РАЗРАБОТКА ОПЕРАЦИЙ ДЛЯ АЛГОРИТМОВ ГОМОМОРФНОГО ШИФРОВАНИЯ

Бабенко Л. К.[1], Русаловский И. Д.[2]

DOI: 10.21681/2311-3456-2024-2-101-106

Цель работы: *расширение круга выполняемых гомоморфных криптографических операций.*

Методы исследования: *теоретические основы математической логики, теория вероятностей, теория чисел, основы алгоритмизации, методы программирования, теория информационной безопасности, теория гомоморфного шифрования.*

Результаты исследования. *В статье рассматриваются результаты работ по разработке инструментария для прикладного применения гомоморфной криптографии. В статье рассматривается проблема гомоморфного деления, приводится краткий анализ возможности выполнения этой операции с помощью различных методов. Разрабатывается алгоритм гомоморфного деления на основе представления чисел в виде простых дробей. Также предлагается метод реализации операции гомоморфного сравнения. Рассматривается проблема выполнения арифметических и логических операций в рамках одного алгоритма полностью гомоморфного алгоритма шифрования, приводится краткий обзор побитной реализации арифметических операций с учетом особенностей гомоморфного шифрования. Решение всех перечисленных выше проблем позволит расширить возможности прикладного применения гомоморфной криптографии. В завершении статьи приводятся выводы и рекомендации по применению предложенных методов и алгоритмов для решения различных прикладных задач.*

Научная новизна: *Разработан новый метод, позволяющий выполнять гомоморфное деление на базе любого полностью гомоморфного алгоритма над целыми числами. Разработан новый метод гомоморфного сравнения чисел. Разработаны алгоритмы гомоморфной реализации побитовых целочисленных операций сложения, разности, умножения и деления. Разработаны алгоритмы гомоморфной реализации побитовых операций сложения, разности, умножения и деления над числами в формате с плавающей точкой.*

Ключевые слова: *информационная безопасность, криптографическая защита, безопасные вычисления, методы и алгоритмы, гомоморфная криптография, гомоморфное деление, гомоморфное сравнение, гомоморфная арифметика.*

DEVELOPMENT OF OPERATIONS FOR HOMOMORPHIC ENCRYPTION ALGORITHMS

Babenko L. K.[3], Rusalovsky I. D.[4]

Purpose of the work: *expanding the number of available homomorphic cryptographic operations.*

Research methods: *theoretical foundations of mathematical logic, probability theory, number theory, fundamentals of algorithmization, programming methods, information security theory, homomorphic encryption theory.*

Research results. *The article discusses the results of work on the development of tools for the applied application of homomorphic cryptography. The article examines the problem of homomorphic division and provides a brief analysis of the possibility of performing this operation using various methods. An algorithm for homomorphic division is being developed based on representing numbers in the form of simple fractions.*

1 Бабенко Людмила Климентьевна, доктор технических наук, профессор, Южный Федеральный Университет «ЮФУ», Институт компьютерных технологий и информационной безопасности, г. Таганрог, Россия. E-mail: lkbabenko@sfedu.ru

2 Русаловский Илья Дмитриевич, аспирант, Южный Федеральный Университет «ЮФУ», Институт компьютерных технологий и информационной безопасности, г. Таганрог, Россия. E-mail: ilya.rusalovskiy@mail.ru

3 Liudmila K. Babenko, Dr.Sc., Professor, Southern Federal University «SFedU», Institute of Computer Technologies and Information Security, Taganrog, Russia. E-mail: lkbabenko@sfedu.ru

4 Ilya D. Rusalovsky, postgraduate student, Southern Federal University «SFedU», Institute of Computer Technologies and Information Security, Taganrog, Russia. E-mail: ilya.rusalovskiy@mail.ru

A method for implementing the homomorphic comparison operation is also proposed. The problem of performing arithmetic and logical operations within one algorithm of a fully homomorphic encryption algorithm is considered, and a brief overview of the bitwise implementation of arithmetic operations taking into account the features of homomorphic encryption is given. Solving all the problems listed above will expand the possibilities of applied applications of homomorphic cryptography. At the end of the article, conclusions and recommendations on the use of the proposed methods and algorithms for solving various applied problems are provided.

Scientific novelty. *A new method has been developed that allows you to perform homomorphic division based on any fully homomorphic algorithm over integers. A new method for homomorphic comparison of numbers has been developed. Algorithms for the homomorphic implementation of bitwise integer operations of addition, difference, multiplication and division have been developed. Algorithms for homomorphic implementation of bitwise operations of addition, difference, multiplication and division over numbers in floating point format have been developed.*

Keywords: *information security, cryptographic protection, secure computing, methods and algorithms, homomorphic cryptography, homomorphic division, homomorphic comparison, homomorphic arithmetic.*

Введение

Гомоморфная криптография – молодое направление в криптографии, которое начало свое активное развитие с 2009 года, когда была предложена первая полностью гомоморфная схема шифрования[5]. Особенность гомоморфного шифрования заключается в том, что оно позволяет обрабатывать данные в зашифрованном виде и получать зашифрованный результат, соответствующий после расшифровки результату выполнения соответствующей операции над незашифрованными данными. 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:

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

Благодаря своим особенностям гомоморфная криптография может эффективно использоваться в различных сферах, где требуется обработка данных третьей стороной [1–6]. К этим областям можно отнести:

- Облачные вычисления.
- Электронное голосование (выборы).
- Защищенный поиск информации.
- Нейронные сети.

Ввиду своей новизны, гомоморфное шифрование еще недостаточно проработано. Его основной проблемой является низкая скорость работы и высокие требования к вычислительным мощностям, поэтому большинство работ направлены на улучшение существующих алгоритмов [7–9], а также разработку новых алгоритмов, показывающих лучшее быстродействие или простоту реализации [10-11], также изучается стойкость существующих алгоритмов [11–13]. В существующих программных комплексах в основном реализованы только основные криптографические и математические операции. Обзор существующих программных комплексов представлен в таблице 1 [14]:

Таблица 1

Сравнение программных комплексов

Операции	SEAL	HElib	TFHE
Сумма, разность	Да	Да	Да
Умножение	Да	Да	Да
Деление	Нет	Нет	Нет
Сравнение	Нет	Нет	Нет
Условные операции	Нет	Нет	Да
Побитовые операции	Да	Да	Да
Матричные операции	Да	Да	Нет
Возведение в степень	Да	Да	Нет
Возведение в квадрат	Да	Да	Да
Отрицание	Да	Да	Нет

Как видно из таблицы, существующие программные комплексы не поддерживают операции деления и сравнения шифртекстов, а данные операции необходимы во многих алгоритмах обработки данных. К примеру, для решения СЛАУ методом Гаусса необходима поддержка операций деления и сравнения шифртекстов [15]. А для простейшей операции нахождения среднего арифметического нужна поддержка операции деления.

Также стоит отметить, что наличие поддержки операций над целыми числами и над битами в приведенной выше таблице не означает, что данные операции поддерживаются одновременно в рамках

5 Gentry C. A fully homomorphic encryption scheme. PhD. – 2009.

одной системы шифрования. К примеру, библиотека HElib на основе схемы BGV может работать и с битами, и с целыми числами в кольце в зависимости от того, какая размерность пространства открытого текста была выбрана в начальных параметрах системы. При этом схема над битами будет поддерживать гомоморфные операции логического «И» и «исключающее ИЛИ», а схема над целыми числами – сложение и умножение.

Отсюда следует актуальность разработки методов и алгоритмов, позволяющих выполнять гомоморфные операции деления и сравнения, а также позволяющих выполнять арифметические и логические операции в рамках одной криптосистемы.

Проблема гомоморфного деления

Проблема гомоморфного деления рассматривалась авторами в ряде статей [16-17]. Разработка метода гомоморфного деления была необходима, чтобы обеспечить поддержку всех арифметических операций над гомоморфно зашифрованными данными. В рамках исследования рассматривались различные алгоритмы и способы реализации гомоморфного деления. Кратко рассмотрим каждый из возможных подходов.

Алгоритмы над полиномами. Еще один вариант реализации гомоморфного алгоритма шифрования – соотнесение открытому тексту некоторого полинома. К примеру, Ф. Буртыка предложил алгоритм шифрования на основе матричных полиномов (полиномов, каждый коэффициент которых представлен матрицей) [10]. Также в выпускной квалификационной работе Яковлева[6] предлагается алгоритм шифрования посредством преобразования целого числа полиному с целочисленными коэффициентами. Между двумя полиномами можно выполнить операцию деления, результатом которой будут частное и остаток от деления. Как было указано в предыдущем примере, для решения некоторых задач может хватить точности деления, при которой остаток полностью отбрасывается. Однако в случае с шифртекстами на основе полиномов возникает ряд проблем.

Величина открытого текста никак не связана с порядком полинома, следовательно большему числу может соответствовать меньший полином и наоборот, а делимое полностью будет остатком.

Результатом деления будут частное и остаток, каждый из которых представлен полиномом. Если расшифровать их, разделить расшифрованный остаток на расшифрованный делитель, то мы получим корректный результат. Но частное и остаток в зашифрованном виде не соответствуют частному и остатку в расшифрованном виде. Таким образом, остаток от деления в зашифрованном виде может содержать большую часть частного, что делает операцию отбрасывания остатка некорректной, но не отбросить остаток мы не можем, так как в рамках алгоритма могут обрабатываться только полиномы.

Рассмотрим численный пример на основе алгоритма Яковлева. Пусть даны целые числа $m_1 = 4$, $m_2 = 1$, $p = 4$, $q = 2$, $x_0 = p / q = 2$ – секретный ключ. Выполним шифрование:

$$f_1(x) = 5x + 2; f_1(x_0) = 12$$
$$g_1(x) = 22 * f_1(x) - 22 * 12 + m_1 = 20x + 8 - 48 + 4 = 20x - 36$$
$$f_2(x) = 3x - 5; f_2(x_0) = 1$$
$$g_2(x) = 22 * f_2(x) - 22 * 1 + m_2 = 12x - 20 - 4 + 1 = 12x - 23$$

В результате деления $20x - 36$ на $12x - 23$ получим $g_3(x) = 1$, остаток $g_4(x) = 8x - 13$. После расшифрования получим:

$$D(g_3(x)) = g_3(x_0) = 1$$
$$D(g_4(x)) = g_4(x_0) = 8 * 2 - 13 = 3$$

Таким образом в результате деления получаем 1 + 3 = 4, однако на остаток пришлось 3, из за этого мы не можем отбросить остаток от деления, а следовательно, продолжать вычисления без перезашифрования результата.

Алгоритмы в кольце вычетов. Одним из вариантов построения гомоморфных алгоритмов является отображение в кольце вычетов. Примером такого алгоритма является RSA. Алгоритм RSA проявляет мультипликативный гомоморфизм, а в кольце вычетов можно найти обратный элемент. Следовательно, возможно реализовать операцию деления как умножение на обратное. Однако, операция деления во множестве действительных чисел R и в кольце вычетов Z_n не во всех случаях эквивалентна. Очевидно, что во множестве целых чисел результатом деления будет частное и остаток, в то время как во множестве действительных чисел – обыкновенной или десятичной дробью, однако для решения некоторых задач было бы достаточно деления с низкой точностью, в результате которого остаток бы полностью отбрасывался.

Рассмотрим кольцо Z_5 в качестве примера. Обратный элемент можно найти, воспользовавшись малой теоремой Ферма (2):

$$m^{-1} \bmod p = m^{p-2} \bmod p \quad (2)$$

Продемонстрируем проблему на численном примере. Пусть $m_1 = 2$, $m_2 = 4$, тогда:

$$m_1 / m_2 \bmod 5 = 2 * 4^{5-2} \bmod 5 = 3 \bmod 5$$

6 Яковлев М. О. Защищенный калькулятор. Разработка клиентского компонента. // Выпускная квалификационная работа бакалавра [Электронный ресурс]. – URL: http://www.nsu.ru/xmlui/bitstream/handle/nsu/471/Text_YakovlevMO.pdf (дата обращения 15.01.2024).

Как видно из примера, 2 / 4 = 3 в кольце Z_5, в то время как мы ожидали получить 0 или 1, в зависимости от стратегии округления результата. Следовательно, данное решение не подходит для реализации гомоморфного деления.

Метод деления на основе представления шифртекста в виде простой дроби. За основу берется любой полностью гомоморфный алгоритм шифрования над целыми числами, поддерживающий операции суммы, разности и умножения, открытый текст (целое или рациональное число) представляется в виде простой дроби. Делимое и делитель шифруются по отдельности с помощью полностью гомоморфного алгоритма шифрования, полученная зашифрованная дробь является шифртекстом. Операции над шифртекстами реализуются как операции над простыми дробями, а при расшифровке делимое и делитель расшифровываются раздельно и делятся друг на друга. Алгоритм можно представить в следующем виде:

Пусть дан некоторый полностью гомоморфный алгоритм шифрования над целыми, для которого определены $E(m)$ – алгоритм шифрования, $D(c)$ – алгоритм расшифрования, обратный к $E(m)$, $\otimes,\oplus$ – операторы гомоморфного умножения и сложения над зашифрованными данными соответственно, где m – открытый текст, c – шифртекст. Тогда схема шифрования целого числа m с поддержкой операции деления может быть построена следующим образом.

Алгоритм шифрования:

1. Представляем открытый текст m в виде простой дроби, где m_1 – делимое, m_2 – делитель.
2. Шифруем делимое и делитель с помощью полностью гомоморфного алгоритма шифрования над целыми числами: $a = E(m_1)$, $b = E(m_2)$
3. Шифртекст в предлагаемой схеме шифрования будет представлен в виде пары зашифрованных гомоморфно чисел: $c = (a; b)$

Алгоритм расшифрования:

1. Расшифруем гомоморфно зашифрованные делимое и делитель, в виде которых представлен шифртекст: $r_1 = D(a)$; $r_2 = D(b)$
2. Выполняем деление, чтобы получить результат в виде десятичной дроби: $r = r_1 / r_2$

Реализация математических операций:

1. Сложение. $C_1 + C_2 = (a_1 \otimes b_2 \oplus a_2 \otimes b_1; b_1 \otimes b_2)$
2. Умножение. $C_1 * C_2 = (a_1 \otimes a_2; b_1 \otimes b_2)$
3. Деление. $C_1 / C_2 = (a_1 \otimes b_2; b_1 \otimes a_2)$

Данный метод прост в реализации, универсален, с его помощью можно добавить операцию деления в любой полностью гомоморфный алгоритм шифрования над целыми. Также метод может быть полезен в том случае, когда реализация гомоморфного деления другим способом требует больших вычислительных мощностей. К минусам можно отнести увеличение размерности шифртекста приблизительно в два раза, так как он представлен двумя гомоморфно зашифрованными числами. Также увеличивается сложность выполнения других операций: умножение усложняется примерно в два раза (из-за необходимости выполнять ее дважды – для делимого и делителя), а сложность операций сложения и разности увеличивается приблизительно в 4 раза (при условии того, что вычислительная сложность операций гомоморфного сложения и умножения эквивалентна) из-за необходимости приведения числа к общему знаменателю.

Операции над целыми через битовые операции. В рамках данного подхода шифртекст представляется в виде массива зашифрованных гомоморфно битов. А все операции реализуются аналогично машинным операциям над битами, но с учетом того, что числа зашифрованы и, хотя операции над ними возможны, но управляющий алгоритм не знает значения того или иного бита. Подробнее битовые операции над целыми числами и числами в формате с плавающей точкой рассматриваются далее в статье.

Гомоморфное сравнение чисел

Сравнение чисел – важная операция, необходимая для гомоморфной реализации многих алгоритмов обработки данных. Например, в алгоритме Гаусса необходимо выполнять сравнение чисел на главной диагонали с нулем и, при необходимости, выполнять перестановку. Выполнить гомоморфное сравнение достаточно просто, если числа будут представлены в двоичном виде и зашифрованы побитно. Алгоритм гомоморфного сравнения чисел в этом случае можно представить следующим образом. Пусть даны числа A, B, зашифрованные с помощью полностью гомоморфного алгоритма шифрования над битами, гомоморфные операции «ИЛИ», «исключающее ИЛИ», отрицание, тогда гомоморфный результат сравнения двух чисел можно получить, вычислив данное выражение (3):

$$r = \overline{(E(a_0) \oplus E(b_0)) \vee E(a_1) \oplus E(b_1)) \vee \ldots \vee E(a_n) \oplus E(b_n))} \quad (3)$$

Числа равны, если все их биты равны. Стоит отметить, что полученный результат будет также гомоморфно зашифрованным битом и управляющий алгоритм не сможет получить его значение, поэтому необходимо будет адаптировать алгоритм таким образом, чтобы результат сравнения использовался в зашифрованном виде. К примеру, для реализации операции выбора из двух значений A и B на основе результата сравнения r, необходимо вычислить следующее выражение (4):

$$c_3 = (c_1 \wedge r) \vee (c_2 \wedge \bar{r}) \quad (4)$$

Таким образом возможна реализация операций сравнения гомоморфно зашифрованных чисел.

Реализация операций над целыми и рациональными числами через операции над битами

Гомоморфные алгоритмы шифрования можно разделить на алгоритмы над целыми и алгоритмы над битами в зависимости от того, какие данные шифруются. Кроме типа шифруемых данных, различаются также и поддерживаемые гомоморфные операции. Для целочисленных алгоритмов, как правило, поддерживаются операции сложения и умножения, а для алгоритмов над битами – логические операции «И» и «исключающее ИЛИ». Из-за этого при решении практических задач возникает проблема, что в рамках одной криптосистемы можно выполнять только небольшой перечень гомоморфных операций. В рамках криптосистемы над целыми числами нельзя реализовать операции над битами, а вот в криптосистеме над битами можно реализовать операции над целыми.

Целочисленные операции. Для реализации операций над целыми числами через операции над битами использовались алгоритмы побитовых операций, аналогичные машинным, которые были адаптированы с учетом особенностей обработки гомоморфно зашифрованных данных [17-18]. Числа в предложенном алгоритме представляются в двоичном виде и поэлементно зашифровываются с помощью полностью гомоморфного алгоритма над битами. Данный подход позволяет выполнять арифметические и логические операции в рамках одной криптосистемы. Конечно, из-за представления числа в виде массива зашифрованных гомоморфно битов увеличивается размер шифртекста, а также сложность вычислений. Также этот подход обеспечивает сравнительно небольшую точность вычислений и размерность открытых данных. В случае, если точность вычислений имеет ключевое значение, стоит рассмотреть представление чисел в формате с плавающей точкой.

Числа в формате с плавающей точкой. В современных ЭВМ числа с плавающей точкой, как правило, представляются в прямом коде в виде мантиссы и порядка. Один из наиболее распространенных форматов представления чисел с плавающей точкой – IEEE 754. При этом мантисса – число с фиксированной запятой в нормализованном виде в диапазоне 0∪[1,2), порядок – целое число. Все гомоморфные операции над числами в формате с плавающей точкой возможно реализовать на основе алгоритмов гомоморфной математики над целыми числами [19]. Однако из-за того, что числа зашифрованы, возникают сложности с реализацией операций нормализации и приведения чисел к одной степени. Нормализация мантиссы выполняется после каждой операции, чтобы она всегда оставалась в диапазоне 0∪[1,2), а приведение чисел к одному порядку необходимо для выполнения операций суммы и разности. Сложность при выполнении этих операций заключается в том, что управляющий алгоритм не знает, когда эти операции завершены, так как данные зашифрованы. Поэтому необходимо выполнять максимально возможное число итераций, чтобы быть уверенным в успешном завершении данных операций. Из-за этого сложность выполнения арифметических операций возрастает, по сравнению с побитовыми операциями над целыми числами. Однако представление чисел в формате с плавающей точкой позволяет повысить размерность шифруемых чисел при том же числе зашифрованных бит, а также многократно повышает точность вычислений, поэтому для решения задач, где требуется высокая точность, а также для задач, где выполняется большое число операций деления, имеет смысл использовать числа в формате с плавающей точкой.

Выводы

В рамках данной статьи рассмотрены некоторые из проблем прикладного применения гомоморфной криптографии, а также сделан обзор результатов, полученных в рамках исследований, посвященных решению вышеперечисленных проблем. В рамках исследования были рассмотрены несколько различных методов и алгоритмов, позволяющих расширить список гомоморфных операций и применять гомоморфную криптографию на практике. Гомоморфная криптография все еще достаточно медленная, поэтому вопрос о ее повсеместном внедрении пока не стоит, однако ее можно применять для обработки наиболее критических данных. Разработанные методы и средства гомоморфной криптографии имеют разные характеристики и подходят для решения разных задач. Желательно применять алгоритм, который обеспечивает минимальные требования. Так, если в рамках гомоморфной обработки необходимо выполнять только одну операцию, то имеет смысл рассмотреть частично гомоморфные схемы шифрования. А если же необходима поддержка всех арифметических и логических операций, а также высокая точность вычислений, то следует использовать гомоморфные побитовые операции над числами в формате с плавающей точкой.

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

Литература

1. *Аракелов Г. Г. Вопросы применения прикладной гомоморфной криптографии // Вопросы кибербезопасности. – 2019. – № 5(33). – С. 70–74.*
2. *Шачина В. А. Гомоморфная криптография в базах данных // Прикладная математика и информатика: современные исследования в области естественных и технических наук: Материалы V Международной научно-практической конференции (школы-семинара) молодых ученых, Тольятти, 22–24 апреля 2019 года. – 2019. – С. 468–473.*
3. *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.*
4. *Волянский Ю. Усовершенствование системы поиска опасных слов с использованием гомоморфного шифрования // Инновации. Наука. Образование. – 2021. – № 38. – С. 687–695.*
5. *Аракелов Г. Г., Михалев А. В. Комбинация частично гомоморфных схем // Электронные информационные системы. – 2020. – № 3(26). – С. 83–92.*
6. *Минаков С. С. Основные криптографические механизмы защиты данных, передаваемых в облачные сервисы и сети хранения данных // Вопросы кибербезопасности. – 2020. – № 3(37). С. 66–75.*
7. *Трусова Ю. О., Вовк Н. Н., Анисимов Ю. А. Увеличение скорости гомоморфного шифрования на основе криптосистемы Эль-Гамаля // Математика и математическое моделирование: Сборник материалов XIII Всероссийской молодежной научно-инновационной школы, Саров, 02–04 апреля 2019 года. – 2019. – С. 97–98.*
8. *L. Ducas, D. Micciancio, FHEW: bootstrapping homomorphic encryption in less than a second, in EUROCRYPT. LNCS, vol. 9056 (Springer, 2015), pp. 617–640.*
9. *Coron J., Mandal A., Naccache D., Tibouchi M. Fully Homomorphic Encryption over the Integers with Shorter Public Keys // Advances in Cryptology – CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011, Proceedings / P. Rogaway – Springer Science+Business Media, 2011. – P. 487–504.*
10. *Буртыка Ф. Б. Пакетное симметричное полностью гомоморфное шифрование на основе матричных полиномов // Труды Института системного программирования РАН. – 2014. – Т. 26. – № 5. – С. 99–116.*
11. *Бабенко Л. К., Буртыка Ф. Б., Макаревич О.Б., Трепачева А.В. Методы полностью гомоморфного шифрования на основе матричных полиномов // Вопросы кибербезопасности, – 2015. – №1. – С. 17–20.*
12. *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.*
13. *Трепачева А. В. Криптоанализ симметричных полностью гомоморфных линейных криптосистем на основе задачи факторизации чисел // Известия ЮФУ. Техническиенауки. – 2015. – № 5 (166). – С. 89–102.*
14. *S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhat-tacharya, «A review of homomorphic encryption libraries for secure computation», arXiv preprint arXiv:1812.02428, 2018.*
15. *Бабенко Л. К., Русаловский И. Д. Гомоморфная реализация метода Гаусса // Вопросы кибербезопасности. – 2023. – № 4(56). – С. 33–40.*
16. *Бабенко Л. К., Русаловский И. Д. Метод реализации гомоморфного деления // Известия ЮФУ. Технические науки. – 2020. – № 4(214). – С. 212–221.*
17. *Русаловский И. Д., Бабенко Л. К., Макаревич О. Б. Разработка методов гомоморфного деления // Известия ЮФУ. Технические науки. – 2022. – № 4(228). – С. 103–112.*
18. *Liudmila Babenko, Ilya Rusalovsky Homomorphic operations on integers via operations on bits // Proceedings – 2022 15th international conference on security of information and networks, sin 2022. – 2022.*
19. *Бабенко Л. К., Русаловский И. Д. Побитовые гомоморфные операции над числами с плавающей точкой // Известия ЮФУ. Технические науки. – 2023. – 4(234). – С. 26–35.*