НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2025 Том 25 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
Jule–August 2025 Vol. 25 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

doi: 10.17586/2226-1494-2025-25-4-703-709
УДК 003.26

Протокол пересечения множеств с сохранением конфиденциальности

Иван Дмитриевич Иогансон✉

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
ivan.ioganson@yandex.ru✉, https://orcid.org/0000-0002-0856-2249

Аннотация

Введение. Протокол пересечения закрытых множеств (Private Set Intersection, PSI) является одним из фундаментальных примитивов конфиденциальных вычислений. Данный примитив позволяет нескольким сторонам, которые не доверяют друг другу, совместными усилиями вычислить пересечение их секретных множеств без разглашения при этом дополнительной информации об этих множествах. Это позволяет пользователям совместно проводить анализ данных без раскрытия конфиденциальной информации друг другу. **Метод.** В работе представлен новый протокол пересечения закрытых множеств для трех и более участников. Протокол работает в сети с топологией типа «кольцо». Это минимизирует количество необходимых каналов связи между пользователями. Протокол основан на идее условного разделения нуля, позволяющей использовать схему разделения секрета для определения принадлежности элемента множества всем пользователям. Для оценки быстродействия нового решения предложена программная реализация протокола на языке C++. **Основные результаты.** Показана безопасность разработанного протокола для трех и более пользователей при условии, что пользователи не будут сговариваться друг с другом для «Honest-But-Curious» модели злоумышленника. Предложенная модель подразумевает, что злоумышленником является один из участников протокола, который корректно выполняет протокол, но может анализировать полученную в ходе этого информацию для извлечения выгоды. Безопасность протокола основывается только на предположении о недостатке у злоумышленника информации для получения каких-либо полезных данных из полученных во время выполнения протокола сообщений. Таким образом, этот протокол обладает информационно-теоретической стойкостью. **Обсуждение.** Представленный протокол может быть использован для конфиденциального анализа данных, например, при обмене несколькими компаниями информацией об общих клиентах. Протокол позволяет трем пользователям найти пересечение множеств размера 10^6 примерно за 42 с. В настоящей реализации существует возможность добавления многопоточности или перенос вычислений больших матриц с процессора на графический ускоритель.

Ключевые слова

конфиденциальные вычисления, криптографический протокол, пересечение закрытых множеств, разделение секрета

Благодарности

Работа выполнена в рамках государственного задания (проект № FSER-2025-0003).

Ссылка для цитирования: Иогансон И.Д. Протокол пересечения множеств с сохранением конфиденциальности // Научно-технический вестник информационных технологий, механики и оптики. 2025. Т. 25, № 4. С. 703–709. doi: 10.17586/2226-1494-2025-25-4-703-709

Set intersection protocol with privacy preservation

Ivan D. Ioganson✉

ITMO University, Saint Petersburg, 197101, Russian Federation
ivan.ioganson@yandex.ru✉, https://orcid.org/0000-0002-0856-2249

Abstract

The Private Set Intersection Protocol (PSI) is one of the fundamental primitives of secure multi-party computations. This primitive allows several parties who do not trust each other to work together to calculate the intersection of their secret

sets without disclosing additional information about these sets. This allows users to jointly analyze data without revealing confidential information to each other. This paper describes a new protocol for the intersection of private sets for 3 or more participants. The protocol works in a network with a “ring” type topology which minimizes the number of necessary communication channels between users. The protocol is based on the idea of conditional zero-sharing which allows using a secret sharing scheme to determine whether an element of the set belongs to all users or not. To evaluate the performance of the proposed solution, a software implementation of the protocol in C++ is proposed. The security of the developed protocol for three or more users is shown, provided that users do not conspire with each other for an “Honest-But-Curious” attacker model. Proposed model implies that the attacker is one of the protocol participants who performs the protocol correctly, but can analyse the information obtained during this process to gain benefits. The security of the protocol is based only on the assumption that the attacker lacks information to obtain any useful data from the messages received during the protocol execution. Thus, this protocol is information-theoretically secure. The presented protocol can be used for confidential data analysis, for example, when several companies exchange information about common customers. The protocol allows three users to find the intersection of sets of sizes 10^6 in about 42 s. In the present implementation, it is possible to add multithreading or transfer large matrix calculations from the processor to the GPU.

Keywords
secure multi-party computations, cryptographic protocol, private set intersection, secret sharing

Acknowledgements
The research was carried out within the State Assignment (project No. FSER-2025-0003).

For citation: Ioganson I.D. Set intersection protocol with privacy preservation. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2025, vol. 25, no. 4, pp. 703–709 (in Russian). doi: 10.17586/2226-1494-2025-25-4-703-709

Введение

На сегодняшний день широкое распространение находят протоколы конфиденциальных вычислений, которые позволяют производить вычисления между несколькими сторонами, не раскрывая при этом их входные данные друг другу [1]. Одним из таких протоколов является протокол пересечения закрытых множеств (Private Set Intersection, PSI) [2–6], и он позволяет группе пользователей, у каждого из которых есть некий набор элементов, найти пересечение их наборов, не раскрывая при этом никакой дополнительной информации. Вариациями подобного протокола можно считать, например, протокол порогового PSI [7] или протокол нахождения мощности PSI [8].

Протоколы PSI находят широкое применение в различных областях [9]. Использование подобных протоколов позволяет повысить уровень конфиденциальности данных, что может быть важно при работе с чувствительной информацией. Одним из примеров может служить обмен конфиденциальными данными между организациями [10]. В таком случае обеспечение защиты информации при передаче будет иметь первостепенное значение, и протокол PSI может помочь предотвратить разглашение дополнительной информации. Другим примером применения может служить использование протоколов PSI в сфере медицины, в частности генетике. С развитием исследований в области генетики человека стало возможным записывать геном человека в цифровом виде, что в свою очередь порождает угрозу безопасности этих данных. Протокол PSI позволяет безопасно проводить такие операции, как тестирование на отцовство и родословную без раскрытия какой-либо дополнительной геномной информации о человеке [11, 12].

Существуют различные подходы к реализации протокола PSI. Их можно разделить на использующие и не использующие третью доверенную сторону. Реализации с использованием третьей доверенной стороны подразумевают наличие сервера, которому участники протокола посылают свои секретные множества, чтобы он вычислил их пересечение и выдал ответ. Такой подход наиболее эффективен с точки зрения производительности, однако требует большого допущения с точки зрения безопасности, а именно того, что сам сервер не скомпрометирован и может сохранить состояние конфиденциальности переданной ему информации. Исходя из этого, в данной работе будут рассмотрены подходы, не требующие третьей доверенной стороны.

Наивным подходом к реализации протокола PSI без третьей доверенной стороны является вычисление некой криптографической хеш-функции от каждого из элементов секретных множеств участников и последующее сравнение полученных значений. Данный подход позволяет скрыть секретные значения благодаря использованию односторонней хеш-функции [13]. Несмотря на то, что свойства хеш-функции не позволяют однозначно восстановить входное значение по выходному в общем случае, часто элементы секретных множеств пользователей в протоколе PSI не являются полностью случайными, а берутся из определенного и вполне конечного пространства допустимых значений (например: номера телефонов, имена людей, почтовые индексы). Знание того, из какого пространства значений взяты элементы множества может позволить злоумышленнику восстановить элементы множеств из выходных значений хеш-функции и привести к тому, что данный подход не будет безопасен.

Другой подход к построению протокола PSI основан на примитиве, называемом забывчивая псевдослучайная функция (Oblivious Pseudorandom Function, OPRF) [14–17]. OPRF — протокол, который позволяет получателю узнать значение некой псевдослучайной функции, известной отправителю, причем так, чтобы отправитель не узнал ни входных данных получателя, ни итогового значения. Данный примитив, в свою очередь, основывается на так называемом протоколе забывчивой передачи [18, 19], который позволяет получателю узнать одно и только одно из нескольких сообщений, отправленных ему отправителем, причем так, что от-

правитель не узнает, какое из сообщений было принято получателем.

Целью данной работы является снижение объема передаваемой в процессе работы протокола информации. Представлен анализ существующих решений. Разработан протокол PSI. Исследованы быстродействие, затраты на коммуникацию и безопасность разработанного протокола. Выполнено сравнение предложенного решения с аналогами.

Дано описание нового протокола PSI для трех и более участников, основанного на идеях схем разделения секрета. Основной особенностью данного решения является использование топологии сети «Кольцо». Это позволяет снизить нагрузку на отдельного участника. В предложенном решении каждый участник отправляет и принимает в процессе работы протокола всего по два сообщения. В работе также представлен метод выбора параметров для протокола. Для оценки быстродействия предложенного решения была разработана программная реализация протокола на языке C++.

Описание предложенного метода

Обозначения, используемые в работе. Идея представленного протокола основана на концепции условного разделения нуля, предложенного в работе [20]. Пусть $\mathbf{M}$ — матрица $n \times w$, состоящая из битовых строк длины m. Пусть $\mathbf{M}[x][y]$ обозначает элемент матрицы $\mathbf{M}$ на пересечении x-го столбца и y-ой строки. Каждый пользователь P_i для $i \in \{1, \ldots, t\}$, где t — количество участвующих в протоколе пользователей, имеет множество $\mathbf{X}_i$ длины u_i.

В протоколе используется простейшая схема разделения секрета для распределения нулевой матрицы на доли. Пусть $\mathbf{0}^{n\times w}$ обозначает нулевую матрицу размерности $n \times w$. Тогда нулевую матрицу можно разделить на t долей $\mathbf{Z}_i$, где $i \in \{1, \ldots, t\}$, таким образом, что:

$$\mathbf{0}^{n\times w} = \mathbf{Z}_1 \oplus \mathbf{Z}_2 \oplus \ldots \oplus \mathbf{Z}_t,$$

где символ $\oplus$ — поэлементная побитовая операция исключающие «ИЛИ» матриц.

Также в протоколе используется криптографическая хеш-функция H_1: $\{0, 1\}^* \to \{1, \ldots, n\}^w$.

Описание протокола. Пользователь P_1 инициирует протокол, и в конце получает выходное значение $\mathbf{I}$ — пересечение всех множеств. Участники предварительно договариваются о параметрах m, n и w.

Создание протокола можно условно разделить на три этапа.

Этап 1. **Генерация матрицы нулей.**

Шаг 1.1. Пользователь P_1 генерирует случайную матрицу $\mathbf{M}_1 \xleftarrow{\$} \{\{0, 1\}^m\}^{n\times w}$.

Шаг 1.2. Каждый пользователь P_i для $i \in \{2, \ldots, t\}$:

1.2.1. получает от пользователя P_{i-1} матрицу $\mathbf{M}_{i-1}$;

1.2.2. генерирует случайную матрицу $\mathbf{Z}_i \xleftarrow{\$} \{\{0, 1\}^m\}^{n\times w}$;

1.2.3. отправляет пользователю P_{i+1} матрицу $\mathbf{M}_i = \mathbf{M}_{i-1} \oplus \mathbf{Z}_i$.

Шаг 1.3. Пользователь P_1 получает от пользователя P_t матрицу $\mathbf{M}_t$.

Шаг 1.4. Пользователь P_1 вычисляет $\mathbf{Z}_1 = \mathbf{M}_t \oplus \mathbf{M}_1$.

Этап 2. **Сбор информации о множествах.**

Каждый пользователь P_i для $i \in \{1, \ldots, t\}$:

Шаг 2.1. Если $i = 1$, то генерирует случайную матрицу $\mathbf{B}_0 \xleftarrow{\$} \{\{0, 1\}^m\}^{n\times w}$. Иначе, получает от пользователя P_{i-1} матрицу $\mathbf{B}_{i-1}$.

Шаг 2.2. Генерирует случайную матрицу $\mathbf{A}_i \xleftarrow{\$} \{\{0, 1\}^m\}^{n\times w}$.

Шаг 2.3. Для каждого x_{ij} из множества $\mathbf{X}_i$ вычисляет $\mathbf{v}_j = H_1(x_{ij}) \in \{1, \ldots, n\}^w$ для $j \in \{1, \ldots, u_i\}$.

Шаг 2.4. Для каждого $\mathbf{v}_j$ обновляет значения $\mathbf{A}_i[l][\mathbf{v}_j[l]] = \mathbf{Z}_i[l][\mathbf{v}_j[l]]$ для $l \in \{1, \ldots, w\}$.

Если $i < t$, то вычисляет матрицу $\mathbf{B}_i = \mathbf{B}_{i-1} \oplus \mathbf{A}_i$ и отправляет ее пользователю P_{i+1}. Иначе, вычисляет матрицу $\mathbf{B}_t = \mathbf{B}_{t-1} \oplus \mathbf{A}_t$ и отправляет ее пользователю P_1.

Этап 3. **Поиск пересечения.**

Шаг 3.1. Пользователь P_1 имеет матрицы $\mathbf{B}_0$ и $\mathbf{B}_t$.

Шаг 3.2. Пользователь P_1 вычисляет $\mathbf{C} = \mathbf{B}_0 \oplus \mathbf{B}_t$.

Шаг 3.3. Пусть $\mathbf{I} = \varnothing$.

Шаг 3.4. Пользователь P_1 для каждого элемента x_{1j}, где $j \in \{1, \ldots, u_1\}$, из множества $\mathbf{X}_1$:

3.4.1. вычисляет $\mathbf{v}_j^{(1)} = H_1(x_{1j}) \in \{1, \ldots n\}^w$;

3.4.2. проверяет выполняется ли равенство:

$$\mathbf{C}[1]\left[\mathbf{v}_j^{(1)}[1]\right] = \mathbf{C}[2]\left[\mathbf{v}_j^{(1)}[2]\right] = \cdots = \mathbf{C}[w]\left[\mathbf{v}_j^{(1)}[w]\right] = 0;$$

3.4.3. если равенство выполняется, то $\mathbf{I} = \mathbf{I} \cup \{x_1 j\}$.

Пример работы протокола. Рассмотрим наглядный пример работы протокола для трех пользователей.

Пусть входные множества пользователей будут три случайные строки ASCII символов:

$$\mathbf{X}_1 = \{\text{«nH/»}, \text{«0Kn»}, \text{«–W9»}\},$$
$$\mathbf{X}_2 = \{\text{«0Kn»}, \text{«sNd»}, \text{«FKh»}\},$$
$$\mathbf{X}_3 = \{\text{«S2P»}, \text{«iDt»}, \text{«0Kn»}\}.$$

Выберем в качестве параметров протокола значения $(m, n, w) = (8, 4, 8)$. Тогда соответственными значениями хеш-функции H_1: $\{0, 1\}^* \to \{1, \ldots, n\}^w$ для этих строк будут:

$$H_1(\mathbf{X}_1) = \{(0, 1, 3, 3, 3, 2, 3, 1), (1, 0, 0, 2, 1, 3, 3, 1), (2, 2, 0, 1, 3, 1, 3, 0)\},$$
$$H_1(\mathbf{X}_2) = \{(1, 0, 0, 2, 1, 3, 3, 1), (0, 3, 2, 1, 3, 0, 2, 1), (2, 2, 1, 3, 3, 2, 1, 3)\},$$
$$H_1(\mathbf{X}_3) = \{(0, 2, 1, 2, 3, 1, 0, 1), (2, 2, 0, 0, 3, 1, 2, 1), (1, 0, 0, 2, 1, 3, 3, 1)\}.$$

На этапе 1 создания протокола генерируются доли нулевой матрицы:

$$\mathbf{Z}_1 = \begin{pmatrix} 96 & d7 & 78 & 96 & f7 & 50 & b2 & d8 \\ 0a & ec & ee & 84 & 19 & d0 & a1 & b1 \\ 01 & 01 & 0a & 5e & e5 & a6 & 86 & 8e \\ 06 & 0e & 46 & 6e & e6 & 59 & 3b & 44 \end{pmatrix},$$

$$\mathbf{Z}_2 = \begin{pmatrix} 8b & f6 & d1 & 5c & 1a & b3 & a0 & 1f \\ 84 & 9e & a0 & 16 & f4 & 88 & 73 & 2e \\ 38 & 58 & 6f & 36 & 72 & f5 & f1 & e3 \\ 11 & 67 & f3 & 17 & 00 & 39 & e1 & 62 \end{pmatrix},$$

$$\mathbf{Z}_3 = \begin{pmatrix} 1d & 21 & a9 & ca & ed & e3 & 12 & c7 \\ 8e & 72 & 4e & 92 & ed & 58 & d2 & 9f \\ 39 & 59 & 65 & 68 & 97 & 53 & 77 & 6d \\ 17 & 69 & b5 & 79 & e6 & 60 & da & 26 \end{pmatrix}.$$

Элементы матрицы представляют из себя в данном случае 8-битные строки, записанные в шестнадцатеричном виде. Можно заметить, что выполняется равенство $\mathbf{Z}_1 \oplus \mathbf{Z}_2 \oplus \mathbf{Z}_3 = \mathbf{0}^{n \times w}$.

На этапе 2 — пользователи генерируют матрицы $\mathbf{A}_1$, $\mathbf{A}_2$ и $\mathbf{A}_3$ на основе своих входных множеств и матриц $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{Z}_3$:

$$\mathbf{A}_1 = \begin{pmatrix} 96 & d7 & 78 & 96 & f7 & 50 & b2 & d8 \\ 0a & ec & 24 & 84 & 19 & d0 & 11 & b1 \\ 01 & 01 & 0a & 5e & d4 & a6 & a3 & e6 \\ d6 & 71 & 46 & 6e & e6 & 59 & 3b & f8 \end{pmatrix},$$

$$\mathbf{A}_2 = \begin{pmatrix} 8b & f6 & d1 & bc & eb & b3 & fd & 81 \\ 84 & ab & a0 & 16 & f4 & 2f & 73 & 2e \\ 38 & 58 & 6f & 36 & be & f5 & f1 & e4 \\ 1f & 67 & b5 & 17 & 00 & 39 & e1 & 62 \end{pmatrix},$$

$$\mathbf{A}_3 = \begin{pmatrix} 1d & 21 & a9 & ca & 1a & 85 & 12 & 6f \\ 8e & ee & 4e & 21 & ed & 58 & 06 & 9f \\ 39 & 59 & 65 & 68 & 0e & 97 & 77 & c4 \\ 25 & 09 & 78 & 48 & e6 & 60 & da & b7 \end{pmatrix}.$$

На этапе 3 — пользователь P_1 получает матрицу $\mathbf{C}$:

$$\mathbf{C} = \begin{pmatrix} 00 & 00 & 00 & 04 & 1f & c0 & 77 & 36 \\ 00 & a9 & ca & b3 & 00 & a7 & 64 & 00 \\ 00 & 00 & 3c & 00 & 64 & c4 & 25 & c6 \\ 8c & 1f & 8b & 31 & 00 & 00 & 00 & 2d \end{pmatrix}.$$

Для матрицы $\mathbf{C}$ выполняется равенство $\mathbf{C} = \mathbf{A}_1 \oplus \oplus \mathbf{A}_2 \oplus \mathbf{A}_3$. Очевидно, что пересечение, полученное в результате работы протокола, будет $\mathbf{I} = \{\text{«0Kn»}\}$.

Исследование предложенного метода

Безопасность протокола. Разработанный протокол безопасен для «Honest-But-Curious» модели злоумышленника при дополнительном условии, что пользователи не будут сговариваться между собой, и количестве участников не менее трех. Подобная модель предполагает, что злоумышленник, являясь одним из участников протокола, честно следует самому протоколу, однако также может анализировать полученную в ходе протокола информацию, для получения интересующих его сведений. Это означает, что пока участники соблюдают протокол, они не узнают ничего о секретных множествах друг друга, кроме результата работы протокола.

Протокол обладает информационно-теоретической стойкостью. Это означает, что его безопасность основана только на предположении, что злоумышленник не обладает достаточной информацией для взлома. В отличии от вычислительно стойких криптосистем, безопасность которых основывается на предположении о вычислительной сложности определенной математической задачи (например, факторизация числа или вычисление дискретного логарифма), информационно-теоретически стойкие системы невозможно взломать даже имея неограниченные вычислительные мощности.

Для обоснования надежности разработанного протокола обратимся к этапам создания протокола. На этапе 2 каждый пользователь P_i получает от пользователя P_{i-1} матрицу $\mathbf{B}_{i-1}$, формирует на основе своего секретного множества матрицу $\mathbf{A}_i$ и отправляет пользователю P_i матрицу $\mathbf{B}_i = \mathbf{B}_{i-1} \oplus \mathbf{A}_i$. По факту именно матрица $\mathbf{A}_i$ содержит информацию о множестве пользователя P_i. В данной матрице часть элементов случайно сгенерирована, а часть взята из матрицы $\mathbf{Z}_i$, сгенерированной на этапе 1. Однако элементы матрицы $\mathbf{Z}_i$ не отличимы от случайных. Потому, не имея информации об элементах матрицы $\mathbf{Z}_i$, невозможно отличить те элементы, что были сгенерированы на этапе 2, от тех, что были взяты из матрицы $\mathbf{Z}_i$. Если принять предположение о том, что пользователи не могут сговариваться между собой, то ни один пользователь не может получить информацию об элементах матрицы $\mathbf{Z}_i$ другого пользователя на этапе 1.

Единственным исключением будет случай, когда количество пользователей равно двум. В этом случае, так как известно, что $\mathbf{Z}_1 \oplus \mathbf{Z}_2 = \mathbf{0}^{n \times w}$, каждый пользователь, зная свою долю нулевой матрицы, может вычислить долю, принадлежащую другому. В связи с этим для данного случая протокол небезопасен.

Выбор параметров протокола. Быстродействие протокола напрямую зависит от выбранных параметров (m, n, w), а также от количества участников t и количества элементов множеств каждого из пользователей. Для упрощения предположим, что количество элементов входного множества у каждого из участников равно u, а количество элементов пересечения множеств пользователей равно q. Параметры протокола выбираются в зависимости значения $R(m, n, w, t, u, q)$ — вероятности того, что в результате работы протокол выдаст некорректный результат. Это может произойти, если на шаге 3.4.2 этапа 3 протокола равенство будет выполнено для элемента, не принадлежащего пересечению.

Чтобы вычислить значение $R(m, n, w, t, u, q)$, для начала следует оценить вероятность того, что элемент матрицы $\mathbf{C}$, полученной на шаге 3.2 этапа 3 протокола, находящийся на пересечении x-го столбца и y-ой строки, равен нулю. Рассмотрим некий элемент s, принадлежащий входному множеству одного из пользователей, и соответствующее ему значение $\mathbf{v} = H_1(s) \in \{1, \ldots, n\}^w$ — значение хеш-функции для этого элемента s. Тогда есть три случая, когда $\mathbf{C}[x][y]$ будет равен 0.

Случай 1. Если $y = \mathbf{v}[x]$ для хотя бы одного значения $\mathbf{v}$ соответствующего элементу s из пересечения.

Случай 2. Если случай 1 не выполняется и для каждого участника существует хотя бы один элемент s и соответствующее ему значение $\mathbf{v}$, такое, что $y = \mathbf{v}[x]$.

Случай 3. Если случаи 1 и 2 не выполняются, то $\mathbf{C}[x][y]$ — случайный элемент из $\{0, 1\}^m$ и может так получиться, что он равен 0.

Соответствующие вероятности для каждого из этих случаев и общая вероятность:

$$p_1 = 1 - \left(1 - \frac{1}{n}\right)^q;$$

$$p_2 = \left(1 - \left(1 - \frac{1}{n}\right)^{u-q}\right)^{t-1};$$

$$p_3 = 2^{-m};$$

$$Prob(\mathbf{C}[x][y] = 0) = p_1 + (1 - p_1)(p_2 + (1 - p_2)p_3).$$

Тогда $R(m, n, w, t, u, q)$ будет:

$$R(m, n, w, t, u, q) = 1 - (1 - Prob(\mathbf{C}[x][y] = 0)^w)^{u-q}. \quad (1)$$

Формула (1) позволяет определить примерную вероятность ошибки для заданных значений m, n, w, t, u, q. Основной недостаток состоит в том, что изначально не может быть известно значение q, так как оно станет известно только в конце работы протокола. Следовательно, для определения параметров (m, n, w) для заданных значений t и u, а также желаемой вероятности ошибки p, следует использовать формулу:

$$\max_{q \in \{0 \ldots u\}} R(m, n, w, t, u, q) \le p\ . \quad (2)$$

Обсуждение

Для оценки быстродействия представленного протокола была реализована программа на языке C++. Программа симулировала выполнение протокола на локальной машине в одном потоке, используя процессор AMD Ryzen 7 5800H. Таким образом, было замерено среднее время работы протокола для разных параметров. Отметим, что полученное среднее время не учитывает передачу данных по сети. Всего в процессе работы протокола по сети отправляется $2 \times t \times m \times n \times w$ бит информации за $2 \times t$ раундов коммуникации.

В табл. 1 приведены время работы протокола и количество переданной информации для разного количества участвующих сторон t и размеров входных данных u. Для заданной вероятности ошибки p были выбраны подходящие параметры (m, n, w) на основании формулы (2).

В табл. 2 приведено сравнение предложенного решения с аналогами. В качестве параметров для сравнения взяты объем трафика на пользователя и количество раундов коммуникации приведены асимптотические приближения от параметров u, t и к, где u и t были определены по табл. 1, а к — статистическая безопасность, зависящая от параметра p. Статистическая безопасность связана с вероятностью ошибки p соотношением $\text{к} = -\log_2 p$.

Из табл. 2 видно, что только предложенное решение требует топологии сети «Кольцо», для которой необходимо меньшее количество активных соединений, чем для полносвязной топологии, поэтому ее проще настраивать. Также некоторые протоколы запрашивают константного числа раундов коммуникации, но это увеличивает количество трафика, приходящегося на пользователя. В работах [5, 6, 15] количество передаваемой информации зависит от количества участников протокола, а в предложенном решении объем переданной информации зависит только от размера входных данных и параметра статистической безопасности.

Отметим, что только в работе [20] объем трафика на пользователя соизмерим с тем же объемом в пред-

Таблица 1. Производительность протокола

Table 1. Protocol performance

t	u	(m, n, w)	p	Количество переданной информации, МБ	Время работы протокола, с
3	10^3	(8, 2000, 20)	10^{-6}	0,234	0,027
3	10^4	(8, 20 000, 25)	10^{-6}	2,860	0,340
3	10^5	(8, 200 000, 25)	10^{-6}	28,600	2,800
3	10^6	(8, 1 000 000, 50)	10^{-6}	286,100	42,400
4	10^4	(8, 20 000, 25)	10^{-6}	3,810	0,330
4	10^5	(8, 200 000, 25)	10^{-6}	38,100	3,700
5	10^5	(8, 200 000, 25)	10^{-6}	47,700	4,800
8	10^5	(8, 200 000, 25)	10^{-6}	76,300	6,900
10	10^5	(8, 200 000, 25)	10^{-6}	95,400	8,400

Таблица 2. Сравнение предложенного протокола с аналогами

Table 2. Comparison with analogues

Протокол	Объем трафика на пользователя	Количество раундов коммуникации	Топология сети	Наличие доверенной стороны
[15]	$O(u \times t \times \text{к})$	$O(1)$	Полносвязная	Да
[6]	$O(u \times t^2 \times \text{к})$	$O(1)$	Полносвязная	Нет
[20]	$O(u \times \text{к})$	$O(t)$	Полносвязная	Нет
[5]	$O(u \times t \times \text{к})$	$O(1)$	Полносвязная	Нет
Предложенное решение	$O(u \times \text{к})$	$O(t)$	Кольцо	Нет

ложенном решении. Также в [20] при размере входных множеств равном 2^{20} и вероятности ошибки — 2^{-40} коммуникационные затраты на пользователя составляют 447,44 и 935,32 МБ для двух вариантов протокола. В описанном в настоящей работе протоколе каждый пользователь передает и получает суммарно $4 \times m \times n \times w$ бит информации, что при таком же размере входных данных и вероятности ошибки будет соответствовать параметрам $(m, n, w) = (8, 2^{20}, 80)$ и равно 320 МБ информации, что в 1,4 и 2,9 раз меньше, чем коммуникационные затраты для протоколов, описанных в [20].

Заключение

В работе описан новый протокол поиска пересечения множеств с сохранением конфиденциальности, основанный на идее условного разделения нуля, предложенного в [20]. Данный протокол обладает информационно-теоретической стойкостью для трех и более пользователей, при условии, что пользователи не будут сговариваться друг с другом. Основной отличительной особенностью данного протокола является тот факт, что для своей работы он требует топологии сети «Кольцо», что позволяет снизить количество активных соединений между участниками. Также, в отличии от многих других подобных протоколов, объем трафика, передаваемого одним пользователем, не зависит от общего числа пользователей. Разработана программная реализация протокола на языке C++ для оценки быстродействия предложенного решения. Протокол позволяет находить пересечение множеств размера 10^6 для трех пользователей примерно за 42 с, что говорит о его практическом применении. Кроме того, время работы протокола, может быть в дальнейшем улучшено путем распараллеливания вычислений.

Литература

1. Evans D., Kolesnikov V., Rosulek M. A pragmatic introduction to secure multi-party computation // Foundations and Trends in Privacy and Security. 2018. V. 2. N 2-3. P. 70–246. http://dx.doi.org/10.1561/3300000019
2. Falzon F., Markatou E.A. Re-visiting Authorized Private Set Intersection: a new privacy-preserving variant and two protocols // Proceedings on Privacy Enhancing Technologies. 2025. V. 1. P. 792–807. https://doi.org/10.56553/popets-2025-0041
3. Sun Z. Efficient multiparty private set intersection protocol based on function secret sharing // Proceedings of SPIE. 2024. V. 13175. P. 1317505. https://doi.org/10.1117/12.3031902
4. Debnath S.K. Provably Secure Private Set Intersection With Constant Communication Complexity // International Journal of Cyber Warfare and Terrorism. 2019. V. 9. N 2. P. 39–64. https://doi.org/10.4018/IJCWT.2019040104
5. Ben-Efraim A., Nissenbaum O., Omri E., Paskin-Cherniavsky A. Psimple: Practical multiparty maliciously-secure private set intersection // Proc. of the ACM on Asia Conference on Computer and Communications Security. 2022. P. 1098–1112. https://doi.org/10.1145/3488932.3523254
6. Cheon J.H., Jarecki S., Seo J.H. Multi-party privacy-preserving set intersection with quasi-linear complexity // IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2012. V. E95.A. N 8. P. 1366–1378. https://doi.org/10.1587/transfun.e95.a.1366
7. Bay A. New threshold private set intersection protocols // Mugla Journal of Science and Technology. 2024. V. 10. N 1. P. 51–60. https://doi.org/10.22531/muglajsci.1387499
8. Gao J., Trieu N., Yanai A. Multiparty private set intersection cardinality and its applications // Proceedings on Privacy Enhancing Technologies. 2024. V. 2024. N 2. P. 73–90. https://doi.org/10.56553/popets-2024-0041
9. Faber S. Variants of privacy preserving set intersection and their practical applications. Dissertation for the degree of PhD in Computer Science. Irvine, University of California, 2016. 192 p.
10. LiY., Jiang Z.L., Yao L., Wang X., Yiu S.M., Huang Z. Outsourced privacy-preserving C4.5 decision tree algorithm over horizontally and vertically partitioned dataset among multiple parties // Cluster Computing. 2019. V. 22. Suppl. 1. P. 1581–1593. https://doi.org/10.1007/s10586-017-1019-9
11. Shen L., Chen X., Wang D., Fang B., Dong Y. Efficient and private set intersection of human genomes // Proc. of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2018. P. 761–764. https://doi.org/10.1109/bibm.2018.8621291
12. Aziz M.M.A., Alhadidi D., Mohammed N. Secure approximation of edit distance on genomic data // BMC Medical Genomics. 2017. V. 10. P. 55–67. https://doi.org/10.1186/s12920-017-0279-9

References

1. Evans D., Kolesnikov V., Rosulek M. A pragmatic introduction to secure multi-party computation. *Foundations and Trends in Privacy and Security*, 2018, vol. 2, no. 2-3, pp. 70–246. http://dx.doi.org/10.1561/3300000019
2. Falzon F., Markatou E.A. Re-visiting Authorized Private Set Intersection: a new privacy-preserving variant and two protocols. *Proceedings on Privacy Enhancing Technologies*, 2025, vol. 1, pp. 792–807. https://doi.org/10.56553/popets-2025-0041
3. Sun Z. Efficient multiparty private set intersection protocol based on function secret sharing. *Proceedings of SPIE*, 2024, vol. 13175, pp. 1317505. https://doi.org/10.1117/12.3031902
4. Debnath S.K. Provably Secure Private Set Intersection With Constant Communication Complexity. *International Journal of Cyber Warfare and Terrorism*, 2019, vol. 9, no. 2, pp. 39–64. https://doi.org/10.4018/IJCWT.2019040104
5. Ben-Efraim A., Nissenbaum O., Omri E., Paskin-Cherniavsky A. Psimple: Practical multiparty maliciously-secure private set intersection. *Proc. of the ACM on Asia Conference on Computer and Communications Security*, 2022, pp. 1098–1112. https://doi.org/10.1145/3488932.3523254
6. Cheon J.H., Jarecki S., Seo J.H. Multi-party privacy-preserving set intersection with quasi-linear complexity. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2012, vol. E95.A, no. 8, pp. 1366–1378. https://doi.org/10.1587/transfun.e95.a.1366
7. Bay A. New threshold private set intersection protocols. *Mugla Journal of Science and Technology*, 2024, vol. 10, no. 1, pp. 51–60. https://doi.org/10.22531/muglajsci.1387499
8. Gao J., Trieu N., Yanai A. Multiparty private set intersection cardinality and its applications. *Proceedings on Privacy Enhancing Technologies*, 2024, vol. 2024, no. 2, pp. 73–90. https://doi.org/10.56553/popets-2024-0041
9. Faber S. Variants of privacy preserving set intersection and their practical applications. *Dissertation for the degree of PhD in Computer Science*. Irvine, University of California, 2016, 192 p.
10. LiY., Jiang Z.L., Yao L., Wang X., Yiu S.M., Huang Z. Outsourced privacy-preserving C4.5 decision tree algorithm over horizontally and vertically partitioned dataset among multiple parties. *Cluster Computing*, 2019, vol. 22, suppl. 1, pp. 1581–1593. https://doi.org/10.1007/s10586-017-1019-9
11. Shen L., Chen X., Wang D., Fang B., Dong Y. Efficient and private set intersection of human genomes. *Proc. of the IEEE International Conference on Bioinformatics and Biomedicine* (BIBM), 2018, pp. 761–764. https://doi.org/10.1109/bibm.2018.8621291
12. Aziz M.M.A., Alhadidi D., Mohammed N. Secure approximation of edit distance on genomic data. *BMC Medical Genomics*, 2017, vol. 10, pp. 55–67. https://doi.org/10.1186/s12920-017-0279-9

13. Hasan H.A., Al-Layla H.F., Ibraheem F.N. A review of hash function types and their applications // Wasit Journal of Computer and Mathematics Science. 2022. V. 1. N 3. P. 75–88. https://doi.org/10.31185/wjcm.52
14. Casacuberta S., Hesse J., Lehmann A. SoK: oblivious pseudorandom functions // Proc. of the IEEE 7th European Symposium on Security and Privacy (EuroS&P). 2022. P. 625–646. https://doi.org/10.1109/eurosp53844.2022.00045
15. Bay A., Kayan A. A new multi-party private set intersection protocol based on OPRFs // Mugla Journal of Science and Technology. 2022. V. 8. N 1. P. 69–75. https://doi.org/10.22531/muglajsci.1075788
16. Chase M., Miao P. Private set intersection in the internet setting from lightweight oblivious PRF // Lecture Notes in Computer Science. 2020. V. 12172. P. 34–63. https://doi.org/10.1007/978-3-030-56877-1_2
17. Kavousi A., Mohajeri J., Salmasizadeh M. Efficient scalable multi-party private set intersection using oblivious PRF // Lecture Notes in Computer Science. 2021. V. 13075. P. 81–99. https://doi.org/10.1007/978-3-030-91859-0_5
18. Pinkas B., Schneider T., Zohner M. Faster private set intersection based on OT extension // Proc. of the 23rd Usenix Security Symposium. 2014. P. 797–812.
19. Tang Y., Guo M., Huo Y., Zhao Z., Yu J., Qin B. An MLWE-based cut-and-choose oblivious transfer protocol // Entropy. 2024. V. 26. N 9. P. 793. https://doi.org/10.3390/e26090793
20. Kolesnikov V., Matania N., Pinkas B., Rosulek M., Trieu N. Practical multi-party private set intersection from symmetric-key techniques // Proc. of the ACM SIGSAC Conference on Computer and Communications Security. 2017. P. 1257–1272. https://doi.org/10.1145/3133956.3134065

13. Hasan H.A., Al-Layla H.F., Ibraheem F.N. A review of hash function types and their applications. *Wasit Journal of Computer and Mathematics Science*, 2022, vol. 1, no. 3, pp. 75–88. https://doi.org/10.31185/wjcm.52
14. Casacuberta S., Hesse J., Lehmann A. SoK: oblivious pseudorandom functions. *Proc. of the IEEE 7th European Symposium on Security and Privacy* (EuroS&P), 2022, pp. 625–646. https://doi.org/10.1109/eurosp53844.2022.00045
15. Bay A., Kayan A. A new multi-party private set intersection protocol based on OPRFs. *Mugla Journal of Science and Technology*, 2022, vol. 8, no. 1, pp. 69–75. https://doi.org/10.22531/muglajsci.1075788
16. Chase M., Miao P. Private set intersection in the internet setting from lightweight oblivious PRF. *Lecture Notes in Computer Science*, 2020, vol. 12172, pp. 34–63. https://doi.org/10.1007/978-3-030-56877-1_2
17. Kavousi A., Mohajeri J., Salmasizadeh M. Efficient scalable multi-party private set intersection using oblivious PRF. *Lecture Notes in Computer Science*, 2021, vol. 13075, pp. 81–99. https://doi.org/10.1007/978-3-030-91859-0_5
18. Pinkas B., Schneider T., Zohner M. Faster private set intersection based on OT extension. *Proc. of the 23rd Usenix Security Symposium*, 2014, pp. 797–812.
19. Tang Y., Guo M., Huo Y., Zhao Z., Yu J., Qin B. An MLWE-based cut-and-choose oblivious transfer protocol. *Entropy*, 2024, vol. 26, no. 9, pp. 793. https://doi.org/10.3390/e26090793
20. Kolesnikov V., Matania N., Pinkas B., Rosulek M., Trieu N. Practical multi-party private set intersection from symmetric-key techniques. *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1257–1272. https://doi.org/10.1145/3133956.3134065

Автор

Иогансон Иван Дмитриевич — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 57883504800, https://orcid.org/0000-0002-0856-2249, ivan.ioganson@yandex.ru

Author

Ivan D. Ioganson — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 57883504800, https://orcid.org/0000-0002-0856-2249, ivan.ioganson@yandex.ru

Статья поступила в редакцию 06.02.2025
Одобрена после рецензирования 04.06.2025
Принята к печати 17.07.2025

Received 06.02.2025
Approved after reviewing 04.06.2025
Accepted 17.07.2025