \end{cases}$$

где $p_i(r) : E_s \to E_s, i \in E_n$ *— некоторые взаимнооднозначные отображения.*

Доказательство. Сопоставим функции запросов $Q(j, i, r)$ матрицу B^Q, элементами которой являются пары: $B^Q_{i,r} = (Q(0, i, r), Q(1, i, r))$, где $i \in E_n, r \in E_s$. Рассмотрим i-ую строку этой матрицы. По свойству защищенности для любого $i \in E_n$ функция $Q(0, i, r)$ принимает все s значений. Тогда существует такая перестановка $p_i(r)$ столбцов матрицы, что $Q(0, i, p_i(r)) = r$. Применим к элементам i-ой строки перестановку, обратную к $p_i(r)$. Выполним эту операцию для всех строк матрицы B^Q. Обозначим результирующую матрицу через B'^Q. По новой матрице строим функцию запросов: $(Q'(0, i, r), Q'(1, i, r)) = B'^Q_{i,r}, i \in E_n, r \in E_s$. Рассмотрим PIR-протокол $\langle Q', A^0, A^1, R\rangle$. Очевидно, свойства корректности и защищенности в этом протоколе сохраняются. Получили искомую функцию $Q'(j, i, r)$. Тем самым лемма доказана.

Доказательство теоремы 1.

Опишем PIR-протокол $I_{n,s} = \langle Q, A^0, A^1, R\rangle \in \mathcal{A}_2$ с датчиком случайных чисел, принимающим s значений.

Обозначим

$$l'_0 = [n/s^2], l' = sl'_0, l'' = n - sl' = n - s^2 l'_0.$$

Протокол состоит из трех шагов.

Шаг 1.

По номеру бита i и случайному числу $r \in E_s$ пользователь вычисляет $q^j, j \in E_2$, по правилу:

$$q^0 = Q(0, i, r) = r,$$

$$q^1 = Q(1, i, r) = \begin{cases} r, & \text{если } i < l', \\ r + 1 \mod s, & \text{если } l' \leqslant i < 2l', \\ \vdots & \\ r + s - 1 \mod s, & \text{если } (s-1)l' \leqslant i. \end{cases}$$

и посылает j-му серверу запрос q^j.

Шаг 2.

Каждый бит ответа первого сервера на любой запрос пользователя представляет собой сумму по модулю 2 ровно двух бит базы данных.

Второй сервер на любой запрос пользователя всегда посылает в качестве ответа некоторую проекцию базы данных.

Далее будем полагать, что $M(n,m)$ — множество $n \times m$ матриц с элементами из $\{0,1\}$. Если

$$B = \begin{pmatrix} a_{0,0} & \dots & a_{0,m-1} \\ \vdots & \ddots & \vdots \\ a_{n-1,0} & \dots & a_{n-1,m-1} \end{pmatrix} \in M(n,m),$$

тогда положим

$$B^T = \begin{pmatrix} a_{0,0} & \dots & a_{n-1,0} \\ \vdots & \ddots & \vdots \\ a_{0,m-1} & \dots & a_{n-1,m-1} \end{pmatrix} \in M(m,n).$$

Обозначим $z = z(s) = \frac{s(s-1)}{2}$.

Для вычисления ответа сервер $S_j, j \in E_2$ использует матрицы $B_0^j, \dots, B_{s-1}^j$, которые будут описаны ниже, причем $B_0^0, \dots, B_{s-1}^0 \in M(zl_0', n)$, а $B_0^1, \dots, B_{s-1}^1 \in M(sl_0'+l'', n)$. Полагаем, что пользователь знает вид всех этих матриц.

Сервер S_j вычисляет ответ a^j используя формулу $a^j = A^j(q^j, x) = (B_{q^j}^j x^T)^T$ и посылает результат пользователю.

Пусть V_0 — множество векторов вида $(t_1^0, t_2^0, \dots, t_1^{z-1}, t_2^{z-1})$, где $t_1^l, t_2^l \in E_{s^2}, l \in E_z$ и выполнены следующие условия:

а) для любого $l \in E_z$ верно:

$$t_1^l < t_2^l \text{ и } [t_1^l/s] \neq [t_2^l/s],$$

б) для всех $l_1, l_2 \in E_z, l_1 \neq l_2$ верно:

$$[t_1^{l_1}/s] \neq [t_1^{l_2}/s] \text{ или } [t_2^{l_1}/s] \neq [t_2^{l_2}/s].$$

Тогда множество пар

$$\{[t_1^0/s],[t_2^0/s]\},\{[t_1^1/s],[t_2^1/s]\},\ldots,\{[t_1^{z-1}/s],[t_2^{z-1}/s]\}$$

является множеством всех двухэлементных подмножеств E_s.

Пусть E — единичная матрица из $M(l'_0,l'_0)$, O — нулевая матрица из $M(l'_0,l'_0)$, то есть

$$E=\begin{pmatrix}1&0&\ldots&0\\0&1&\ldots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\ldots&1\end{pmatrix},\quad O=\begin{pmatrix}0&0&\ldots&0\\0&0&\ldots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\ldots&0\end{pmatrix}.$$

Каждому вектору $v=(t_1^0,t_2^0,\ldots,t_1^{z-1},t_2^{z-1})\in V_0$ поставим в соответствие матрицу $T_v^0\in M(zl'_0,s^2l'_0)$, вида:

$$T_v^0=\begin{pmatrix}T_{0,0}&\ldots&T_{0,s^2-1}\\\vdots&\ddots&\vdots\\T_{z-1,0}&\ldots&T_{z-1,s^2-1}\end{pmatrix},$$

где $T_{l,t}\in M(l'_0,l'_0), l\in E_z, t\in E_{s^2}$, причем

$$T_{l,t_1^l}=T_{l,t_2^l}=E \text{ и } T_{l,t}=O \text{ для любых } t\neq t_1^l,t_2^l, l\in E_z.$$

Для матрицы T_v^0 будем использовать более короткое выражение: $T_v^0=(T_{l,t})$. Обозначим $\mathbf{T_0}=\{T_v^0: v\in V_0\}$.

Пусть V_1 — множество векторов вида $(t^0,\ldots,t^{s-1})$, где $t^l\in E_{s^2}, l\in E_s$ и выполнено:

$$[t^l/s]=l,$$

для любого $l\in E_s$.

Каждому вектору $v=(t^0,\ldots,t^{s-1})\in V_1$ поставим в соответствие матрицу $T_v^1\in M(sl'_0,s^2l'_0)$ следующим образом: $T_v^1=(T_{l,t})$, где $T_{l,t}\in M(l'_0,l'_0); l\in E_s, t\in E_{s^2}$, причем

$$T_{l,t^l}=E \text{ и } T_{l,t}=O, \text{ для любых } t\neq t^l;\ t,t^l\in E_{s^2},\forall l\in E_s.$$

Обозначим: $\mathbf{T_1}=\{T_v^1: v\in V_1\}$.

Пусть вектора $v_j(q), j \in E_2, q \in E_s$ определяются следующим образом. Для каждого $q \in E_s$ положим: $v_1(q) = (t_q^0, \ldots, t_q^{s-1}) \in V_1$, где $t_q^l = sl + q, \forall l \in E_s$. Заметим, что $t_q^l \mod s = q$ и $[t_q^l/s] = l$ для любого $l \in E_s$. $v_0(q) = (t_{q,1}^0, t_{q,2}^0, \ldots, t_{q,1}^{z-1}, t_{q,2}^{z-1}) \in V_0$.

Пусть

$$([t_{q,1}^l/s], [t_{q,2}^l/s]) = (c_{q,1}^l, c_{q,2}^l), \forall l \in E_z, q \in E_s$$

для некоторых $c_{q,1}^l, c_{q,2}^l \in E_s, c_{q,1}^l \neq c_{q,2}^l$. Функция запросов Q задает на множестве E_n отношение эквивалентности ρ следующего вида: для любых $i_1, i_2 \in E_n$, $i_1 \rho i_2$ тогда и только тогда, когда для любых $j \in E_2, r \in E_s$ верно $Q(j, i_1, r) = Q(j, i_2, r)$.

Нетрудно заметить, что это отношение делит множество E_n на s непересекающихся подмножеств $I_0, \ldots, I_{s-1}$, причем для любого $c \in E_{s-1}$ верно $I_c = \{cl', \ldots, (c+1)l' - 1\}, I_{s-1} = \{(s-1)l', \ldots, n-1\}$.

Для каждого $l \in E_z$ существуют такие $r_{q,1}^l, r_{q,2}^l \in E_s$ что для всех индексов $i_1 \in I_{c_{q,1}^l}, i_2 \in I_{c_{q,2}^l}$ верно $Q(0, i_1, r_{q,1}^l) = Q(0, i_2, r_{q,2}^l) = q$. Согласно лемме 2, $Q(0, i_1, q) = Q(0, i_2, q) = q$. Обозначим

$$Q(1, i_1, q) = q_{l,q,1}^1,$$
$$Q(1, i_2, q) = q_{l,q,2}^1.$$

Тогда

$$t_{q,1}^l = sc_{q,1}^l + q_{l,q,2}^1,$$
$$t_{q,2}^l = sc_{q,2}^l + q_{l,q,1}^1.$$

Если $l'' \neq 0$, то пусть E' — единичная матрица из $M(l'', l'')$, O' — нулевая матрица из $M(zl_0', l'')$, O'' — нулевая матрица из $M(l'', s^2 l_0')$, O''' — нулевая матрица из $M(sl_0', l'')$.

Тогда матрицы, используемые первым сервером S_0, имеют следующий вид:

$$B_q^0 = \begin{pmatrix} T_{v_0(q)}^0 & O' \end{pmatrix},$$

матрицы, используемые вторым сервером S_1, имеют следующий вид:

$$B_q^1 = \begin{pmatrix} T_{v_1(q)}^1 & O''' \\ O'' & E' \end{pmatrix},$$

где $T_{v_j(q)}^j \in \mathbf{T_j}, v_j(q) \in V_j, j \in E_2, q \in E_s$.

Шаг 3. Пользователь действует по следующей схеме.

1) $i < s^2 l'_0$.
   Если для создания запросов пользователь использовал индекс $i \in E_n$ и случайное число $r \in E_s$, то запросами к серверам была пара $(q^0, q^1) = (r, [i/s] + r \mod s)$.
   а) Если $i \mod s = q^1 = [i/s] + r \mod s$, то x_i извлекается из ответа второго сервера, а именно, $x_i = a^1_{i-q^1(n-sl'_0)}$.
   б) Если $i \mod s \neq q^1$, то x_i получается как сумма двух бит из ответов обоих серверов. А именно, находим число $l \in E_z$ такое что выполнены два условия:
   1. $\{[t^l_{q,1}/s], [t^l_{q,2}/s]\} \ni \{[i/s]\}$,
   2. $\{[t^l_{q,1} \mod s], [t^l_{q,2} \mod s]\} \ni \{[i \mod s]\}$.
   Пусть $\{[t^l_{q,1}/s], [t^l_{q,2}/s]\} = \{[i/s], c\}$ для некоторого $c \in E_s$.
   Тогда $x_i = a^0_{ll'_0+i \mod s^2} + a^1_{cl'_0+i \mod s^2} \mod 2$.
2) Если $i \geqslant s^2 l'_0$, то x_i извлекается из ответа второго сервера, а именно, $x_i = a^1_{i-(n-sl'_0)}$.

Легко видеть, что сложность этого протокола равна $C(I_{n,s}) = 2\,]\log_2 s[+ \frac{s(s-1)}{2} l'_0 + sl'_0 + l'' = 2\,]\log_2 s[+ \frac{s(s+1)}{2}[\frac{n}{s^2}] + n \mod s^2 \leqslant 2\,]\log_2 s[+ \frac{(s+1)}{2s} n + (n \mod s^2)$. Теорема доказана.

4. Пример PIR-протокола при $s = 2$

В качестве наглядной иллюстрации описанного выше PIR-протокола приведем невырожденный PIR-протокол $I_{n,2} = \langle Q, A^0, A^1, R\rangle$ для двух серверов, со случайным числом, принимающим два значения, для n кратного 4. Протокол состоит из трех шагов.
Шаг 1.

По номеру бита i и случайному числу $r \in E_2$ пользователь вычисляет $q^j, j \in E_2$, по правилу $q^0 = Q(0, i, r) = r$,

$$q^1 = Q(1, i, r) = \begin{cases} r, & \text{если } i < \frac{n}{2}, \\ (1 + r) \mod 2, & \text{если } i \geqslant \frac{n}{2}. \end{cases}$$

и посылает j-му серверу запрос q^j.

Шаг 2.

Каждый бит ответа первого сервера на любой запрос пользователя представляет собой сумму по модулю 2 ровно двух бит базы данных.

Если первый сервер S_0 получает в качестве запроса бит $q^0 = 0$, то он побитово складывает вторую четверть базы данных с третьей четвертью и посылает результат сложения, всего $\frac{n}{4}$ бит, в качестве ответа пользователю. Если сервер S_0 получает в качестве запроса бит $q^0 = 1$, то он побитово складывает первую четверть базы данных с последней четвертью и посылает результат пользователю.

Второй сервер на любой запрос пользователя всегда посылает в качестве ответа некоторую проекцию базы данных. Если сервер S_1 получает в качестве запроса бит $q^1 = 0$, то в качестве ответа он посылает пользователю первую и третью четверти базы данных, всего $\frac{n}{2}$ бит. Если сервер S_1 получает в качестве запроса бит $q^1 = 1$, то в качестве ответа он посылает пользователю вторую и последнюю четверти базы данных.

Если графически представить базу данных, как прямую, на которой расположены биты базы данных от x_0 до x_{n-1}, то алгоритм вычисления ответов серверами S_0 и S_1 будет выглядеть так, как показано на рисунке 1.

Шаг 3.

Используя i, r и ответы серверов a^0, a^1, пользователь вычисляет значение бита x_i как линейную комбинацию бит ответов серверов. Для любой пары запросов, пользователь должен иметь возможность определить значение либо первой либо второй половины базы данных. А именно, при запросах $(q^0, q^1) \in \{(0,0),(1,1)\}$ он может вычислить любой бит из первой половины базы данных, при запросах $(q^0, q^1) \in \{(0,1),(1,0)\}$ — любой бит из второй половины базы данных. По виду функции ответов ясно, что пользователь всегда может получать желаемый бит. Для примера рассмотрим пару запросов $(q^0, q^1) = (0,0)$. В этом случае, пользователь должен узнать первую половину базы данных. Первую четверть базы данных он берет из ответа второго сервера S_1, вторую четверть он получает, складывая побитово ответ первого сервера S_0 с третьей четвертью базы данных, которую он получил в открытом виде от второго сервера (см. рис. 2).

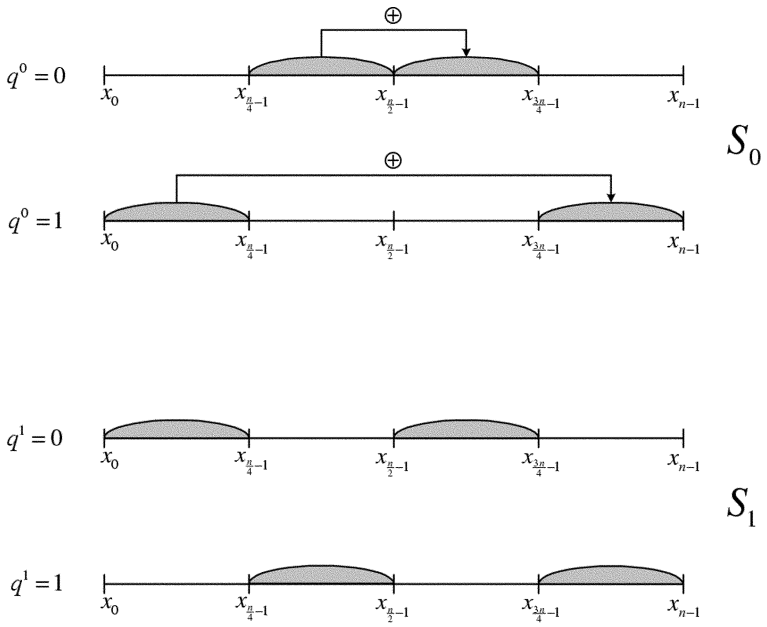

Рис. 1.

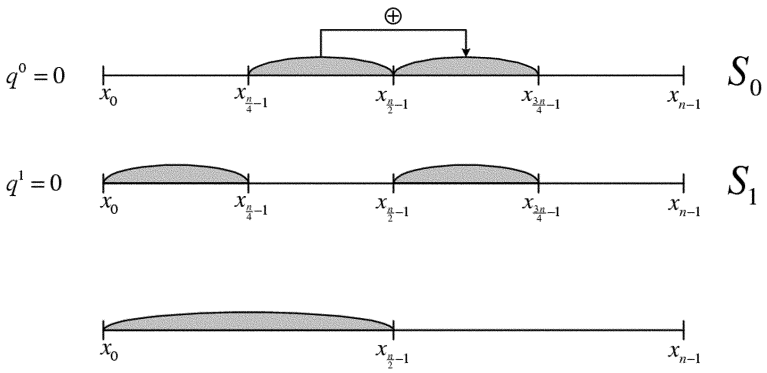

Рис. 2.

Легко видеть, что сложность этого протокола равна $C(I_{n,2}) = 2 + \frac{n}{4} + \frac{n}{2} = 2 + \frac{3n}{4}$.

5. Пример PIR-протокола при $s = 3$

Опишем невырожденный PIR-протокол $I_{n,3} = \langle Q, A^0, A^1, R\rangle$ для 2-х серверов с датчиком случайных чисел, принимающем три значения.

Обозначим $l'_0 = [n/9], l' = 3l'_0$, $l'' = n \mod 9 = n - 9l'_0$. Протокол состоит из трех шагов.

Шаг 1.

По номеру бита i и случайному числу $r \in E_3$ пользователь вычисляет $q^j, j \in E_3$, по правилу $q^0 = Q(0, i, r) = r$,

$$q^1 = Q(1,i,r) = \begin{cases} r, & \text{если } i < 3l'_0, \\ (1+r) \mod 3, & \text{если } 3l'_0 \leqslant i < 6l'_0, \\ (2+r) \mod 3, & \text{если } i \geqslant 6l'_0, \end{cases}$$

и посылает j-му серверу запрос q^j.

Шаг 2.

Для вычисления ответа сервер $S_j, j \in E_3$ использует матрицы B^j_0, B^j_1, B^j_2, причем $B^0_0, B^0_1, B^0_2 \in M(3l'_0, n)$, а $B^1_0, B^1_1, B^1_2 \in M(3l'_0 + l'', n)$. Полагаем, что пользователь знает вид всех таких матриц.

Сервер S_j вычисляет ответ a^j используя формулу $a^j = A^j(x, q^j) = (B^j_{q^j} x^T)^T$ и посылает результат пользователю.

Положим E — единичная матрица из $M(l'_0, l'_0)$, O — нулевая матрица из $M(l'_0, l'_0)$. Если $l'' \neq 0$, то E' — единичная матрица из $M(l'', l'')$, O' — нулевая матрица из $M(3l'_0, l'')$, O'' — нулевая матрица из $M(l'', 9l'_0)$.

Тогда матрицы, используемые первым сервером S_0 имеют следующий вид:

$$B^0_q = \begin{pmatrix} T^0_{v_0(q)} & O' \end{pmatrix}, q \in E_3,$$

$$T^0_{v_0(0)}, T^0_{v_0(1)}, T^0_{v_0(2)} \in M(3l'_0, 9l'_0),$$

$$T^0_{v_0(0)} = \begin{pmatrix} O & E & O & E & O & O & O & O & O \\ O & O & E & O & O & O & E & O & O \\ O & O & O & O & O & E & O & E & O \end{pmatrix},$$

$$T^0_{v_0(1)} = \begin{pmatrix} O & O & E & O & E & O & O & O & O \\ E & O & O & O & O & O & O & E & O \\ O & O & O & E & O & O & O & O & E \end{pmatrix},$$

$$T^0_{v_0(2)} = \begin{pmatrix} E & O & O & O & O & E & O & O & O \\ O & E & O & O & O & O & O & O & E \\ O & O & O & O & E & O & E & O & O \end{pmatrix},$$

матрицы используемые вторым сервером S_1 имеют следующий вид:

$$B^1_q = \begin{pmatrix} T^1_{v_1(q)} & O' \\ O'' & E' \end{pmatrix}, q \in E_3,$$

$$T^1_{v_1(0)}, T^1_{v_1(1)}, T^1_{v_1(2)} \in M(3l'_0 + l'', 9l'_0),$$

$$T^1_{v_1(0)} = \begin{pmatrix} E & O & O & O & O & O & O & O & O \\ O & O & O & E & O & O & O & O & O \\ O & O & O & O & O & O & E & O & O \end{pmatrix},$$

$$T^1_{v_1(1)} = \begin{pmatrix} O & E & O & O & O & O & O & O & O \\ O & O & O & O & E & O & O & O & O \\ O & O & O & O & O & O & O & E & O \end{pmatrix},$$

$$T^1_{v_1(2)} = \begin{pmatrix} O & O & E & O & O & O & O & O & O \\ O & O & O & O & O & E & O & O & O \\ O & O & O & O & O & O & O & O & E \end{pmatrix}.$$

Шаг 3.

Используя матрицы $B^0_{q^0}, B^1_{q^1}$ и ответы серверов a^0, a^1, пользователь строит систему из $3l'$ уравнений и $3l'$ неизвестных — бит базы данных. Таким образом, пользователь вычисляет значение бита x_i как линейную комбинацию бит ответов серверов. А именно: для запросов q^0, q^1 расширенная матрица R'_{q^0,q^1} системы линейных уравнений для переменных $x_0, \ldots, x_{3l'-1}$ имеет следующий вид:

$$R'_{q^0,q^1} = \left(\begin{array}{ccc|c} E & O & O & \\ O & E & O & C'_{q^0,q^1} \\ O & O & E & \end{array} \right),$$

где $(q^0, q^1) \in \{(0,0), (1,1), (2,2)\}$ и $C'_{q^0,q^1} \in M(3l', 1)$,

$$C'_{0,0} = \begin{pmatrix} a^1_0 \\ \cdots \\ a^1_{l'-1} \\ a^0_0 + a^1_{l'} \\ \cdots \\ a^0_{l'-1} + a^1_{2l'-1} \\ a^0_{l'} + a^1_{2l'} \\ \cdots \\ a^0_{2l'-1} + a^1_{3l'-1} \end{pmatrix}, \quad C'_{1,1} = \begin{pmatrix} a^0_{l'} + a^1_{2l'} \\ \cdots \\ a^0_{2l'-1} + a^1_{3l'-1} \\ a^1_0 \\ \cdots \\ a^1_{l'-1} \\ a^0_0 + a^1_{l'} \\ \cdots \\ ,a^0_{l'-1} + a^1_{2l'-1} \end{pmatrix},$$

$$C'_{2,2} = \begin{pmatrix} a^0_0 + a^1_{l'} \\ \cdots \\ a^0_{l'-1} + a^1_{2l'-1} \\ a^0_{l'} + a^1_{2l'} \\ \cdots \\ a^0_{2l'-1} + a^1_{3l'-1} \\ a^1_0 \\ \cdots \\ a^1_{l'-1} \end{pmatrix}.$$

Расширенная матрица R''_{q^0,q^1} системы уравнений для $x_{3l'}, \ldots, x_{6l'-1}$ имеет следующий вид:

$$R''_{q^0,q^1} = \left(\begin{array}{ccc|c} E & O & O & \\ O & E & O & C''_{q^0,q^1} \\ O & O & E & \end{array} \right),$$

где $(q^0, q^1) \in \{(0,1),(1,2),(2,0)\}$,

$$C''_{0,1} = \begin{pmatrix} a^0_0 + a^1_0 \\ \cdots \\ a^0_{l'-1} + a^1_{l'-1} \\ a^1_{l'} \\ \cdots \\ a^1_{2l'-1} \\ a^0_{2l'} + a^1_{2l'} \\ \cdots \\ a^0_{3l'-1} + a^1_{3l'-1} \end{pmatrix}, \quad C''_{1,2} = \begin{pmatrix} a^0_{2l'} + a^1_{2l'} \\ \cdots \\ a^0_{3l'-1} + a^1_{3l'-1} \\ a^0_0 + a^1_0 \\ \cdots \\ a^0_{l'-1} + a^1_{l'-1} \\ a^1_{l'} \\ \cdots \\ a^1_{2l'-1} \end{pmatrix},$$

$$C''_{2,0} = \begin{pmatrix} a^1_{l'} \\ \cdots \\ a^1_{2l'-1} \\ a^0_{2l'} + a^1_{2l'} \\ \cdots \\ a^0_{3l'-1} + a^1_{3l'-1} \\ a^0_0 + a^1_0 \\ \cdots \\ a^0_{l'-1} + a^1_{l'-1} \end{pmatrix}.$$

Расширенная матрица R'''_{q^0,q^1} системы уравнений для $x_{6l'}, \ldots, x_{9l'-1}$ имеет следующий вид:

$$R'''_{q^0,q^1} = \left(\begin{array}{ccc|c} E & O & O & \\ O & E & O & C'''_{q^0,q^1} \\ O & O & E & \end{array} \right),$$

где $(q^0, q^1) \in \{(0,2), (1,0), (2,1)\}$,

$$C'''_{0,2} = \begin{pmatrix} a^0_{l'} + a^1_0 \\ \cdots \\ a^0_{2l'-1} + a^1_{l'-1} \\ a^0_{2l'} + a^1_{l'} \\ \cdots \\ a^0_{3l'-1} + a^1_{2l'-1} \\ a^1_{2l'} \\ \cdots \\ a^1_{3l'-1} \end{pmatrix}, \quad C'''_{1,0} = \begin{pmatrix} a^1_{2l'} \\ \cdots \\ a^1_{3l'-1} \\ a^0_{l'} + a^1_0 \\ \cdots \\ a^0_{2l'-1} + a^1_{l'-1} \\ a^0_{2l'} + a^1_{l'} \\ \cdots \\ a^0_{3l'-1} + a^1_{2l'-1} \end{pmatrix},$$

$$C'''_{2,1} = \begin{pmatrix} a^0_{2l'} + a^1_{l'} \\ \cdots \\ a^0_{3l'-1} + a^1_{2l'-1} \\ a^1_{2l'} \\ \cdots \\ a^1_{3l'-1} \\ a^0_{l'} + a^1_0 \\ \cdots \\ a^0_{2l'-1} + a^1_{l'-1} \end{pmatrix}.$$

Расширенная матрица $R^{(4)}$ системы уравнений для $x_{9l'}, \ldots, x_n$ имеет следующий вид:

$$R^{(4)} = \left(\begin{array}{c|c} & \\ E' & C^{(4)} \\ & \end{array} \right),$$

где

$$C^{(4)} = \begin{pmatrix} a^1_{4l'} \\ \cdots \\ a^1_{4l'+l''-1} \end{pmatrix}.$$

Легко видеть, что сложность этого протокола равна $C(I_{n,3}) = 4 + 6l'_0 + l'' = 4 + 6[\frac{n}{s^2}] + n \mod s^2$.

6. Нижняя оценка

Лемма 3. *Для произвольного* $(2, n, s, p)$ *PIR-протокола, где* $p = p^0 + p^1$, *верно* $p^0 + sp^1 \geqslant n$ *и* $p^1 + sp^0 \geqslant n$.

Доказательство. Рассмотрим произвольный $(2, n, s, p)$ PIR-протокол $I = \langle Q, A^0, A^1, R\rangle$, где $p = p^0 + p^1$.

Пусть $x \in \{0,1\}^n$, $q \in E_s$. Определим следующий вектор

$$\mathcal{A}(q, x) \quad = \quad (A(0, x, q), A(1, x, 0), \ldots, A(1, x, s-1)).$$

Длина вектора $\mathcal{A}(q, x)$ равна $p^0 + sp^1$.

Пусть отображение $R' : E_n \times E_s \times \{0,1\}^{p^0+sp^1} \to \{0,1\}$ задано следующим образом

$$\begin{aligned} R'(i, r, \mathcal{A}(Q(0, i, r), x)) \quad &= \quad R(i, r, A(0, x, Q(0, i, r)), \\ &\qquad A(1, x, Q(1, i, r))) = x_i. \end{aligned}$$

Отметим, что каждый из векторов $A^j(Q(j, i, r), x), j \in E_2$ содержится в векторе $\mathcal{A}(Q(0, i, r), x)$.

Предположим, что длина вектора $\mathcal{A}$ меньше чем n. Тогда существует число $q \in E_s$ и два вектора $x', x'' \in \{0,1\}^n$, $x' \neq x''$ такие что $\mathcal{A}(q, x') = \mathcal{A}(q, x'')$.

Поскольку $x' \neq x''$, существует индекс $l \in E_n$ такой что $x'_l \neq x''_l$. Пусть r такое, что $Q(0, l, r) = q$. Тогда

$$x'_l = R'(l, r, \mathcal{A}(q, x')) = R'(l, r, \mathcal{A}(q, x'')) = x''_l.$$

Это противоречие доказывает, что $p^0 + sp^1 \geqslant n$. Второе неравенство доказывается аналогично.

Рассмотрим произвольный $(2, n, s, p)$ PIR-протокол $I = \langle Q, A^0, A^1, R\rangle \in \mathcal{A}_2$.

Для любой булевой функции $f(x_0, \ldots, x_{l-1})$ через $M(f) \subseteq E_l$ обозначим множество номеров переменных, от которых она существенно зависит.

Будем говорить, что бит с индексом $i \in E_n$ участвует в ответе сервера S_j на запрос $q \in E_s$ в закрытом виде, или является закрытым битом этого ответа, если существуют $h \geqslant 2, l \in E_{p^j}$ такие, что $i \in V(A^j_l(q))$ и $S(A^j_l(q)) = h$. В этом случае будем говорить, что бит i используется в компоненте A^j_l функции ответа сервера S_j на запрос q. Если $i \in V(A^j_l(q))$ и $S(A^j_l(q)) = 1$, то будем говорить, что бит участвует в ответе в открытом виде или является открытым битом этого ответа, и, если для любого $l \in E_{p^j}$ верно $V(A^j_l(q)) \cap \{i\} = \varnothing$, то бит не участвует в ответе.

Поскольку вид всех компонент функций ответов серверов известен всем участникам протокола, и любая функция, которая существенно зависит ровно от одной переменной — это либо некоторый бит базы данных либо его отрицание, то будем считать, что любая компонента, степень существенности которой равна 1, является просто некоторым битом базы данных.

Тогда для любых $j \in E_2, q \in E_s, h \in E_2$ положим

$$A^j_{q,h} = \{V(A^j_l(q)) : l \in E_{p^j}, S(A^j_l(q)) = h + 1\}$$

— множество индексов бит, которые участвуют в функциях, степень существенности которых равна $h + 1$, в ответе $A^j(q, x)$ сервера S_j на запрос q. Отметим, что для каждого множества $A \in A^j_{q,h}$ верно $|A| = h + 1$.

Для любых $B \subseteq E_n, j \in E_2, q \in E_s, h \in E_2$ положим

$$A^j_{q,h}(B) = \{M \in A^j_{q,h} : M \cap B \neq \varnothing\},$$

тогда $A^j_{q,h} = A^j_{q,h}(E_n)$, для любых $j \in E_2, q \in E_s, h \in E_2$.

Согласно лемме 2 можно считать, что

$$Q(0,i,r) = q^0 = r,$$
$$Q(1,i,r) = q^1 = q^1(i,r) = q_r(i) \in E_s,$$
$$q_r^{-1}(l) = \{i : q_r(i) = l\}, l \in E_s,$$

положим $A^{0,l}_{r,h} = A^0_{r,h}(q_r^{-1}(l))$.

Тогда $|A^{0,l}_{r,h}|$ — множество бит из $A^0_{r,h}$, которые используются в паре с ответом $a^1 = A^1(l,x)$. То есть количество компонент a^0, используя которые вместе с некоторыми компонентами ответа a^1 пользователm может узнать значение искомого бита x_i базы данных.

При этом существует такое $M \in A^{0,l}_{r,h}$, что $i \in M$, то есть искомый бит используется в некоторой компоненте с битами из M.

Ясно, что

$$A^0_{r,h} = \bigcup_{l \in E_s} A^{0,l}_{r,h}.$$

Для любых $B \subseteq E_n, j \in E_2, q, l \in E_s$ положим

$$A^{j,l}_{q,h}(B) = \{M \in A^{j,l}_{q,h} : M \cap B \neq \varnothing\}.$$

Для любого множества $A \subseteq 2^{E_n}$ положим $V(A) = \bigcup\limits_{B \in A} B$.

В лемме 3 было показано, что если второй сервер получает запрос q^1 то должно выполняться неравенство $p^1 + sp^0 \geqslant n$, другими словами, используя один ответ второго сервера и все ответы первого сервера, пользователь узнает всю базу данных. Учитывая, что пользователь знает вид всех функций ответов, попробуем найти соотношение между количеством всех функций, степень существенности которых больше или равна 2 и функций, степень существенности которых равна 1, в ответах серверов, достаточных для выполнения этого неравенства.

Пусть $1 \leqslant m \leqslant s$, $\{q_0, \ldots, q_{m-1}\}, \subseteq E_s$ — произвольное упорядоченное множество. Для любых $j \in E_2, q \in E_s, 1 \leqslant l \leqslant m-1$ положим

$$\widehat{A}^j_{q_l,0} = V(A^j_{q_l,0}) \backslash \{V(A^j_{q_0,0}) \cup V(A^j_{q_1,0}) \cup \ldots \cup V(A^j_{q_{l-1},0}\},$$
$$\widehat{A}^{j,q_l}_{q,0} = A^{j,q_l}_{q,0} \backslash \{A^{1-j}_{q_0,0} \cup A^{1-j}_{q_1,0} \cup \ldots \cup A^{1-j}_{ql-1,0}\},$$
$$\widehat{A}^{j,q_l}_{q,1} = \{M \in A^{j,q_l}_{q,1} : \{M \backslash A^{1-j}_{q_l,0}\} \in E_n \backslash \{A^{1-j}_{q_0,0} \cup A^{1-j}_{q_1,0} \cup \ldots \cup A^{1-j}_{q_{l-1},0}\}\},$$

$$\widehat{A}^{j}_{q_0,0} = A^{j}_{q_0,0}, \widehat{A}^{j,q_0}_{q,0} = A^{j,q_0}_{q,0}, \widehat{A}^{j,q_0}_{q,1} = A^{j,q_0}_{q,1}.$$

Множество $A^{j,q_l}_{q,1}$ состоит из бит, которые используются в компонентах функции ответов $A^j(q,x)$, существенность которых равна 2, которые также используются в паре с ответом $A^{1-j}(q_l,x)$. Рассмотрим только те компоненты, которые при использовании в паре с ответом $A^{1-j}(q_l,x)$ дают биты из множества $\widehat{E}_n(j,l) = E_n \backslash \{A^{1-j}_{q_0,0} \cup A^{1-j}_{q_1,0} \cup \ldots \cup A^{1-j}_{q_{l-1},0}\}$.

Лемма 4. *Для любого $(2,n,s,p)$ PIR-протокола $I = \langle Q, A^0, A^1, R\rangle \in \mathcal{A}_2$, любого $1 \leqslant m \leqslant s$, любого упорядоченного подмножества $\{q_0,\ldots,q_{m-1}\} \subseteq E_s$, верны неравенства*

$$\begin{aligned}\sum_{q\in E_s}\sum_{l\in E_m}|\widehat{A}^{0,q_l}_{q,1}| \;&\geqslant\; mn - \sum_{l\in E_{m-1}}(m-l-1)|\widehat{A}^{1}_{q_l,0}| - \sum_{l\in E_m}|\widehat{A}^{1}_{q_l,0}| - \\ &- \sum_{q\in E_s}\sum_{l\in E_m}|\widehat{A}^{0,q_l}_{q,0}| - \sum_{l\in E_m}|A^{1}_{q_l,1}|, \qquad (1)\end{aligned}$$

$$\begin{aligned}\sum_{q\in E_s}\sum_{l\in E_m}|\widehat{A}^{1,q_l}_{q,1}| \;&\geqslant\; mn - \sum_{l\in E_{m-1}}(m-l-1)|\widehat{A}^{0}_{q_l,0}| - \sum_{l\in E_m}|\widehat{A}^{0}_{q_l,0}| - \\ &- \sum_{q\in E_s}\sum_{l\in E_m}|\widehat{A}^{1,q_l}_{q,0}| - \sum_{l\in E_m}|A^{0}_{q_l,1}|. \qquad (2)\end{aligned}$$

Доказательство. Докажем первое неравенство индукцией по длине набора m. Основание индукции. Пусть $m = 1$.

Допустим, сервер S_1 получил запрос $q^1 = q_0 \in E_s$. Поскольку каждый бит ответа получен либо с помощью существенной функции, степень существенности которой не превышает d, либо является битом базы данных, и пользователь знает вид всех функций ответов, можно указать максимальное количество бит, которое пользователь может узнать из ответов $A^1(q_0,x), A^0(0,x),\ldots,A^0(s-1,x)$.

Во-первых, пользователь может узнать все биты, которые используются во всех компонентах функции ответа сервера $A^1(q_0,x)$, всего $|\bigcup\limits_{h\in E_2} V(A^1_{q_0,h})|$ бит. Также, он может узнать все биты из всех компонент функции ответов сервера S_0, которые могут использоваться в паре с ответом сервера S_1 на запрос q_0, всего $|\bigcup\limits_{h\in E_2}\bigcup\limits_{q\in E_s} A^{0,q_0}_{q,h}|$ бит.

Тогда неравенство из леммы 3 примет следующий вид:

$$|\bigcup_{h\in E_2} V(A^1_{q_0,h})| + |\bigcup_{h\in E_2}\bigcup_{q\in E_s} A^{0,q_0}_{q,h}| \geqslant n.$$

Для того, чтобы получить значение битов из линейной комбинации длины 2, необходимо знать один из битов этой линейной комбинации. Таким образом, поскольку мы учитываем все открытые биты сервера S_0, которые используются в паре с сервером S_1 и учитывая

$$|\bigcup_{h\in E_2}\bigcup_{q\in E_s} \widehat{A}^{0,q_0}_{q,h}| \leqslant \sum_{h\in E_2}\sum_{q\in E_s} |\widehat{A}^{0,q_0}_{q,h}|$$

можно записать следующее неравенство

$$\sum_{h\in E_2} |A^1_{q_0,h}| + \sum_{h\in E_2}\sum_{q\in E_s} |A^{0,q_0}_{q,h}| \geqslant n.$$

Шаг индукции. Допустим, неравенство верно для m, покажем, что оно также выполняется и для $m+1$. Пусть для некоторого набора $\{q_0,\ldots,q_{m-1}\} \subseteq E_s, 1 \leqslant m \leqslant s$ выполнено неравенство (1).

Это неравенство учитывает все компоненты функции ответов во всех ответах сервера S_0, которые используются в паре с ответами сервера S^1 на запросы $q_l, l \in E_m$.

Рассмотрим некоторый запрос $q_m \in E_s\backslash\{q_0,\ldots,q_{m-1}\}$.

Выпишем неравенство из леммы 3 для базы данных $(x_{i_1},\ldots,x_{i_{l_1}})$, где $\{i_1,\ldots,i_{l_1}\} = \widehat{E}_n = E_n\backslash \bigcup\limits_{l\in E_m} V(A^1_{q_l,0})$, для случая, когда сервер S_1 получает запрос q_m. Для того, чтобы получить весь вектор $(x_{i_1},\ldots,x_{i_{l_1}})$, пользователь может использовать $|A^1_{q_m,0}\backslash \bigcup\limits_{l\in E_m} V(A^1_{q_l,0})| = |V(\widehat{A}^1_{q_m,0})|$ открытых бит из ответа $A^1(q_m,x)$, индексы которых лежат в $\widehat{E}_n$ и $|A^1_{q_m,1}|$ закрытых бит из всех компонент того же ответа. Также пользователь может узнать все биты

$$|\bigcup_{h\in E_2}\bigcup_{q\in E_s} A^{0,q_m}_{q,h}\backslash \bigcup_{l\in E_m} V(A^1_{q_l,0})| = |\bigcup_{h\in E_2}\bigcup_{q\in E_s} \widehat{A}^{0,q_m}_{q,h}|$$

из всех ответов сервера S_0, которые при использовании в паре с ответом $A^1(q_m,x)$ дают биты с индексами из $\widehat{E}_n$.

Учитывая

$$|\bigcup_{h\in E_2} V(\widehat{A}^1_{q_m,0})| \leqslant |\widehat{A}^1_{q_m,0}|,$$

и

$$|\bigcup_{h\in E_2}\bigcup_{q\in E_s} \widehat{A}^{0,q_m}_{q,h}| \leqslant \sum_{h\in E_2}\sum_{q\in E_s} |\widehat{A}^{0,q_m}_{q,h}|,$$

получаем

$$|\widehat{A}^1_{q_m,0}| + |A^1_{q_m,1}| + \sum_{h\in E_2}\sum_{q\in E_s} |\widehat{A}^{0,q_m}_{q,h}| \geqslant n - \sum_{l\in E_m} |\widehat{A}^1_{q_l,0}|.$$

Складываем последних два неравенства, получаем

$$\sum_{q\in E_s}\sum_{l\in E_{m+1}} |\widehat{A}^{0,q_l}_{q,1}| \geqslant (m+1)n - \sum_{l\in E_m}(m-l)|\widehat{A}^1_{q_l,0}| - \sum_{l\in E_{m+1}} |\widehat{A}^1_{q_l,0}| - \\ - \sum_{q\in E_s}\sum_{l\in E_{m+1}} |\widehat{A}^{0,q_l}_{q,0}| - \sum_{l\in E_{m+1}} |A^1_{q_l,1}|,$$

Второе неравенство доказывается аналогично. Тем самым лемма 4 доказана.

Положим

$$\mathbb{D}_{n,s} = \{\{D_0,\dots,D_{s-1}\}, D_l \subseteq E_n, l \in E_s\}.$$

Для множеств из $\mathbb{D}_{n,s}$ будем использовать более короткое обозначение: $\{D_t\}_{t\in E_s}$.

Пусть $\tau : E_s \to E_s$ — перестановка в E_s. Для любого $t \in \{1,\dots,s{-}1\}$ положим

$$\widehat{D}_{\tau(t)} = D_{\tau(t)} \backslash (D_{\tau(0)} \cup D_{\tau(1)} \cup \ldots \cup D_{\tau(t-1)}),$$
$$\widehat{D}_{\tau(0)} = D_{\tau(0)}.$$

Для любых натуральных n, s, любой перестановки $\tau : E_s \to E_s$, любого множества $\{D_t\}_{t\in E_s}$ из $\mathbb{D}_{n,s}$ положим

$$F_{n,s}(\{D_t\}_{t\in E_s}) = \min_{\tau: E_s\to E_s} \sum_{t\in E_s}(s-t)|\widehat{D}_{\tau(t)}| - \sum_{t\in E_s}|D_t|. \qquad (3)$$

Лемма 5. *Для любых натуральных n, s, таких что $n \geqslant s$, и любого множества $\{D_t\}_{t\in E_s}$ из $\mathbb{D}_{n,s}$ верно:*

$$F_{n,s}(\{D_t\}_{t\in E_s}) \leqslant \frac{s-1}{2}n + \frac{s(s+1)}{2}(\operatorname{sign}(d_2) - \frac{d_2}{s}),$$

где $d_2 = n \mod s$.

Доказательство этого утверждения очень объемно, ограничение на размер статьи не позволяет привести его целиком, поэтому здесь мы приведем основную идею.

Пусть $\{D_t\}_{t\in E_s}$ — произвольное множество из $\mathbb{D}_{n,s}$, τ — некоторая перестановка. Рассмотрим функцию

$$F_\tau(\{D\}_{t\in E_s}) = \sum_{t\in E_s}(s-t)|\widehat{D}_{\tau(t)}| - \sum_{t\in E_s}|D_t|.$$

Положим l — количество ненулевых множеств $\widehat{D}_{\tau(t)}$, $\{0, t_1, \ldots, t_{l-1}\} = \{t : \widehat{D}_{\tau(t)} \neq \varnothing\}$ — номера, соответствующие ненулевым множествам $\widehat{D}_{\tau(t)}$.

Заметим, что первое слагаемое суммы в точности соответствует площади фигуры под отрезками (включая сами отрезки) с координатами

$$(0, s), (|D_{\tau(0)}|, s),$$
$$(|D_{\tau(0)}|, s-1), (|D_{\tau(0)}| + |\widehat{D}_{\tau(t_1)}|, s-1),$$
$$\ldots,$$
$$(|D_{\tau(0)}| + \ldots + |\widehat{D}_{\tau(t_{l-2})}|, 1), (|D_{\tau(0)}| + \ldots + |\widehat{D}_{\tau(t_{l-1})}|, 1).$$

Здесь под площадью фигуры понимается число точек с натуральными координатами в этой фигуре. Второе слагаемое равно площади столбиков (ступенек), соответствующих множествам $D_{\tau(t)}$, $t \in E_s$, (см. рис. 3). Тогда значение функции $F_\tau(\{D\}_{t\in E_s})$ равно площади фигуры под столбиками (не включая столбики).

Оптимальным назовем такое множество $\{D_t\}_{t\in E_s}$ из $\mathbb{D}_{n,s}$, на котором функция $F_{n,s}(\{D\}_{t\in E_s})$ достигает максимума.

Значение функции равно площади описанной ранее фигуры в случае минимальной перестановки множеств $D_t, t \in E_s$. Покажем, что

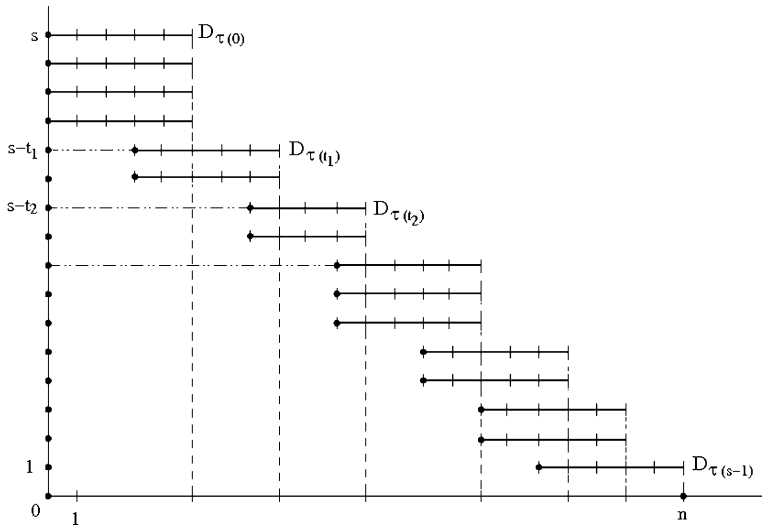

Рис. 3.

фигура, которая соответствует минимальной перестановке, а именно, кривая, проведенная по нижним левым точкам столбиков, — всегда невыпуклая. Допустим, это не так, пусть для некоторого множества, кривая, соответствующая минимальной перестановке, является выпуклой (см. рис. 4).

Тогда рассмотрим обратную перестановку. Ясно, что мы получим вогнутую кривую, соответствующую фигуре с меньшей площадью (см. рис. 5). Получаем, что в случае новой перестановки значение функции уменьшилось, то есть начальная перестановка не была минимальной.

Если кривая состоит из невыпуклых и невогнутых кривых одновременно, мы всегда можем рассмотреть каждый участок кривой в отдельности, для каждого выпуклого участка кривой подбирая перестановку так, что в результате получим невыпуклую кривую.

Таким образом, из геометрических соображений ясно, что площадь фигуры, соответствующей оптимальному множеству, будет ограничена невыпуклой кривой наиболее близкой к прямой линии.

Нетрудно заметить, что оптимальное множество $\{D_t\}_{t\in E_s}$ состоит

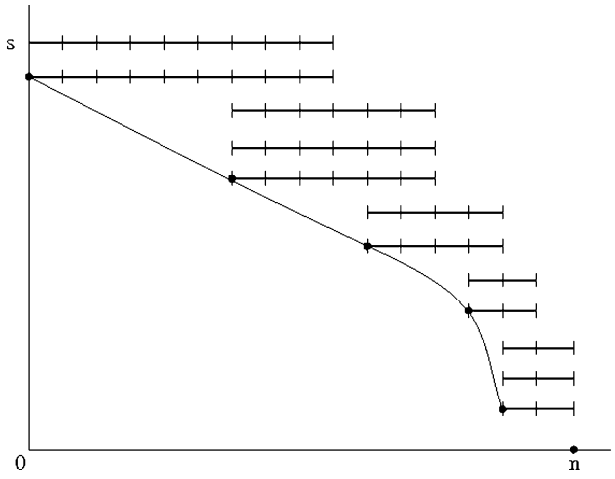

Рис. 4.

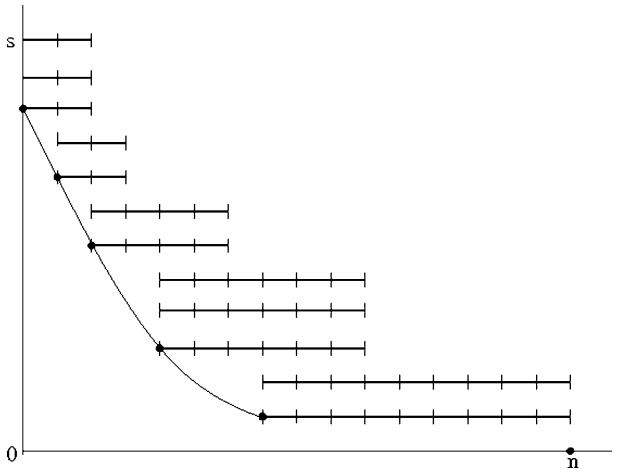

Рис. 5.

из попарно непересекающихся множеств с почти одинаковой мощностью (см. рис. 6), а именно, когда для любых $t_1, t_2 \in E_s, t_1 \neq t_2$ верно $D_{t_1} \cap D_{t_2} = \varnothing$, $||D_{t_1}| - |D_{t_2}|| \leqslant 1$.

Поскольку для любого $t \in E_s$ верно $D_t \subseteq E_n$, получаем $\sum\limits_{t \in E_s} |D_t| = n$ и

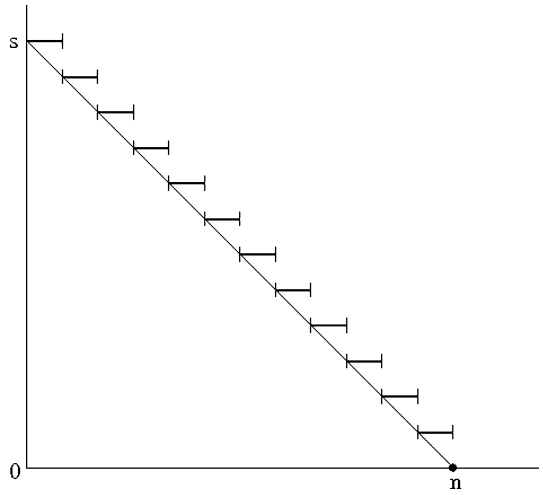

Рис. 6.

$$F_{n,s}(\{D_t\}_{t\in E_s}) = \min_{\tau:E_s\to E_s}\sum_{t\in E_s}(s-t)|\widehat{D}_{\tau(t)}| - \sum_{t\in E_s}|D_t| \leqslant$$

$$\leqslant \min_{\tau:E_s\to E_s}\sum_{t\in E_s}(s-t)|D_t| - n \leqslant \Big]\frac{n}{s}\Big[\sum_{t\in E_s}(s-t) - n = \frac{s(s+1)}{2}\Big]\frac{n}{s}\Big[- n.$$

Пусть $n = d_1 s + d_2$. Поскольку $\frac{n}{s} = d_1 + \frac{d_2}{s}$, получаем

$$\Big]\frac{n}{s}\Big[= d_1 + \operatorname{sign}(d_2) = \frac{n}{s} + \operatorname{sign}(d_2) - \frac{d_2}{s},$$

тогда

$$F_{n,s}(\{D_t\}_{t\in E_s}) \leqslant \frac{s(s+1)}{2}\left(\frac{n}{s} + \operatorname{sign}(d_2) - \frac{d_2}{s}\right) - n =$$

$$= \frac{s-1}{2}n + \frac{s(s+1)}{2}\left(\operatorname{sign}(d_2) - \frac{d_2}{s}\right).$$

Лемма доказана.

Доказательство теоремы 2.

Рассмотрим произвольный $(2, n, s, p)$ PIR-протокол $I = \langle Q, A^0, A^1, R\rangle \in \mathcal{A}_2$. Для этого протокола верны неравенства из леммы 4. Пусть $\mathcal{Q}^0 = \{q_0^0, \ldots, q_{s-1}^0\}, \mathcal{Q}^1 = \{q_0^1, \ldots, q_{s-1}^1\} \subseteq E_s$ — произвольные упорядоченные множества.

Сложим первое неравенство из леммы 4 для множества $\{q_0^1, \ldots, q_{s-1}^1\}$ со вторым неравенством из этой же леммы для множества $\{q_0^0, \ldots, q_{s-1}^0\}$, получим следующее соотношение:

$$\sum_{q\in E_s}\sum_{l\in E_s}(|\widehat{A}_{q,1}^{0,q_l^1}| + |\widehat{A}_{q,1}^{1,q_l^0}|) \geqslant 2sn - \sum_{l\in E_{s-1}}(s-l-1)(|\widehat{A}_{q_l^1,0}^{1}| + |\widehat{A}_{q_l^0,0}^{0}|) -$$
$$-\sum_{l\in E_s}(|\widehat{A}_{q_l^1,0}^{1}|+|\widehat{A}_{q_l^0,0}^{0}|)-\sum_{q\in E_s}\sum_{l\in E_s}(|\widehat{A}_{q,0}^{0,q_l^1}|+|\widehat{A}_{q,0}^{1,q_l^0}|)-\sum_{l\in E_s}(|A_{q_l^1,1}^{1}|+|A_{q_l^0,1}^{0}|). \tag{4}$$

Как было показано в лемме 1, для любых $j \in E_2, i \in E_n$ не существуют таких $r_1, r_2 \in E_s, r_1 \neq r_2$ что $Q(j, i, r_1) = Q(j, i, r_2)$, следовательно, каждый бит каждой компоненты $A_{q,h}^j$ учитывается в сумме $\sum\limits_{l\in E_s} |\widehat{A}_{q,h}^{j,q_l^{1-j}}|$ не более чем один раз.

Тогда для любых $q \in E_s, j \in E_2$ верно

$$\sum_{h\in E_2}\sum_{l\in E_s}|\widehat{A}_{q,h}^{j,q_l^{1-j}}| \leqslant \sum_{h\in E_2}|A_{q,h}^{j}|,$$

и следовательно

$$\sum_{q\in E_s}(|A_{q,1}^{0}| + |A_{q,1}^{1}|) \geqslant 2sn - \sum_{l\in E_{s-1}}(s-l-1)(|\widehat{A}_{q_l^1,0}^{1}| + |\widehat{A}_{q_l^0,0}^{0}|) -$$
$$-\sum_{l\in E_s}(|\widehat{A}_{q_l^1,0}^{1}| + |\widehat{A}_{q_l^0,0}^{0}|) - \sum_{q\in E_s}(|A_{q,0}^{0}| + |A_{q,0}^{1}|) - \sum_{l\in E_s}(|A_{q_l^1,1}^{1}| + |A_{q_l^0,1}^{0}|),$$

или

$$2\sum_{q\in E_s}\sum_{l\in E_s}(|\widehat{A}_{q,1}^{0}| + |\widehat{A}_{q,1}^{1}|) \geqslant 2sn - \sum_{l\in E_s}(s-l)(|\widehat{A}_{q_l^1,0}^{1}| + |\widehat{A}_{q_l^0,0}^{0}|) -$$
$$- \sum_{q\in E_s}(|A_{q,0}^{0}| + |A_{q,0}^{1}|),$$

Неравенство (4) должно быть выполнено для любой последовательности запросов для каждого конкретного протокола, следовательно,

$$2\sum_{q\in E_s}(|\widehat{A}^0_{q,1}|+|\widehat{A}^1_{q,1}|)\geqslant \min_{A^j_{q,0}\subseteq E_n, q\in E_s, j\in E_2}\max_{\mathcal{Q}^0,\mathcal{Q}^1\subseteq E_s}[2sn-$$
$$-\sum_{l\in E_s}(s-l)(|\widehat{A}^1_{q^1_l,0}|+|\widehat{A}^0_{q^0_l,0}|)-\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)]=$$
$$=2sn-\max_{A^j_{q,0}\subseteq E_n, q\in E_s, j\in E_2}\min_{\mathcal{Q}^0,\mathcal{Q}^1\subseteq E_s}\sum_{l\in E_s}(s-l)(|\widehat{A}^1_{q^1_l,0}|+|\widehat{A}^0_{q^0_l,0}|)+$$
$$+\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|).$$

Прибавим к обеим частям неравенства выражение: $4s\,]\log_2 s[+2\sum\limits_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)$, получим

$$4s\,]\log_2 s[+2\sum_{h\in E_2}\sum_{q\in E_s}(|\widehat{A}^0_{q,h}|+|\widehat{A}^1_{q,h}|)=2sC(I)\geqslant 4s\,]\log_2 s[+2sn+$$
$$+2\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)-\max_{\substack{A^j_{q,0}\subseteq E_n,\\ q\in E_s, j\in E_2}}\min_{\mathcal{Q}^0,\mathcal{Q}^1\subseteq E_s}\sum_{l\in E_s}(s-l)(|\widehat{A}^1_{q^1_l,0}|+|\widehat{A}^0_{q^0_l,0}|)+$$
$$+\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|).$$

Из леммы 5 следует следующее неравенство: для любых n, s верно:

$$\max_{\substack{A^j_{q,0}\subseteq E_n,\\ q\in E_s, j\in E_2}}\min_{\mathcal{Q}^0,\mathcal{Q}^1\subseteq E_s}\sum_{l\in E_s}(s-l)(|\widehat{A}^1_{q^1_l,0}|+|\widehat{A}^0_{q^0_l,0}|)+\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)-$$
$$-\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)\leqslant \max_{\substack{A^j_{q,0}\subseteq E_n,\\ q\in E_s, j\in E_2}}\min_{\mathcal{Q}^0,\mathcal{Q}^1\subseteq E_s}\sum_{l\in E_s}(s-l)(|\widehat{A}^1_{q^1_l,0}|+|\widehat{A}^0_{q^0_l,0}|)+$$
$$+\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)-2\max_{\substack{A^j_{q,0}\subseteq E_n,\\ q\in E_s, j\in E_2}}\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)=$$

$$= \max_{\substack{A^j_{q,0}\subseteq E_n,\\ q\in E_s, j\in E_2}} \min_{\mathcal{Q}^0,\mathcal{Q}^1\subseteq E_s} \sum_{l\in E_s}(s-l)(|\widehat{A}^1_{q^1_l,0}|+|\widehat{A}^0_{q^0_l,0}|)-\sum_{q\in E_s}(|A^0_{q,0}|+|A^1_{q,0}|)\leqslant$$

$$\leqslant (s-1)n+s(s+1)\left(\operatorname{sign}(d_2)-\frac{d_2}{s}\right),$$

где $d_2 = n \mod s$. Тогда

$$2sC(I)\geqslant 2sn+4s]\log_2 s[-(s-1)n-s(s+1)\left(\operatorname{sign}(d_2)-\frac{d_2}{s}\right)=$$

$$=4s]\log_2 s[+(s+1)n-s(s+1)\left(\operatorname{sign}(d_2)-\frac{d_2}{2s}\right).$$

$$C(I)\geqslant 2]\log_2 s[+\frac{s+1}{2s}n-\frac{s+1}{2}\left(\operatorname{sign}(d_2)-\frac{d_2}{s}\right).$$

$$C(2,n,s,\mathcal{A}_2)=C(2,n,s,\mathcal{D}_2)\geqslant$$

$$\geqslant 2]\log_2 s[+\frac{s+1}{2s}n-\frac{s+1}{2}\left(\operatorname{sign}(d_2)-\frac{d_2}{s}\right).$$

Теорема доказана.

Список литературы

[1] Chor B., Goldreich O., Kushilevitz E., Sudan M. Private information retrieval // Proc. of the 36th Annu. IEEE Symp. on Foundations of Computer Science. P. 41–51. 1995.

[2] Beimel A., Ishai Y., Kushilevitz E., Raymond J.-F. Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval // Proc. of the 43st IEEE Sym. on Found. of Comp. Sci. 2002.

[3] Goldreich O., Karloff H., Schulman L., Trevisan L. Lower bounds for linear locally decodable codes and private information retrieval systems // Proc. of the 17th IEEE Conf. on Complexity Theory. IEEE Computer Society Press, 2002.

[4] Chor B., Goldreich O., Kushilevitz E., Sudan M. Private information retrieval / Journal version // J. of the ACM. 45: 965–981. 1998.

[5] Kerendis I., de Wolf R. Exponential lower bound for 2-query locally decodable codes // Proc. of the 35th ACM Sym. on Theory of Computing. P. 106–115. 2003.

[6] Beigel R., Fortnow L., Gasarch W. A nearly tight lower bound for private information retrieval protocols. Technical Report TR03–087. Electronic Colloquim on Computational Complexity (ECCC). 2003.

[7] Itoh T. On lower bounds for the communication complexity of private information retrieval. IEICE Trans. Fundamentals, ES40A(1). 2001.

[8] Yekhanin S. New Locally Decodable Codes and Private Information Retrieval Schemes. Electronic Colloquium on Computational Complexity (ECCC). TR06–127.

[9] Alexander A. Razborov, Sergey Yekhanin. An Omega(n1/3) Lower Bound for Bilinear Group Based Private Information Retrieval. FOCS 2006. 739–748

[10] Гасанов Э. Э., Майлыбаева Г. А. Доступ к базам данных без раскрытия запроса // Материалы конференции «Математика и безопасные информационные технологии». Москва, 23–24 октября 2003 г. 393–395.

[11] Майлыбаева Г. А. Границы вырожденности протоколов доступа к данным без раскрытия запроса // Дискретная математика. 2006. 18. № 2. 98–110.

[12] Майлыбаева Г. А. Порядок коммуникационной сложности для одного класса PIR-протоколов // Дискретная математика, в печати.