УДК 004.056 DOI 10.18522/2311-3103-2023-4-26-34

Л.К. Бабенко, И.Д. Русаловский

ПОБИТОВЫЕ ГОМОМОРФНЫЕ ОПЕРАЦИИ НАД ЧИСЛАМИ С ПЛАВАЮЩЕЙ ТОЧКОЙ*

Гомоморфная криптография – это особый вид криптографии, который позволяет выполнять операции над зашифрованными данными без их предварительной расшифровки. Благодаря этим особенностям гомоморфная криптография может эффективно применяться для выполнения безопасных облачных вычислений. Для решения различных прикладных задач требуется поддержка всех математических операций, а также поддержка рациональных чисел, чтобы эффективно реализовать операцию деления и снизить потери точности во время округлений результата. Также для повышения точности вычислений необходимо использовать числа в формате с плавающей точкой, однако эта тема недостаточно проработана. Поддержка всех арифметических и логических операций в рамках одной схемы гомоморфного шифрования позволит выполнить гомоморфную реализацию практически любого алгоритма обработки данных, а представление чисел в формате с плавающей точкой позволит повысить точность вычислений и максимальную размерность обрабатываемых данных при том же объеме потребляемой памяти, если сравнивать с побитовым гомоморфным алгоритмом над целыми числами. К примеру, для решения СЛАУ методом Гаусса необходима поддержка операций разности, умножения, деления и сравнения чисел, а также необходимо представлять числа в формате с плавающей точкой, иначе во время обратного хода после каждой операции деления будет возникать округление результата, и ошибка будет накапливаться. В данной статье рассматривается возможность выполнения гомоморфных побитовых операций над числами в формате с плавающей точкой. Рассматривается наиболее распространенный формат представление чисел в формате с плавающей точкой – IEEE 754. Рассматриваются альтернативные решения для гомоморфной обработки рациональных чисел. Выполнен анализ возможности реализации побитовых гомоморфных арифметических операций – сложения, разности, умножения и деления, над зашифрованными гомоморфно числами в формате с плавающей точкой. Анализируются сложности, возникающие при реализации гомоморфных арифметических операций, рассматриваются способы их решения и приводятся результирующие алгоритмы над гомоморфно зашифрованными данными. Выполняется анализ полученных результатов и даются рекомендации касательно выбора способа представления гомоморфно зашифрованных данных в зависимости от решаемой задачи.

Гомоморфное шифрование; криптографическая защита; методы и алгоритмы; числа в формате с плавающей точкой; стандарт IEEE 754.

L.K. Babenko, I.D. Rusalovsky

BITWISE HOMOMORPHIC OPERATIONS ON FLOATING POINT NUMBERS

Homomorphic cryptography is a special kind of cryptography that allows you to perform operations on encrypted data without first decrypting it. Due these features, homomorphic cryptography can be effectively used to perform secure cloud computing. To solve various applied problems, support for all mathematical operations is required, as well as support for rational numbers in order to effectively implement the division operation and reduce the loss of accuracy during rounding of the result. Also, to improve the accuracy of calculations, it is necessary to use numbers in floating point format, but this topic has not been sufficiently researched. Support for all arithmetic and logical operations within a single homomorphic encryption scheme will allow us to perform a homomorphic implementation of almost any data processing algorithm, and the representation of numbers in floating point format will improve the accuracy of calculations and the maximum dimension of the processed data with the same amount of memory consumed, when compared with bitwise homomorphic

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-90140.

algorithm over integers. For example, to solve SLAE by the Gaussian method, it is necessary to support the operations of difference, multiplication, division, and comparison of numbers, and it is also necessary to represent numbers in floating point format, otherwise, during the back substitution after each division operation, rounding of the result will occur, and the error will accumulate. This article discusses the possibility of performing homomorphic bitwise operations on numbers in floating point format. The most common floating-point representation format, IEEE 754, is considered. Alternative solutions for homomorphic processing of rational numbers are considered. An analysis is made of the possibility of implementing bitwise homomorphic arithmetic operations - addition, difference, multiplication and division, over homomorphically encrypted numbers in floating point format. Difficulties arising in the implementation of homomorphic arithmetic operations are analyzed, methods for solving them are considered, and the resulting algorithms for homomorphically encrypted data are presented. The analysis of the results obtained is carried out and recommendations are given regarding the choice of a method for representing homomorphically encrypted data, depending on the problem being solved.

Homomorphic encryption; cryptographic protection; methods and algorithms; floating point numbers; IEEE 754 standard.

Введение. Гомоморфная криптография – это особый вид криптографии, позволяющий выполнять операции над зашифрованными данными без их предварительной расшифровки [1–11]. В общем виде гомоморфную криптографию можно представить следующим образом. Пусть $E(m)$ – некоторая функция шифрования, $D(c)$ – функция расшифрования, обратная функции E, где m – открытые данные, c – зашифрованные данные. Функция E называется гомоморфной относительно некоторой операции op над открытыми данными, если существует эффективный алгоритм M, который удовлетворяет условию:

$$m_1 \; op \; m_2 = D(M(E(m_1), E(m_2))). \quad (1)$$

Благодаря своим особенностям гомоморфная криптография может эффективно использоваться в следующих сферах:

- Облачные вычисления.
- Облачная обработка изображений.
- Электронное голосование (выборы).
- Защищенный поиск информации.

Применение гомоморфного шифрования в облачных сервисах [12–17] гарантирует, что данные не будет перехвачены даже в случае подмены сервера, т.к. они остаются зашифрованными на протяжении всего процесса передачи и обработки, а к секретному ключу имеет доступ только пользователь. Благодаря этому повышается уровень защищенности конфиденциальных данных и, как следствие, повышается уровень доверия пользователей к облачным технологиям.

На данный момент гомоморфная криптография только начинает свое развитие, для эффективного применения на практике необходима разработка методов и средств гомоморфной криптографии, с помощью которых будет возможно выполнение гомоморфных реализаций для различных алгоритмов обработки данных, применяемых для решения прикладных задач. К примеру, одной из таких задач в вычислительной алгебре является задача решения систем линейных алгебраических уравнений (СЛАУ). Для эффективного решения многих прикладных задач необходима поддержка чисел с плавающей точкой для повышения точности вычислений, а также необходима поддержка всех возможных арифметических и логических операций, что позволит выполнить гомоморфную реализацию практически любого алгоритма. Однако существующие схемы гомоморфного шифрования не предоставляют эффективное решение данной проблемы, поэтому актуальна разработка схемы шифрования или метода, который позволит эффективно выполнять арифметические и логические операции над числами в формате с плавающей точкой.

Альтернативные решения. Битовые операции над целыми. Как было рассмотрено в предыдущей статье [18–19], возможно преобразовать алгоритм над целыми числами в алгоритм над дробными числами с низкой точностью вычислений. Предположим, что дано целое число *m*, алгоритм шифрования *Enc*, алгоритм расшифрования *Dec*, гомоморфная реализация операций сложения, разности, умножения и деления. Тогда для поддержки дробных чисел можно ввести дополнительный коэффициент *k*, равный 10^n, где *n* - число знаков после запятой, определяющее точность вычислений. Перед шифрованием число умножается на этот коэффициент, а также после каждой операции результат корректируется с учетом этого коэффициента.

Умножение:

$$c_1 \otimes c_2 = \frac{Enc(m_1 * k) \otimes Enc(m_2 * k)}{Enc(k)} = Enc(m_1 * m_2 * k). \quad (2)$$

Сумма и разность:

$$c_1 \oplus c_2 = Enc(m_1 * k) \oplus Enc(m_2 * k) = Enc((m_1 + m_2) * k). \quad (3)$$

Деление:

$$c_1 / c_2 = \frac{Enc(m_1 * k)}{Enc(m_2 * k)} * Enc(k) = Enc(\frac{m_1}{m_2} * k). \quad (4)$$

После расшифровки необходимо разделить полученный результат на коэффициент и будет получен результат выполнения операций в виде рационального числа. Это достаточно простой в реализации и гибкий подход, подходящий для любого полностью гомоморфного алгоритма над целыми числами, однако точность вычислений намного ниже, чем в случае использования представления чисел с плавающей точкой.

Альтернативные решения. Схемы с поддержкой рациональных чисел. На данный момент разработаны различные схемы полностью гомоморфного шифрования, а на их базе выполнены программные реализации [8]. Рассмотрим наиболее популярные из них:

- BGV, поддерживает обработку целых чисел.
- BFV, поддерживает обработку целых чисел.
- CKKS, поддерживает обработку вещественных чисел.

Наибольший интерес в нашем анализе представляет схема CKKS, так как поддерживает обработку вещественных чисел. Кроме того, схема позволяет вычислять и трансцендентные функции (экспонента, логарифм и т.д.) с помощью разложения в ряд Тейлора. Однако получаемый после расшифровки результат является приближенным и имеет достаточно большую погрешность, что не позволяет использовать данную схему в задачах, где требуется высокая точность вычислений. Также использование вышеописанных алгоритмов не позволяет выполнять логические операции над шифротекстами, что может быть необходимо в рамках решения прикладных задач.

Представление чисел в ЭВМ. В современных ЭВМ числа с плавающей точкой, как правило, представляются в прямом коде в виде мантиссы и порядка. Перед записью в память числа нормализуются, а для экономии места первая значащая единица мантиссы не записывается, но подразумевается. Один из наиболее распространенных форматов представления чисел с плавающей точкой – IEEE 754 [20]. Кроме представления самих чисел стандарт имеет несколько особых значений, отражающих ошибки при выполнении вычислений. Как правило в программных комплексах используются форматы представления чисел одинарной точности (32 бита) и двойной точности (64 бита). Однако также встречаются кратные реализации – половинная точность (16 бит), четверная точность (128 бит) и так далее.

Рассмотрим в качестве примера представление числа в половинной точности. Другие форматы представляются абсолютно аналогично, однако имеют другую размерность составляющих. Стандарт определяет для числа с плавающей точкой половинной точности следующий формат:

- Общая размерность – 16 бит.
- Знаковый бит – 1 бит.
- Смещенный порядок – 5 бит.
- Мантисса – 10 бит. При этом за счет неявной единицы точность составляет 11 бит.
- Знаковый бит определяет знак числа. 0 – положительное число, 1 – отрицательное.

Смещенный порядок – это значение порядка минус смещение. Смещение вводится специально, чтобы не вводить еще один бит знака. Смещение порядка равно половине максимального значения, например для числа половинной точности оно равно 01111_2. При этом значения смещенного порядка 00000_2 и 11111_2 – это специальные значения, используемые для представления специальных чисел (табл. 1).

Таблица 1

Значения чисел в формате половинной точности в стандарте IEEE 754

Порядок	Мантисса = 0	Мантисса ≠ 0
00000_2	+0 и -0	Денормализованные числа
00001_2 ... 11110_2	Нормализованные числа	
11111_2	$\pm\infty$	NaN («не число»)

Мантисса имеет неявный старший бит равный 1, если только смещенный порядок не равен 0. Это позволяет дополнительно увеличить точность до 11 бит, в то время как явно в памяти хранятся только 10 бит.

К примеру, представим числа 1.99 и -2.5 в данном формате:

$1.99_{10} = 1.1111110101_2 = 1.1111110101_2 * 100 = 0\ 01111\ 1111110101.$

$-2.5_{10} = -10.1_2 = -1.01 * 101 = 1\ 10000\ 0100000000.$

Рассмотрим выполнение основных арифметических операций над числами в формате с плавающей точкой.

Сложение и вычитание. Пусть m – мантисса, e – смещенный порядок, тогда сложение и вычитание чисел в формате с плавающей точкой выполняется по следующему алгоритму:

1. Вычисляем разницу порядков $p = p_1 - p_2$.
2. Если порядки равны и разность равна 0, переходим к следующему шагу, иначе выполняем выравнивание порядков. Для выравнивания порядков нужно мантиссу числа с меньшим порядком сдвинуть вправо на абсолютное значение разницы порядков |p|. Сдвинутые младшие разряды при этом теряются.
3. После выравнивания порядков можно выполнить требуемую операцию над мантиссами с учетом знака числа.
4. Полученное значение мантиссы нормализуется при необходимости.
5. Порядок результата берется равным большему порядку чисел.
6. Нормализовать результат при необходимости.

Умножение и деление. Пусть m – мантисса, e – смещенный порядок, тогда сложение и вычитание чисел в формате с плавающей точкой выполняется по следующему алгоритму:

1. При делении необходимо найти разницу порядков, при умножении – сумму.
2. Мантиссы необходимо умножить/разделить.
3. Нормализовать результат при необходимости.

Гомоморфная реализация. Ввиду того, что числа с плавающей точкой представляются в виде целочисленных мантиссы и порядка, их можно реализовать гомоморфно, зашифровав побитно с помощью полностью гомоморфного алгоритма шифрования над битами. Выполним анализ данного подхода.

Шифрование. Для шифрования представим шифруемое число в двоичном виде в формате с плавающей точкой. Каждый бит исходного числа шифруется с помощью полностью гомоморфного алгоритма шифрования над битами. Полученный массив зашифрованных битов является шифротекстом.

Расшифрование. Чтобы расшифровать шифротекст, необходимо применить к нему обратное преобразование. Каждый зашифрованный бит расшифровать с помощью полностью гомоморфного алгоритма шифрования над битами, полученный массив битов и будет открытым текстом в двоичном виде в формате с плавающей точкой. При необходимости полученный результат можно перевести в десятичную дробь.

Логические операции. Так как мантиссы нормализованы, определить равенство чисел можно сравнив все биты между собой. Сравнить числа на больше/меньше также возможно, если сравнить раздельно их порядки и мантиссы как целые числа [18–19].

Арифметические операции. Все операции над числами в формате с плавающей точкой состоят из операций над целочисленными порядком и мантиссой, следовательно, легко могут быть реализованы через битовые операции. Однако сложности возникают с операциями нормализации и приведения чисел к одной степени. Сложность заключается в том, что данные обрабатываются в зашифрованном виде, а управляющий алгоритм не имеет к ним доступа. Из-за этого он должен обрабатывать данные “вслепую” и должен корректно обработать любой возможный результат. Можно получить, к примеру, разницу степеней двух чисел, но эта разница будет в зашифрованном виде и воспользоваться ей для определения величины сдвига или числа итераций не получится. Рассмотрим каждую из этих операций подробнее.

Приведение чисел к одной степени. Данная операция осложнена тем, что и мантиссы, и порядки зашифрованы и управляющий алгоритм не знает, сколько раз ему нужно повторить сдвиг или в какой момент нужно остановиться. Поэтому понадобится найти модуль разницы степеней, а после этого циклически сдвигать меньшее из чисел вправо на один разряд и уменьшать разницу степеней на 1, пока степень не достигнет нуля. Так как значение разницы степеней неизвестно, необходимо рассматривать худший вариант - сдвиг мантиссы на максимальное число ее разрядов. Чем больше бит выделено на мантиссу, тем выше сложность данной операции.

Нормализация результата вычислений. В результате выполнения операций над мантиссами может получиться ненормализованный результат. Так как данные зашифрованы, мы не можем определить факт возникновения ненормализованного результата, поэтому нужно выполнять нормализацию после каждой операции. Однако можно предугадать диапазон отклонения числа после выполнения операции и упростить нормализацию для некоторых арифметических операций. Известно, что нормализованное число имеет формат 1.xxx в двоичном виде, а также может быть нулевым, т.е. находится в диапазоне 0 ИЛИ [1…2) в десятичном виде.

Сложение и разность. Обе данные операции в результате сводятся к операции суммы чисел, которые могут иметь одну из четырех комбинаций знаков. В случае, если суммируются два числа с одним знаком, результат не может уменьшиться, а может только увеличиться на один разряд за счет переноса в старший разряд, а модуль результата будет в диапазоне 0 ИЛИ [1…4). Если же склады-

ваются числа с разными знаками, то модуль результата может быть в диапазоне [0…2). Таким образом, при выполнении операции необходимо учитывать возможное переполнение на 1 бит. Для нормализации необходимо проверить, что она требуется, сравнив старший бит с единицей – если старший бит равен единице, а бит переполнения равен нулю, то число уже нормализовано. Иначе его необходимо либо сдвинуть вправо на 1 разряд, либо влево на n разрядов, пока в старший разряд не будет смещена единица. Над зашифрованными данными алгоритм будет иметь следующий вид. Пусть мантисса имеет n бит в памяти, старший единичный бит не записывается явно, бит переноса p. Тогда для нормализации числа необходимо:

1. Вычисляем бит, указывающий на необходимость нормализации e = p.
2. Сдвигаем число вправо на 1 разряд. Записываем в результат $(\gg m \wedge e) \vee (m \wedge \overline{e})$. Прибавляем к степени e. Обнуляем бит переноса.
3. Устанавливаем индекс цикла i = 0
4. Вычисляем бит, указывающий на необходимость нормализации e = $\overline{m_n}$
5. Сдвигаем число влево на 1 разряд. Записываем в число $(\ll m \wedge e) \vee (m \wedge \overline{e})$. Вычитаем из степени e.
6. Если i < n, возвращаемся к шагу 3, иначе завершаем алгоритм.

Умножение. Модуль произведения будет находиться в диапазоне 0 ИЛИ [1, 4). Поэтому при нормализации достаточно учесть бит переноса. Алгоритм нормализации будет иметь вид:

1. Вычисляем бит, указывающий на необходимость нормализации e = p.
2. Сдвигаем число вправо на 1 разряд. Записываем в число $(\gg m \wedge e) \vee (m \wedge \overline{e})$. Прибавляем к степени e. Обнуляем бит переноса.

Деление. Модуль частного будет находится в диапазоне 0 или (0,5…2) или бесконечность. Таким образом, если в результате деления не была получена бесконечность или ноль, то результат достаточно сдвинуть 1 раз вправо, чтобы нормализовать. Ноль и бесконечность – особые значения, при которых мантисса равна нулю. Таким образом алгоритм нормализации будет иметь вид:

1. Вычисляем бит, указывающий на необходимость нормализации $e = \overline{m_n} \wedge M \neq 0$.
2. Сдвигаем число влево на 1 разряд. Записываем в число $(\ll m \wedge e) \vee (m \wedge \overline{e})$. Вычитаем из степени e.

Заключение. В данной статье была рассмотрена возможность выполнения побитовых гомоморфных операций над числами в формате с плавающей точкой. Так как число в формате с плавающей точкой представляется в виде целочисленных мантиссы и порядка, его можно побитно зашифровать гомоморфно с применением полностью гомоморфного алгоритма шифрования над битами. Данное представление чисел позволит:

- расширить диапазон обрабатываемых чисел при той же размерности в битах.
- повысить точность вычислений.

Однако подобное представление чисел значительно увеличит сложность вычислений, особенно сложность сложения и разности, так как для выполнения этих операций необходимо выполнять две трудоемких операции – приведение чисел к одному порядку и нормализацию. Операции же умножения и деления будут усложнены незначительно, так как результат этих операций может незначительно отклоняться от нормализованного значения и операция нормализации достаточно проста. Поэтому имеет смысл выбирать представление чисел на основе задачи, которую необходимо решить. Если в процессе вычислений потребуется большое число операций сложения и разности, а точность вычислений не очень важна, то стоит воспользоваться целочисленным представлением чисел. Также можно добавить сдвиг к целочисленному представлению, если требуется поддержка дробных

чисел. В обратном случае, если требуется высокая точность, большая размерность обрабатываемых чисел, то имеет смысл воспользоваться представлением с плавающей точкой, несмотря на повышение трудоемкости выполнения операций сложения и разности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Бабенко М.Г., Голимблевская Е.И, Ширяев Е.М.* Сравнительный анализ алгоритмов гомоморфного шифрования на основе обучения с ошибками // Тр. института системного программирования РАН. – 2020. – Т. 8, № 2. – С. 37-52.
2. *Бабенко Л.К., Русаловский И.Д.* Библиотека полностью гомоморфного шифрования целых чисел // Известия ЮФУ. Технические науки. – 2020. – № 2. – С. 79-88.
3. *Бабенко Л.К., Русаловский И.Д.* Метод реализации гомоморфного деления // Известия ЮФУ. Технические науки. – 2020. – № 4. – С. 212-221.
4. *Бабенко Л.К., Трепачева А.В.* О нестойкости двух симметричных гомоморфных криптосистем, основанных на системе остаточных классов // Труды Института системного программирования РАН. – 2019. – Т. 18, № 1. – С. 230-262.
5. *Аракелов Г.Г.* Вопросы применения прикладной гомоморфной криптографии // Вопросы кибербезопасности. – 2019. – № 5 (33). – С. 70-74.
6. *Шачина В.А.* Гомоморфная криптография в базах данных // Прикладная математика и информатика: современные исследования в области естественных и технических наук: Матер. V Международной научно-практической конференции (школы-семинара) молодых ученых, Тольятти, 22–24 апреля 2019 года. – 2019. – С. 468-473.
7. *Трусова Ю.О., Вовк Н.Н., Анисимов Ю.А.* Увеличение скорости гомоморфного шифрования на основе криптосистемы Эль-Гамаля // Математика и математическое моделирование: Сб. материалов XIII Всероссийской молодежной научно-инновационной школы, Саров, 02–04 апреля 2019 года. – 2019. – С. 97-98.
8. *Гаража А.А., Герасимов И.Ю., Николаев М.В., Чижов И.В.* Об использовании библиотек полностью гомоморфного шифрования // International Journal of Open Information Technologies. – 2021. – Т. 9, № 3. – С. 11-22.
9. *Волянский Ю.* Усовершенствование системы поиска опасных слов с использованием гомоморфного шифрования // Инновации. Наука. Образование. – 2021. – № 38. – С. 687-695.
10. *Аракелов Г.Г., Михалев А.В.* Комбинация частично гомоморфных схем // Электронные информационные системы. – 2020. – № 3 (26). – С. 83-92.
11. *Ширяев Е.М., Сотникова Н.А., Ващенко И.С., Кучуков В.А.* Исследование производительности полностью гомоморфного шифрования для задач видеообработки // Наука и инновации в XXI веке: актуальные вопросы, открытия и достижения: Сб. статей XXII Международной научно-практической конференции, Пенза, 12 декабря 2020 года. – 2020. – С. 57-59.
12. *Петренко А.С.* О реализации полностью гомоморфной криптосистемы Джентри-Халеви-Смарта // The 2019 Symposium on Cybersecurity of the Digital Economy - CDE'19: Третья международная научно-техническая конференция, Казань, 22–24 мая 2019 года. – 2019. – С. 272-275.
13. *Петренко А.С.* О реализации частично гомоморфной криптосистемы Пэйе // The 2019 Symposium on Cybersecurity of the Digital Economy - CDE'19: третья международная научно-техническая конференция, Казань, 22–24 мая 2019 года. – 2019. – С. 269-271.
14. *Минаков С.С.* Основные криптографические механизмы защиты данных, передаваемых в облачные сервисы и сети хранения данных // Вопросы кибербезопасности. – 2020. – № 3 (37). – С. 66-75.
15. *Дерябин М.А., Кучеров Н.Н.* Обзор безопасных методов шифрования для облачных вычислений // Новости науки в АПК. – 2019. – № 3 (12). – С. 298-303.
16. *Бабенко Л.К., Шумилин А.С., Алексеев Д.М.* Алгоритм обеспечения защиты конфиденциальных данных облачной медицинской информационной системы // Известия ЮФУ. Технические науки. – 2021. – № 5 (222). – С. 120-134.
17. *Мартишин С.А., Храпченко М.В.* Основные подходы к работе с конфиденциальными данными в облачных вычислениях // Образование. Технологии. Качество: Матер. V Всероссийской научно-практической конференции, Саратов, 26 марта 2021 года. – 2021. – С. 125-129.

18. *Babenko Liudmila, Rusalovsky Ilya.* Homomorphic operations on integers via operations on bits // Proceedings - 2022 15th International Conference on Security of Information and Networks, SIN 2022. – 2022.
19. *Русаловский И.Д., Бабенко Л.К., Макаревич О.Б.* Разработка методов гомоморфного деления // Известия ЮФУ. Технические науки. – 2022. – № 4 (228). – С. 103-112.
20. IEEE Standard for Floating-Point Arithmetic, IEEE Computer Society, IEEE Std 754, 2019.

REFERENCES

1. *Babenko M.G., Golimblevskaya E.I, Shiryaev E.M.* Sravnitel'nyy analiz algoritmov gomomorfnogo shifrovaniya na osnove obucheniya s oshibkami [Comparative analysis of homomorphic encryption algorithms based on learning with errors], *Tr. instituta sistemnogo programmirovaniya RAN* [Proceedings of the Institute for System Programming of the RAS], 2020, Vol. 8, No. 2, pp. 37-52.
2. *Babenko L.K., Rusalovskiy I.D.* Biblioteka polnost'yu gomomorfnogo shifrovaniya tselykh chisel [Library of fully homomorphic encryption of integer numbers], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2020, No. 2, pp. 79-88.
3. *Babenko L.K., Rusalovskiy I.D.* Metod realizatsii gomomorfnogo deleniya [Method for implementing homomorphic division], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2020, No. 4, pp. 212-221.
4. *Babenko L.K., Trepacheva A.V.* O nestoykosti dvukh simmetrichnykh gomomorfnykh kriptosistem, osnovannykh na sisteme ostatochnykh klassov [On the instability of two symmetric homomorphic cryptosystems based on a system of residual classes], *Trudy Instituta sistemnogo programmirovaniya RAN* [Proceedings of the Institute of System Programming of the Russian Academy of Sciences], 2019, Vol. 18, No. 1, pp. 230-262.
5. *Arakelov G.G.* Voprosy primeneniya prikladnoy gomomorfnoy kriptografii [Issues of application of applied homomorphic cryptography], *Voprosy kiberbezopasnosti* [Cybersecurity issues], 2019, No. 5 (33), pp. 70-74.
6. *Shachina V.A.* Gomomorfnaya kriptografiya v bazakh dannykh [Homomorphic cryptography in databases], *Prikladnaya matematika i informatika: sovremennye issledovaniya v oblasti estestvennykh i tekhnicheskikh nauk: Mater. V Mezhdunarodnoy nauchno-prakticheskoy konferentsii (shkoly-seminara) molodykh uchenykh, Tol'yatti, 22–24 aprelya 2019 goda* [Applied mathematics and computer science: modern research in the field of natural and technical sciences: Proceedings of the V International scientific and practical conference (school-seminar) of young scientists, Tolyatti, April 22–24, 2019], 2019, pp. 468-473.
7. *Trusova Yu.O., Vovk N.N., Anisimov Yu.A.* Uvelichenie skorosti gomomorfnogo shifrovaniya na osnove kriptosistemy El'-Gamalya [Increasing the speed of homomorphic encryption based on the El-Gamal cryptosystem], *Matematika i matematicheskoe modelirovanie: Sb. materialov XIII Vserossiyskoy molodezhnoy nauchno-innovatsionnoy shkoly, Sarov, 02–04 aprelya 2019 goda* [Mathematics and mathematical modeling: Collection of materials of the XIII All-Russian Youth Scientific and Innovation School, Sarov, April 02–04, 2019], 2019, pp. 97-98.
8. *Garazha A.A., Gerasimov I.Yu., Nikolaev M.V., Chizhov I.V.* Ob ispol'zovanii bibliotek polnost'yu gomomorfnogo shifrovaniya [On the use of fully homomorphic encryption libraries], *International Journal of Open Information Technologies*, 2021, Vol. 9, No. 3, pp. 11-22.
9. *Volyanskiy Yu.* Usovershenstvovanie sistemy poiska opasnykh slov s ispol'zovaniem gomomorfnogo shifrovaniya [Improving the search system for dangerous words using homomorphic encryption], *Innovatsii. Nauka. Obrazovanie* [Innovations. The science. Education], 2021, No. 38, pp. 687-695.
10. *Arakelov G.G., Mikhalev A.V.* Kombinatsiya chastichno gomomorfnykh skhem [Combination of partially homomorphic schemes], *Elektronnye informatsionnye sistemy* [Electronic information systems], 2020, No. 3 (26), pp. 83-92.
11. *Shiryaev E.M., Sotnikova N.A., Vashchenko I.S., Kuchukov V.A.* Issledovanie proizvoditel'nosti polnost'yu gomomorfnogo shifrovaniya dlya zadach videoobrabotki [Study of the performance of fully homomorphic encryption for video processing tasks], *Nauka i innovatsii v XXI veke: aktual'nye voprosy, otkrytiya i dostizheniya: Sb. statey XXII Mezhdunarodnoy nauchno-prakticheskoy konferentsii, Penza, 12 dekabrya 2020 goda* [Science and innovation in the 21st century: current issues, discoveries and achievements: Collection of articles of the XXII International Scientific and Practical Conference, Penza, December 12, 2020], 2020, pp. 57-59.

12. *Petrenko A.S.* O realizatsii polnost'yu gomomorfnoy kriptosistemy Dzhentri-Khalevi-Smarta [On the implementation of a fully homomorphic Gentry-Halevi-Smart cryptosystem], *The 2019 Symposium on Cybersecurity of the Digital Economy - CDE'19: Tret'ya mezhdunarodnaya nauchno-tekhnicheskaya konferentsiya, Kazan', 22–24 maya 2019 goda* [The 2019 Symposium on Cybersecurity of the Digital Economy - CDE'19: Third International Scientific and Technical Conference, Kazan, May 22–24, 2019], 2019, pp. 272-275.
13. *Petrenko A.S.* O realizatsii chastichno gomomorfnoy kriptosistemy Peye [On the implementation of the partially homomorphic Payeux cryptosystem], *The 2019 Symposium on Cybersecurity of the Digital Economy - CDE'19: tret'ya mezhdunarodnaya nauchno-tekhnicheskaya konferentsiya, Kazan', 22–24 maya 2019 goda* [The 2019 Symposium on Cybersecurity of the Digital Economy - CDE'19: third international scientific and technical conference, Kazan, May 22–24, 2019], 2019, pp. 269-271.
14. *Minakov S.S.* Osnovnye kriptograficheskie mekhanizmy zashchity dannykh, peredavaemykh v oblachnye servisy i seti khraneniya dannykh [Basic cryptographic mechanisms for protecting data transmitted to cloud services and data storage networks], *Voprosy kiberbezopasnosti* [Issues of cybersecurity], 2020, No. 3 (37), pp. 66-75.
15. *Deryabin M.A., Kucherov N.N.* Obzor bezopasnykh metodov shifrovaniya dlya oblachnykh vychisleniy [Review of secure encryption methods for cloud computing], *Novosti nauki v APK* [Science news in agro-industrial complex], 2019, No. 3 (12), pp. 298-303.
16. *Babenko L.K., SHumilin A.S., Alekseev D.M.* Algoritm obespecheniya zashchity konfidentsial'nykh dannykh oblachnoy meditsinskoy informatsionnoy sistemy [Algorithm for ensuring the protection of confidential data of a cloud medical information system], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2021, No. 5 (222), pp. 120-134.
17. *Martishin S.A., Khrapchenko M.V.* Osnovnye podkhody k rabote s konfidentsial'nymi dannymi v oblachnykh vychisleniyakh [Basic approaches to working with confidential data in cloud computing], *Obrazovanie. Tekhnologii. Kachestvo: Mater. V Vserossiyskoy nauchno-prakticheskoy konferentsii, Saratov, 26 marta 2021 goda* [Education. Technologies. Quality: Materials of the V All-Russian Scientific and Practical Conference, Saratov, March 26, 2021], 2021, pp. 125-129.
18. *Babenko Liudmila, Rusalovsky Ilya.* Homomorphic operations on integers via operations on bits, *Proceedings - 2022 15th International Conference on Security of Information and Networks, SIN 2022*, 2022.
19. *Rusalovskiy I.D., Babenko L.K., Makarevich O.B.* Razrabotka metodov gomomorfnogo deleniya [Development of homomorphic division methods], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2022, No. 4 (228), pp. 103-112.
20. IEEE Standard for Floating-Point Arithmetic, IEEE Computer Society, IEEE Std 754, 2019.

Статью рекомендовал к опубликованию д.т.н. Г.Е. Веселов.

Бабенко Людмила Климентьевна – Южный федеральный университет; e-mail: blk@tsure.ru; г. Таганрог, Россия; тел.: +79054530191; кафедра безопасности информационных технологий; д.т.н.; профессор.

Русаловский Илья Дмитриевич – e-mail: ilya.rusalovskiy@mail.ru; тел.: +79885526701; кафедра безопасности информационных технологий; аспирант.

Babenko Liudmila Kliment'evna – Southern Federal University; e-mail: blk@tsure.ru; Taganrog, Russia; phone: +79054530191; the department of information technologies security; dr. of eng. sc.; professor.

Rusalovsky Ilya Dmitrievich – e-mail: ilya.rusalovskiy@mail.ru; phone: +79885526701; the department of information technologies security; postgraduate student.