Ссылка на статью:
// Ученые записки УлГУ. Серия Математика и информационные технологии. 2024, № 2, с. 61–70.
Поступила: 31.07.2024
Окончательный вариант: 20.10.2024

УДК 519.7

Протокол безопасных вычислений для четырех участников с активным противником

Рацеев С. М.[1,*], ***Череватенко О. И.***[2]

[*]ratseevsm@mail.ru
[1]УлГУ, Ульяновск, Россия
[2]УлГПУ им. И.Н. Ульянова, Ульяновск, Россия

Протоколы безопасных многосторонних вычислений позволяют группе участников, взаимодействуя между собой, совместно выполнять вычисления некоторой функциональности без раскрытия личных данных участников. Безопасные протоколы для случая пассивного противника гарантируют безопасность до тех пор, пока нечестные участники строго следуют инструкциям протокола. Безопасные протоколы для случая активного противника гарантируют безопасность даже если нечестные участники начнут отклоняться от инструкций протокола. В этом случае для достижения безопасности протокол требует большей вычислительной и коммуникационной сложности. Схемы разделения секрета играют важную роль в обеспечении конфиденциальности во время многосторонних вычислений. В 2020 г. авторы Dalskov A., Escudero D., Keller M. представили новый четырехсторонний протокол безопасных вычислений с честным большинством для случая активного противника. Этот протокол обладает эффективностью, сравнимой с аналогичными протоколами с теми же настройками, при этом имеет гораздо более простую конструкцию. Указанные авторы не приводят полного протокола для вычисления арифметических схем, показывая лишь идеи для этого протокола. В данной работе приводится полный протокол безопасных вычислений.

Ключевые слова: *криптографический протокол, многосторонние вычисления, схема разделения секрета*

Введение

Существуют два основных подхода к построению протоколов многосторонних вычислений [1, 6, 7]: протоколы на основе *схем разделения секрета*, которые работают путем взаимодействия участников для каждого элемента схемы; протоколы на основе *искаженных схем*, которые работают при помощи того, что участники создают искаженную версию схемы, которая может быть вычислена сразу. Оба подхода важны и имеют настройки, при которых один подход работает лучше, чем другой. С одной стороны, подход с искаженными схемами дает протоколы с постоянным числом раундов. Таким образом, в сетях с высокой задержкой они намного превосходят протоколы, основанные на схемах разделения секрета, у которых количество раундов линейно по глубине вычисляемой схемы. С другой стороны, протоколам, основанным на схемах разделения секрета, обычно достаточно низкой пропускной способности канала и они передают небольшие сообщения на каждый элемент схемы, в отличие от искаженных схем, которые передают большие объемы информации, и требуют большой (и дорогостоящей) пропускной способности.

В работе [4] представлен четырехсторонний протокол безопасных вычислений DEK (Dalskov A., Escudero D., Keller M.) с честным большинством для случая активного противника, который обладает эффективностью, сравнимой с аналогичными протоколами с теми же настройками, при этом имеет гораздо более простую конструкцию. Одна из вариаций представленного протокола удовлетворяет требованиям безопасности с прерыванием, но также предложены некоторые расширения для обеспечения гарантированного получения выходных данных.

В оригинальной работе [4] авторы не приводят полного протокола для вычисления арифметических схем, показывая лишь идеи для этого протокола. В данной работе приводится полный протокол безопасных вычислений, дополняющий работу [4]. Протоколы с честным большинством, защищенные от пассивного или активного противника, в последнее время привлекают большое внимание благодаря своей эффективности.

Рассмотрим безопасные вычисления с участием четырех участников. Четырехсторонний протокол основан на реплицированной схеме разделения секрета [2, 3]. В этом случае секрет x разделяется как (x_1, x_2, x_3, x_4) с условием $x = x_1 + x_2 + x_3 + x_4$, причем участник P_i обладает долей $\{x_j \mid j = 1, 2, 3, 4,\ j \neq i\}$. Заметим, что порог равен двум, поэтому доля одного участника не приводит к утечке информации об общем секрете x, но любые две доли вместе полностью определяют значение секрета. Напомним, что данная схема обладает свойством линейности. Поэтому элементы сложения и умножения на константу являются локальными операциями. Для операции умножения выполнено равенство $xy = \Big(\sum\limits_i x_i\Big)\Big(\sum\limits_j y_j\Big) = \sum\limits_{i,j} x_i y_j$. Это значит, учитывая свойство линейности, что $[xy] = \sum\limits_{i,j}[x_i y_j]$, где

$$[x] = \big((x_2, x_3, x_4),\ (x_1, x_3, x_4),\ (x_1, x_2, x_4),\ (x_1, x_2, x_3)\big)$$

— вектор долей. Рассмотрим слагаемые $x_i y_j$ и $x_j y_i$, $i \neq j$. У участника P_i нет x_i и y_i, а у всех остальных участников имеются эти значения. Аналогично, у участника P_j нет x_j и y_j,

а у всех остальных участников имеются эти значения. Поэтому значения $x_i y_j$ и $x_j y_i$ могут быть вычислены локально всеми участниками, за исключением участников P_i и P_j. Если записать $[xy] = \sum\limits_{1 \le i < j \le 4} [x_i y_j + x_j y_i] + \sum\limits_{i=1}^{4} [x_i y_i]$, то это значит, что каждый участник может получить часть рассматриваемой суммы. Проще говоря, частичная сумма $x_i y_j + x_j y_i$ будет составлять долю произведения xy и участникам затем просто нужно будет распределить свои частичные суммы таким образом, чтобы в итоге у каждого была новая доля произведения реплицированной схемы разделения секрета.

Для четырех участников защита от активного противника обеспечивается при помощи использования избыточности каждой доли секрета, которым владеют три участника (например, значением x_4 владеют участники P_1, P_2, P_3). В результате участники могут легко распределить долю произведения так, что противник не сможет вмешаться в процесс.

Протокол безопасных многосторонних вычислений должен обладать следующими основными свойствами:

– *корректность* (correctness): для любых входных значений $x_1, \ldots, x_n$ протокол возвращает $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$, при этом каждый (честный) участник P_i в итоге должен получить свое правильное выходное значение y_i, даже если некоторые из участников вычислений отклоняются от предписанных им действий;

– *конфиденциальность* (privacy): в результате вычислений ни один из участников не получает никакой дополнительной информации о секретах других участников, а именно о x_i и y_i.

Помимо этих основных свойств также выделяют следующие свойства:

– *независимость входных данных* (independence of inputs) — нечестные участники должны выбирать свои входные данные независимо от входных данных честных участников;

– *гарантированное получение выходных данных* (guaranteed output delivery) — нечестные участники не должны иметь возможности помешать честным участникам получать свои выходные значения;

– *справедливость* (fairness) — нечестные участники должны получить свои выходные значения тогда и только тогда, когда честные участники получат свои выходные значения.

Заметим, что приведенные свойства представляют собой не определение безопасности протокола, а скорее набор требований, которые должны соблюдаться для любого безопасного протокола. Также заметим, что достижение справедливости или гарантированного получения выходных данных обычно обходится дорого с точки зрения сложности протокола. Следовательно, на практике рассматривают различные типы протоколов: протоколы, которые не обеспечивают справедливость; протоколы, которые обеспечивают справедливость, но не гарантируют получение выходных данных; протоколы, которые обеспечивают гарантированную выдачу выходных данных, и т. д.

Все необъясняемые ниже понятия можно найти в [1, 5, 8].

1. Подпротоколы основного протокола

Схема разделения секрета. В нашем случае будет использована реплицированная схема разделения секрета с порогом два. Заметим, что для приводимых ниже протоколов достаточно алгебраической структуры кольца, поэтому будем через R обозначать некоторое конечное кольцо (например, кольцо вычетов по произвольному модулю).

Протокол 1 (реплицированная схема разделения секрета)**.**

Вход: $s \in R$ — секрет.

Дилер выбирает $s_1, s_2, s_3 \in_R R$ и определяет $s_4 = s - (s_1 + s_2 + s_3)$, где $\in_R$ означает, что выбор происходит равновероятным образом.

Выход:

P_1 получает $\{s_2, s_3, s_4\}$. P_2 получает $\{s_1, s_3, s_4\}$.

P_3 получает $\{s_1, s_2, s_4\}$. P_4 получает $\{s_1, s_2, s_3\}$.

Через $[s]$ будем обозначать вектор долей секрета s:

$$[s] = \big((s_2, s_3, s_4),\ (s_1, s_3, s_4),\ (s_1, s_2, s_4),\ (s_1, s_2, s_3)\big).$$

Так как реплицированная схема разделения секрета является линейной, то

$$[x] + [y] = \big((x_2, x_3, x_4),\ (x_1, x_3, x_4),\ (x_1, x_2, x_4),\ (x_1, x_2, x_3)\big) +$$

$$+\big((y_2, y_3, y_4),\ (y_1, y_3, y_4),\ (y_1, y_2, y_4),\ (y_1, y_2, y_3)\big) =$$

$$= \big((x_2 + y_2, x_3 + y_3, x_4 + y_4),\ (x_1 + y_1, x_3 + y_3, x_4 + y_4),\ (x_1 + y_1, x_2 + y_2, x_4 + y_4),$$

$$(x_1 + y_1, x_2 + y_2, x_3 + y_3)\big) = [x + y].$$

Аналогично, $c \cdot [s] = [c \cdot s]$.

Как следует из названия, реплицированная (повторяемая) схема разделения секрета обладает некоторой избыточностью. Это позволяет использовать простые и эффективные протоколы. Далее мы опишем примитивы, которые необходимы для основного протокола.

Совместная передача сообщений. Рассмотрим протокол совместной передачи сообщений, который позволяет паре участников, знающих общий секрет, корректно передавать его другому участнику. Как сказано выше (при рассмотрении долей произведения xy), участники P_s и P_t знают значения x_i, x_j, y_i, y_j, где $i \neq j$, $\{s, t\} = \{1, 2, 3, 4\} \setminus \{i, j\}$. Поэтому для передачи корректного вектора долей $[x_i y_j + x_j y_i]$ участникам P_i и P_j оба участника P_s и P_t будут участвовать в протоколе, причем хотя бы один из этих участников (P_s и P_t) является заведомо честным. Это будет показано в протоколе 3. Но сначала приведем вспомогательный протокол, который назовем Π_{jmp} (joint message passing). Под обозначением $\Pi_{jmp}(x, P_i, P_j, P_k)$ будем понимать то, что участники P_i и P_j обладают значением x, которое должны передать участнику P_k. Приведенный ниже протокол также будет использоваться для получения согласованных долей входных данных, а также для восстановления выходных значений участников.

Протокол 2 (совместная передача сообщений, Π_{jmp})**.**

Вход: участники P_i и P_j обладают значением x.

H — хеш-функция, устойчивая к коллизиям.

Выход: участник P_k получает x.

Протокол: P_i передает x участнику P_k.

Проверка: P_j передает сообщение $h = H(x, \ldots)$ участнику P_k, который проверяет, что значение h согласовано со значением x (т. е. значения хеш-функции равны), которое ранее отправил P_i. Если согласованности нет, то P_k определяет $\perp$.

Идентификация мошенничества.

Если P_k определил $\perp$, то участники действуют следующим образом, чтобы получить набор из не более двух участников, который точно содержит нечестного участника.

1) P_k транслирует сообщение «`обвинение`, P_i, P_j, h_i, h_j», где $h_i = H(x, \ldots)$ — значение хеш-функции от x, переданного участником P_i, h_j — значение, полученное от P_j.

2) Если $h_i = h_j$, то участники возвращают в качестве результата множество $\{P_k\}$. Иначе происходит следующее.

   - Если h_i отлично от значения хеш-функции от сообщения, которое передал P_i участнику P_k, то P_i транслирует «`обвинение`, P_k». После этого участники определяют $\{P_i, P_k\}$ как результат.
   - Если h_j отлично от значения хеш-функции, которое передал P_j участнику P_k, то P_j транслирует «`обвинение`, P_k». После этого участники определяют $\{P_j, P_k\}$ как результат.
   - Если оба участника P_i и P_j обвиняют P_k, то участники определяют $\{P_k\}$.
   - Если ни один из участников P_i и P_j не обвиняет P_k, то участники определяют $\{P_i, P_j\}$.

Разделение на доли входных данных. Пусть PRG — псевдослучайный генератор, возвращающий значение $x \in R$. Назовем следующий протокол Π_{input} (shared input).

Протокол 3 (разделение на доли входных данных, Π_{input})**.**

Предварительный этап. P_i, P_j, P_t знают доли ключа (или сам ключ) k_s для псевдослучайного генератора PRG, причем $\{i, j, s, t\} = \{1, 2, 3, 4\}$.

Вход: P_i и P_j обладают входным значением x.

Выход: $[x]$, причем участники P_s и P_t знают только свои компоненты вектора $[x]$.

Протокол.

1) Каждый из P_i, P_j и P_t определяет $x_s = PRG(k_s)$. Участник P_t обладает долями $(x_i, x_j, x_s) = (0, 0, x_s)$, при этом он не знает x_t, так как он не знает x.

2) P_i и P_j определяют $x_i = 0$, $x_j = 0$, $x_t = x - x_i - x_j - x_s = x - x_s$.

3) P_i и P_j запускают протокол $\Pi_{jmp}(x_t, P_i, P_j, P_s)$, в результате чего P_s получает x_t. Участник P_s обладает долями $(x_i, x_j, x_t) = (0, 0, x_t)$, при этом он не знает x_s.

Идентификация мошенничества. Если в результате протокола участник P_s получил $\perp$, то данный протокол возвращает множество участников, которые возвратил протокол Π_{jmp}.

Замечание 1. Ключ k_s, который знают только участники P_i, P_j и P_t, можно сгенерировать на предварительном этапе, например, следующим образом. Для этого каждый из данных участников генерирует свое значение ключа и передает его остальным (из заданных трех) участников. Теперь у каждого участника три ключа. Каждый участник применяет к этим трем ключам некоторую заранее фиксированную функцию (например, поразрядное сложение по модулю два всех трех ключей). После этого эти три участника используют это-трансляцию (между собой) для проверки того, что получилось одно и то же значение, передавая друг другу свое значение итогового ключа. Если у всех получилось одно и то же, то это и есть k_s.

Протокол Π_{jmp} будет использоваться для получения долей значений вида $x_iy_j + x_jy_i$ при $i < j$. Для случая разделения долей значения x_iy_i не требуется взаимодействия участников. В этом случае три участника с индексами $\{1, 2, 3, 4\} \setminus \{i\}$ знают x_i и y_i. Например, пусть участники P_1, P_2, P_3 знают x. Тогда они полагают $x_1 = x_2 = x_3 = 0$, $x_4 = x$. Это значит, что долей участника P_4 будет $(x_1, x_2, x_3) = (0, 0, 0)$. Обозначим этот локальный метод через $[x] \leftarrow \Pi_{input\_local}(x, P_i, P_j, P_k)$.

Безопасное умножение. Ниже приводится протокол безопасного умножения, который принимает на вход два вектора долей $[x]$ и $[y]$ и возвращает вектор долей $[x \cdot y]$.

Протокол 4 (протокол умножения Π_{mult})**.**
Вход: $[x]$ и $[y]$.
Выход: $[x \cdot y]$.
Протокол.

1) Для любой пары $s, t \in \{1, 2, 3, 4\}$, для которой $s < t$, участники P_i и P_j, где $\{i, j\} = \{1, 2, 3, 4\} \setminus \{s, t\}$, которые знают x_s, x_t, y_s, y_t, запускают протокол Π_{input}: $[x_sy_t + x_ty_s] \leftarrow \Pi_{input}(x_sy_t + x_ty_s,\ P_i,\ P_j)$.

2) Для каждого $s = 1, 2, 3, 4$ участники вызывают неинтерактивный метод $[x_sy_s] \leftarrow \Pi_{input\_local}(x_sy_s, P_i, P_j, P_t)$.

3) Участники локально вычисляют доли вектора долей

$$[x \cdot y] = \sum_{1 \le i < j \le 4} [x_iy_j + x_jy_i] + \sum_{i=1}^{4} [x_iy_i].$$

Идентификация мошенничества. Если в результате протокола на каком-то шаге участник получил $\perp$, то данный протокол возвращает множество участников, которые возвратил протокол Π_{jmp}.

Рассмотрим коммуникационную сложность протокола умножения. Всего существует 6 пар вида $1 \leq i < j \leq 4$. Для каждой из этих пар происходит один вызов протокола Π_{input}, в котором передается один элемент кольца. Поэтому асимптотическая коммуникационная сложность составляет 6 элементов кольца. При этом не требуется какой-либо предварительной обработки.

2. Протокол для вычисления арифметической схемы над кольцом

Сначала дадим определение арифметической схемы.

Определение 1. *Арифметической схемой* над полем F называется совокупность $C = (G, V_{in}, V_{out}, \lambda)$, где

- $G = (V, E)$ — ориентированный ациклический граф, в котором входящие ребра каждой вершины упорядочены;
- $V_{in} \subseteq \{v \in V \mid \overrightarrow{deg}(v) = 0\}$ и $V_{out} \subseteq V$ — множества входных и выходных вершин схемы, где $\overrightarrow{deg}(v)$ означает число входящих ребер в вершину v;
- λ для каждой вершины представляет собой отображение (операцию) $\lambda : F^{\overrightarrow{deg}(v)} \to F$.

Будем считать, что участники синхронно обмениваются данными по защищенным и аутентифицированным каналам. Если канал, соединяющий двух участников, является защищенным, то противник ничего не может узнать о сообщениях, которыми обмениваются эти два участника. Аутентифицированный канал гарантирует, что никто не сможет изменить сообщение, передаваемое по каналу, во время его передачи.

Полный четырехсторонний протокол работает следующим образом. Участники сначала обмениваются своими входными данными, используя схему разделения секрета и протокол Π_{jmp} для проверки согласованности долей. Например, участник P_1 разделяет свое входное значение x на доли x_1, x_2, x_3, x_4 с условием $x = x_1 + x_2 + x_3 + x_4$. Участники P_2, P_3, P_4 соответственно получают (x_1, x_3, x_4), (x_1, x_2, x_4), (x_1, x_2, x_3). Рассмотрим значение x_2. Участник P_1 передает участнику P_4 значение x_2, а участник P_3 передает значение хеш-функции от x_2 участнику P_4 с помощью протокола Π_{jmp}. Это гарантирует, что у P_1, P_3, P_4 будет одинаковое значение x_2. Аналогично и с другими значениями.

Затем участники вычисляют каждый элемент схемы (сложение, умножение на константу, умножение) в соответствии с заданным топологическим порядком для схемы (учитывая определение арифметической схемы). При этом для элементов сложения и умножения на константу вычисления являются локальными, для элемента умножения используется протокол 4. Наконец, участники восстанавливают свои выходные данные на выходных проводах схемы с помощью протокола Π_{jmp}.

Протокол 5 (вычисление арифметической схемы над кольцом R)**.**

Вход. Каждый участник P_j, $j = 1, 2, 3, 4$, обладает входным значением $x_j \in R^l$. Участники обладают описанием арифметической схемы C, которая вычисляет функциональность f, с входными данными (общей) длины $M = 4 \cdot l$.

Протокол.

1) *Получение долей входных данных на основе схемы разделения секрета.*

   а) Для каждого значения s_i участника P_j этот участник с помощью схемы разделения секрета и протокола Π_{jmp} разделяет секрет s_i среди всех участников следующим образом. Пусть для определенности $j = 1$. Участник P_1 разделяет свое входное значение s_i на доли x_1, x_2, x_3, x_4 с условием $s_i = x_1 + x_2 + x_3 + x_4$. Участники P_2, P_3, P_4 соответственно должны получить (x_1, x_3, x_4), (x_1, x_2, x_4), (x_1, x_2, x_3). Сначала происходят следующие действия.

   - Участник P_1 передает участнику P_2 значения x_1, x_3, x_4.
   - Участник P_1 передает участнику P_3 значение x_2.

   Теперь используется протокол Π_{jmp}.

   - Участники P_1 и P_2 запускают протокол $\Pi_{jmp}(x_1, x_4, P_1, P_2, P_3)$, в результате чего P_3 получает x_1 и x_4.
   - Участники P_1 и P_2 запускают протокол $\Pi_{jmp}(x_1, x_3, P_1, P_2, P_4)$, в результате чего P_4 получает x_1 и x_3.
   - Участники P_1 и P_3 запускают протокол $\Pi_{jmp}(x_2, P_1, P_3, P_4)$, в результате чего P_4 получает x_2.

   Теперь участники P_1, P_2, P_3, P_4 соответственно владеют долями (x_2, x_3, x_4), (x_1, x_3, x_4), (x_1, x_2, x_4), (x_1, x_2, x_3) секрета s_i.

   Аналогичные действия проделывают участники P_2, P_3, P_4.

   б) Каждый участник P_j сохраняет вектор долей $(s_1^j, \ldots, s_M^j)$ всех входных значений.

2) *Вычисление арифметической схемы.* Пусть $G_1, \ldots, G_T$ — топологический порядок следования элементов схемы. Для $k = 1, \ldots, T$ участники делают следующее.

   - Пусть G_k является элементом сложения. Пусть $[s_1]$ и $[s_2]$ — векторы долей участников на входных проводах элемента G_k. Тогда на выходном проводе элемента G_k будет вектор долей $[s_1] + [s_2] = [s_1 + s_2]$, который вычисляется участниками локально.
   - Пусть G_k является элементом умножения на константу $c \in R$. Пусть $[s]$ — вектор долей участников на входном проводе элемента G_k. Тогда на выходном проводе элемента G_k будет вектор долей $c \cdot [s] = [c \cdot s]$, который вычисляется участниками локально.

- Пусть G_k является элементом умножения. Пусть $[s_1]$ и $[s_2]$ — векторы долей участников на входных проводах элемента G_k. Тогда на выходном проводе элемента G_k будет вектор долей $[s_1] \cdot [s_2] = [s_1 \cdot s_2]$, который вычисляется с помощью протокола Π_{mult}.

3) *Восстановление выходных значений.* Для каждого выходного провода схемы C участники с помощью протокола Π_{jmp} следующим образом передают свои доли вектора $[s]$ участнику P_j для получения им $[s]$, где $[s]$ — вектор долей значения на выходном проводе, соответствующее участнику P_j. Пусть i — минимальный элемент множества $\{1,2,3,4\} \setminus \{j\}$, k — минимальный элемент множества $\{1,2,3,4\} \setminus \{j,i\}$. Участники P_i и P_k запускают протокол $\Pi_{jmp}(x_j, P_i, P_k, P_j)$, в результате чего P_j получает x_j. На основе вектора долей $[s]$ участник P_j получает выходное значение.

Замечание 2. Пусть выходной провод w_i схемы C соответствует участнику P_j, $[s]$ — вектор долей, соответствующий проводу w_i, т. е. участник P_j должен получить все доли вектора $[s]$ для вычисления своего выходного значения s. Пусть все участники P_i, $i \in \{1,2,3,4\} \setminus \{j\}$, передали участнику P_j значение x_j, т. е. в передаче участвуют не два (как в протоколе), а три участника. Тогда в качестве x_j участник P_j выбирает то значение, которое встречается чаще всего, учитывая что из полученных трех значений два значения принадлежат честным участникам. Как только P_j получил вектор $[s]$, он восстанавливает выходное значение s. При таком способе получения выходных данных будет достигнуто свойство справедливости.

Заключение

Протоколы безопасных вычислений позволяют нескольким участникам с личными входными данными совместно вычислять функцию своих входных данных, не раскрывая ничего, кроме выходных данных. Безопасность протокола определяется тем, что он ведет себя как идеальное исполнение с доверенной стороной, которая вычисляет функцию для участников. Среди всех возможных настроек поведения противника случай протоколов с честным большинством и с прерыванием — это один из случаев, которому уделяется много внимания из-за его практической эффективности, и он уже используется для таких приложений, как безопасное обучение и прогнозирование методом машинного обучения.

Список литературы

1. Рацеев С. М. *Криптография. Безопасные многосторонние вычисления* : учеб. пособие для вузов. СПб. : Лань, 2025. 468 с.

2. Рацеев С. М. *Криптографические протоколы. Схемы разделения секрета* : учебное пособие для вузов. СПб. : Лань, 2024. 336 с.

3. Chandramouli A., Choudhury A., Patra A. A Survey on Perfectly-Secure Verifiable Secret-sharing // *ACM Computing Surveys.* 2022, v. 54, i. 11, p. 1–36.

4. Dalskov A., Escudero D., Keller M. Fantastic four: Honest-majority four-party secure computation with malicious security // *Cryptology ePrint Archive*, Report 2020/1330, 2020.

5. Escudero D. An introduction to secret-sharing-based secure multiparty computation // *Cryptology ePrint Archive, paper* 2022/062. 2022. https://eprint.iacr.org/2022/062.

6. Evans D., Kolesnikov V., Rosulek M. A pragmatic introduction to secure multi-party computation // *Foundations and Trends® in Privacy and Security.* 2018,v. 2, № 2-3, p. 70–246.

7. Feng D., Yang K. Concretely efficient secure multi-party computation protocols: survey and more // *Security and Safety.* 2022, v 1, p. 1–43.

8. Hazay C., Lindell Y. *Efficient Secure Two-party Protocols: Techniques and Constructions.* Information security and cryptography. Springer Berlin Heidelberg, 2010.

Active secure four-party computation with an honest majority

***Ratseev, S. M.*[1,*], *Cherevatenko, O.I.*[2]**

*ratseevsm@mail.ru

[1]Ulyanovsk State University, Russia

[2]Ul'yanovsk State Pedagogical University, Russia

Secure multiparty computation (MPC) enables a set of parties to securely carry out a joint computation of their private inputs without revealing anything but the output. Protocols for semi-honest adversaries guarantee security as long as the corrupted parties run the specified protocol and ensure that nothing is leaked in the transcript. In contrast, protocols for malicious adversaries guarantee security in the presence of arbitrary adversaries who can run any attack strategy. Secret sharing plays an important role for maintaining privacy during the computation. In 2020, the authors Dalskov A., Escudero D., Keller M. introduced a new four-party honest-majority MPC protocol with active security that achieves comparable efficiency to equivalent protocols in the same setting, while having a much simpler design and not relying on functiondependent preprocessing. The authors did not provide a complete protocol. This paper provides a complete protocol for secure multiparty computation.

Keywords: *cryptographic protocol, multiparty computation, secret sharing*