# ГОМОМОРФНАЯ РЕАЛИЗАЦИЯ МЕТОДА ГАУССА

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

***Цель работы:*** *повышение безопасности обработки данных в облачных сервисах посредством разработки и применения методов и алгоритмов гомоморфного шифрования.*

***Метод исследования:*** *анализ возможной реализации метода Гаусса с использованием гомоморфного шифрования, анализ существующих проблем выполнения гомоморфной реализации метода Гаусса.*

***Результаты исследования:*** *проведен анализ возможности выполнения гомоморфной реализации решения системы линейных алгебраических уравнений (СЛАУ) методом Гаусса; отмечены проблемы, возникающие ввиду специфики задачи – обрабатываемые данные зашифрованы и управляющий алгоритм не имеет к ним доступа; предложено решение отмеченных проблем; предложен алгоритм гомоморфной реализации метода Гаусса, позволяющий выполнить решение СЛАУ над зашифрованными гомоморфно данными и получить гомоморфно зашифрованный результат, содержащий численное решение СЛАУ, а также бит ошибки, указывающий на несовместность системы или на бесконечное число решений; выполнен анализ и предложен наилучший вариант представления шифротекста, позволяющий избежать роста размерности шифротекста при решении СЛАУ, содержащих большое число неизвестных; выполнен анализ предложенной реализации и рассмотрены возможные улучшения, повышающие скорость обработки данных.*

***Научная новизна:*** *предложен алгоритм гомоморфной реализации метода Гаусса для решения СЛАУ, который может использоваться в облачных сервисах для безопасной обработки данных. Алгоритм может быть использован для решения СЛАУ в чистом виде, либо как шаг другого алгоритма.*

***Ключевые слова:*** *информационная безопасность, криптографическая защита, гомоморфная криптография, безопасные вычисления, облачные вычисления, методы и алгоритмы.*

**DOI:**10.21681/2311-3456-2023-4-33-40

## Введение

Криптографическая защита данных долгое время эффективно используется для обеспечения конфиденциальности данных во время их передачи по незащищенным каналам связи. Не так давно зародилось новое направление криптографии – гомоморфная криптография [1-10]. 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:

$$m_1 \, op \, m_2 = D\left(M\left(E\left(m_1\right), E\left(m_2\right)\right)\right) \quad (1)$$

Из данного выражения следует основная особенность гомоморфной криптографии – возможность выполнять над зашифрованными данными некоторые операции таким образом, что результат операций над зашифрованными данными после расшифровки будет эквивалентен результату соответствующей операции над открытыми данными. Таким образом применение гомоморфной криптографии позволяет обрабатывать данные без их предварительной расшифровки, благодаря чему 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 [11-15] гарантирует, что данные не будет перехвачены даже в случае подмены сервера, т.к. они остаются зашифрованными на протяжении всего

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.

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. Одной из таких задач в вычислительной алгебре является задача решения систем линейных алгебраических уравнений (СЛАУ). Несмотря на то, что на практике редко встречается задача решения СЛАУ в чистом виде, значительная часть численных методов решения различных задач включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма. Выполнение гомоморфной реализации для процесса решения СЛАУ позволит в дальнейшем разработать гомоморфную реализацию более сложных численных методов решения задач.

Для нахождения решения СЛАУ применяются различные методы, одним из наиболее известных и часто применяемых является метод Гаусса. Метод Гаусса – это классический метод решения СЛАУ, в ходе которого последовательно исключаются переменные при помощи элементарных преобразований, и система приводится к треугольному виду (прямой ход метода Гаусса), а затем из полученной системы последовательно находятся все переменные (обратный ход метода Гаусса). Также прямой ход метода Гаусса применяется для вычисления определителя матриц высших порядков, что в свою очередь используется при исследовании СЛАУ на совместность. К основным преимуществам метода Гаусса можно отнести:

- отсутствие необходимости исследовать систему на совместность
- возможность применения к системам, в которых число уравнений не равно числу неизвестных, а основная матрица системы вырожденная
- результативность при сравнительно небольшом числе операций
- возможность получить результат при фиксированном числе операций

Ввиду перечисленных преимуществ, предлагается рассмотреть метод Гаусса для решения СЛАУ и выполнить его гомоморфную реализаци. Начальные исследования в данном направлении уже были выполнены и опубликованы[3], однако решить все проблемы, возникшие при выполнении гомоморфной реализации, на тот момент не удалось из-за недостаточной проработанности методов гомоморфной арифметики. Впоследствии были проведены дополнительные исследования и разработаны методы и средства для гомоморфной арифметики, что позволяет решить все проблемы и выполнить гомоморфную реализацию алгоритма Гаусса.

### Гомоморфная арифметика

Гомоморфные криптографические алгоритмы подразделяются на алгоритмы над целыми числами и над битами, в зависимости от того, какой тип данных обрабатывается. Алгоритмы над битами поддерживают только логические операции, а алгоритмы над целыми – только арифметические. Многие прикладные алгоритмы обработки данных требуют поддержки как арифметических, так и логических операций. Однако существующие алгоритмы поддерживают только узкий набор операций (либо логические, либо арифметические), поэтому была выполнена реализация алгоритма, который позволяет выполнять и логические и арифметические операции над зашифрованными данными в рамках одной криптосистемы.

В рамках разработанных алгоритмов шифруемые числа представляются в виде массива битов, каждый бит поочередно шифруется с помощью некоторого полностью гомоморфного алгоритма шифрования над битами, а полученный массив зашифрованных битов является шифротекстом. Благодаря тому, что используется гомоморфный алгоритм шифрования над битами, мы изначально получаем доступ к логическим операциям, а на их основе выполняем реализацию арифметических операций над целыми [16-17].

В приведенных выше статьях рассматриваются операции над целыми числами, однако алгоритм можно легко преобразовать для поддержки дробных чисел. Для поддержки n знаков после запятой необходимо предварительно умножить шифротексты на $10^n$, сложение и разность выполнять как и раньше, после умножения дополнительно разделить результат на $10^n$, а при делении предварительно умножить делимое на $10^n$. В будущих статьях планируется рассмотреть возможность реализации алгоритма над дробными

3 Русаловский И.Д. Гомоморфная реализация алгоритма Гаусса // Сборник статей IV Всероссийской научно-технической конференции молодых ученых, аспирантов и студентов «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности». 2014. С. 364-367.

числами, с представлением шифротекста в виде числа с плавающей точкой, что позволит увеличить размерность обрабатываемых чисел и точность вычислений.

### Анализ возможности выполнения гомоморфной реализация алгоритма Гаусса

Для выполнения гомоморфной реализации алгоритма Гаусса представим данный алгоритм над открытыми данными в вербальном виде. Проанализируем каждый шаг полученного алгоритма и внесем изменения, которые потребуются при его гомоморфной реализации над зашифрованными данными.

Пусть дано:

- коэффициенты СЛАУ представлены в виде двумерного массива (матрицы системы) A размерности mxn, где m – число уравнений, n – число неизвестных;
- свободные члены заданы в виде массива (матрицы-столбца) B размерности m (соответствует числу уравнений в системе);
- бит e (бит ошибки), указывающий на несовместность системы или отсутствие численных решений (бесконечное количество решений).

Тогда для решения СЛАУ методом Гаусса необходимо выполнить следующие шаги:

Предварительное исследование матрицы:

*Если число переменных больше числа уравнений, то можно сразу сделать вывод о бесконечном числе решений в случае совместности системы или об отсутствии решений в обратном случае, устанавливаем бит ошибки e = 1, завершаем алгоритм.*

Так как шифруются только сами коэффициенты, индексы при этом сохраняются, поэтому данный этап при обработке гомоморфно зашифрованных данных будет аналогичным.

Прямой ход (приведение матрицы к треугольному виду):

1. Пусть i = 1

Так как шифруются только сами коэффициенты, индексы при этом сохраняются, поэтому данный этап при обработке гомоморфно зашифрованных данных будет аналогичным.

2. Переставить строки [i; m] так, чтобы в позиции $a_{ii}$ был ненулевой элемент. Если подходящего уравнения нет, то система не имеет численных решений, устанавливаем бит e = 1, завершаем алгоритм.

Когда мы выполняем данную операцию над открытыми данными, она проста и тривиальна, однако при обработке зашифрованных данных возникает проблема - мы можем получить результат сравнения шифротекста с нулем, однако это будет зашифрованный бит, к значению которого управляющий алгоритм не может получить доступ. Поэтому, чтобы быть уверенными, что, при наличии хотя бы одного подходящего уравнения, в позиции $a_{ii}$ будет ненулевой элемент, необходимо выполнить поочередную попарную перестановку всех уравнений. Эта операция будет вычислительно сложной, однако позволит выполнить весь алгоритм над гомоморфно зашифрованными данными, без их частичной расшифровки и дополнительного клиент-серверного взаимодействия.

Если подходящего уравнения нет, то управляющий алгоритм не сможет это определить, ведь данные зашифрованы. Поэтому выполнение поиска решения продолжится, но это не является проблемой. Факт того, что система имеет бесконечное число решений, выяснится во время выполнения обратного хода метода Гаусса, так как будет обнаружена попытка деления на 0. Если значение соответствующего свободного члена будет также равно 0, значит система имеет бесконечное число решений, иначе - несовместна.

3. Обнулить коэффициенты столбца i в строках [i+1; m]

Для реализации данного шага необходима поддержка гомоморфных операций умножения и разности, которые поддерживаются предложенным нами алгоритмом. Однако на этом шаге возникает другая проблема - необходимость поддержки слишком большой разрядной сетки или вероятность переполнения разрядной сетки. Шифротексты представлены в виде массивов зашифрованных битов, поэтому у нас есть два варианта реализации:

- Задать фиксированную размерность шифротекста (максимальное количество бит), например 64 бита
- Задать минимальную размерность шифротекста (например, 8 бит) и увеличивать ее на 1 бит после каждой операции сложения/умножения и в 2 раза после каждой операции умножения.

Сравнение этих решений будет приведено ниже.

4. Увеличить i на 1, если $i \geq n$ – перейти к пункту 5, иначе вернуться к пункту 2 и повторить шаги для нового значения i.

Так как шифруются только сами коэффициенты, индексы при этом сохраняются, поэтому данный этап при обработке гомоморфно зашифрованных данных не требует адаптаций.

5. Если число уравнений больше числа переменных, обнуляем их коэффициенты и проверяем, что уравнения [i; m] имеют смысл (имеют вид 0=0), иначе

система несовместна и не имеет решений, завершаем алгоритм.

Для выполнения данной операции необходим алгоритм гомоморфного сравнения чисел. Для каждого уравнения [i; m] необходимо проверить, что значение свободного члена равно нулю. Если хоть один из них отличен от нуля, необходимо установить флаг e = 1. Предварительно завершить алгоритм мы не можем ввиду того, что данные зашифрованы и управляющий алгоритм не может определить, что дальнейшее решение не имеет смысла.

Обратный ход:

1. Пусть i = m
2. Вычислим $x_i$

$$x_i = \left( b_i - \sum_{j=i+1}^{m} a_{ij} \times x_j \right) / a_{ii} \qquad (2)$$

3. Уменьшить i на 1, если i < 1 – завершить обратный ход, все коэффициенты найдены, иначе вернуться к шагу 2.

Гомоморфная реализация обратного хода алгоритма Гаусса требует поддержки гомоморфных операций умножения, разности и деления. Данные операции поддерживаются в нашем алгоритме операций над целыми через операции над битами.

Также в ходе обратного хода необходимо проверять, что не происходит деления на 0. Если деление на 0 выполняется, значит система несовместна, либо имеет бесконечное количество решений. Нам необходимо сравнивать каждый делитель $a_{ii}$ с нулем и записывать результат в бит ошибки e.

### Представление и обработка шифротекстов

Как было рассмотрено ранее, мы можем задать фиксированную размерность шифротекстов или увеличивать размерность шифротекста после каждой гомоморфной операции. Оба решения имеют недостатки в случае, когда система имеет большое число неизвестных. Рассмотрим их подробнее:

— Фиксированная размерность. В данном случае возникает вероятность переполнения разрядной сетки. Предположим, что все коэффициенты уравнения - 8-ми битные числа, а размерность шифротекста – 64 бит. В этом случае мы достигнем переполнения уже после 4-го перемножения коэффициентов. Решить данную проблему можно нормализацией коэффициентов. Для этого предлагается после каждой операции умножения делить все коэффициенты уравнения на $a_{ii}$ коэффициент, если он больше единицы, либо на 1 в противном случае. Это решит проблему переполнения, однако увеличит сложность обработки данных и может снизить точность вычислений в случае, если соотношение коэффициентов слишком велико. Но проблема с точностью будет минимальной, в случае использования алгоритма над числами с плавающей точкой.

— Увеличение размерности. В данном случае возникает рост размерной сетки шифротекста после каждой гомоморфной операции. В ходе прямого хода умножение каждого коэффициента выполняется (n-1) число раз, в ходе обратного хода умножение выполняется 1 раз, следовательно размерность шифротекста вырастет до x*n, где x – начальная размерность шифротекста в битах. Данный подход обеспечит более высокую точность, так как реже будет выполняться операция с максимальной погрешностью вычислений – операция деления, однако увеличит потребление памяти и повысит сложность вычислений.

Так как гомоморфные операции вычислительно сложные и любой из подходов увеличивает сложность алгоритма, необходимо сравнить их в данном аспекте. Так как логические операции в предложенной реализации над битами куда менее сложны вычислительно, чем арифметические операции, мы их не учитываем в данной оценке. Для упрощения также будем считать все арифметические операции приблизительно равными по сложности. В этом случае нам необходимо выполнить на каждой итерации i:

— $2(n-i)(m-i)$ операций умножения (прямой ход)
— $(n-i)(m-i)$ операций разности (прямой ход)
— $(n-i)(m-i)$ операций деления (нормализация)
— $2(n-i)$ операций разности (обратный ход)
— 1 операция деления (обратный ход)

Тогда сложность алгоритма при использовании нормализации можно приблизительно выразить как:

$$O(n) = \sum_{i=1}^{n} (n-i)(4m-4i+2)+1 \qquad (3)$$

В случае увеличения размерности коэффициентов в алгоритме не будет дополнительного шага, однако к сложности всех шагов алгоритма необходимо будет ввести дополнительный коэффициент. В зависимости

от того, какая начальная размерность шифротекста, значение коэффициента может быть меньше единицы для первых итераций прямого хода. Пусть x – начальная размерность шифротекста в алгоритме без нормализации, y – размерность шифротекста в алгоритме с нормализацией, тогда сложность операций можно представить в виде:

– $2(n-i)(m-i)\left(x\times 2^{i-1}/y\right)$ операций умножения (прямой ход)
– $(n-i)(m-i)\left(x\times 2^{i}/y\right)$ операций разности (прямой ход)
– $(n-i)(m-i)\left(x\times 2^{n-1}/y\right)$ операций разности (обратный ход)
– $1\left(x\times 2^{n-1}/y\right)$ операций деления (обратный ход)

Тогда сложность алгоритма без использования нормализации можно приблизительно выразить как:

$$O(n)=\sum_{i=1}^{n}(n-i)(m-i)\frac{\left(2^{i+1}+2^{n-1}\right)x}{y}+\frac{2^{n-1}x}{y} \quad (4)$$

При размерности системы n = 5 и более вариант реализации с нормализацией коэффициентов показывает более низкие значения сложности вычислений, что означает лучшую производительность. К тому же для случая, когда алгоритм решения СЛАУ методом Гаусса над гомоморфно зашифрованными данными является только одним из шагов обработки данных, излишнее разрастание коэффициентов нежелательно. Поэтому для реализации выбран вариант с нормализацией коэффициентов.

## Алгоритма Гаусса над зашифрованными гомоморфно данными

В прошлом пункте были рассмотрены адаптации, необходимые для выполнения гомоморфной реализации алгоритма Гаусса. Просуммируем их и сформулируем алгоритм.

Пусть дано:

— некоторый полностью гомоморфный алгоритм шифрования над битами
— некоторая гомоморфная криптосистема, шифротекст которой представлен в виде массива гомоморфно зашифрованных битов, поддерживающая следующие операции над шифротекстами:
  - Enc(x) – шифрование бита x
  - сложение
  - разность
  - умножение
  - деление
  - логическое И (между битом и шифротекстом)
  - логическое ИЛИ (между битом и шифротестом)
  - >, < – операторы сравнения двух шифротекстов на больше/меньше, результатом которого является зашифрованный бит
  - Eq(x, y) – сравнение двух зашифрованных битов или шифротекстов, результатом которого является зашифрованный бит
— СЛАУ
  - коэффициенты которой являются шифротекстами гомоморфной криптосистемы и представлены в виде двумерного массива A размерности m-свободные члены которой являются шифротекстами гомоморфной криптосистемы и заданы в виде массива B размерности m.
  - бит e (бит ошибки), зашифрованный с помощью данного полностью гомоморфного алгоритма и указывающий на несовместность системы / отсутствие численных решений

Тогда алгоритм Гаусса для решения гомоморфно зашифрованной СЛАУ будет иметь следующий вид.

Предварительное исследование матрицы:

Если число переменных больше числа уравнений, то можно сразу сделать вывод о бесконечном числе решений в случае совместности системы или об отсутствии решений в обратном случае, устанавливаем бит e = Enc(1), завершаем алгоритм.

Прямой ход (приведение матрицы к треугольному виду):

1. Пусть i = 1

2. Переставить строки [i; m] так, чтобы в позиции $a_{ii}$ был ненулевой элемент. Для каждого j из [i+1; m]:

2.1. выполняем проверку на ноль:

$$c_i = Eq\left(a_{ii}, Enc(0)\right) \quad (5)$$

2.2. для каждого k из [1; n] выполняем

$$a_{ik} = (a_{ik}\wedge c_i)\vee\left(a_{jk}\wedge\overline{c_i}\right) \quad (6)$$

$$a_{jk} = (a_{jk}\wedge c_i)\vee\left(a_{ik}\wedge\overline{c_i}\right) \quad (7)$$

То есть, если $a_{ii}$ не равен нулю, то оставляем строки без изменения, иначе меняем их местами. Выполнив перестановку каждой пары уравнений, мы будем уверены, что если в системе есть хоть одно уравнение с ненулевым элементом в столбце i, оно будет поставлено в строку i.

3. Обнулить коэффициенты столбца i в строках [i+1; m]. Для этого:

3.1 для каждых k из [i; n] и l из [i+1; m] вычисляем

$$a_{lk} = a_{lk} \times a_{ii} - a_{ik} \times a_{il} \quad (8)$$

3.2 для каждого l из [i+1; m] вычисляем

$$b_l = b_l \times a_{ii} - b_i \times a_{il} \quad (9)$$

4. Нормализовать коэффициенты, чтобы избежать переполнения. Для этого для j = i+1 и для каждых k из [i+1; n], l из [i+1; m] вычислим:

$$c_l = a_{lj} > Enc(1) \quad (10)$$

$$d_l = \left(a_{lj} \wedge c\right) \vee \left(Enc(1) \wedge \bar{c}\right) \quad (11)$$

$$a_{lk} = a_{lk} / d_l \quad (12)$$

$$b_l = b_l / d_l \quad (13)$$

5. Увеличить i на 1, если $i \geq n$ — перейти к пункту 6, иначе вернуться к пункту 2 и повторить шаги для нового значения i.

6. Если число уравнений больше числа переменных, обнулить их коэффициенты и проверить, что уравнения [i; m] имеют смысл (имеют вид 0=0), иначе система несовместна и не имеет решений, завершаем алгоритм. Так как левая часть была приведена к 0 на шаге 3, нам достаточно проверить, что правая часть каждого уравнения также равна нулю. Для этого для каждого k из [i; m] вычислим

$$c_k = Eq\left(a_{kn}, Enc(0)\right) \quad (14)$$

$$e = c_i \vee \ldots \vee c_m \quad (15)$$

Обратный ход:

1. Пусть i = n

2. Вычислим $x_i$

$$x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij} \times x_j\right) / a_{ii} \quad (16)$$

3. Уменьшить i на 1, если i < 1 – завершить обратный ход, все коэффициенты найдены, иначе вернуться к шагу 2.

Результатом выполнения алгоритма Гаусса служит массив коэффициентов X размерности n, а также бит ошибки e. После расшифровки сперва необходимо проверить бит ошибки, если он равен 1, значит в процессе решения было обнаружено, что система несовместна и не имеет решений, либо имеет бесконечное число решений, а значит массив коэффициентов содержит случайные или ошибочные числа. В обратном случае массив X содержит решение системы.

## Параллельные вычисления

Ввиду побитного представления чисел, многие логические операции, выполняемые побитно между двумя шифротекстами, можно выполнить параллельно, что позволит увеличить скорость выполнения предложенного алгоритма.

Также многие операции в ходе выполнения алгоритма Гаусса могут быть выполнены параллельно друг от друга. Например:

- Обнуление коэффициентов ниже главной диагонали. Во время данной операции поочередно обрабатываются пары уравнений, при этом результаты обработки отдельных пар никак не используются при обработке других пар (операции выполняются независимо). Поэтому обработка каждой пары уравнений может быть выполнена параллельно.
- Проверка деления на 0. Данная операция выполняется каждую итерацию, однако можно выполнить проверку сразу для всех элементов главной диагонали параллельно и потом обработать полученные результаты.

## Выводы

В статье рассмотрена проблема решения СЛАУ с применением гомоморфной криптографии методом Гаусса. Проведен анализ проблем, возникающих при обработке зашифрованных данных и предложено решение для каждой из них. Предложен алгоритм гомоморфной реализации метода Гаусса, позволяющий обработать гомоморфно зашифрованные коэффициенты СЛАУ без их предварительной расшифровки и вернуть зашифрованное решение системы в виде бита ошибки, указывающего на бесконечное количество решений системы или несовместность системы, и массива значений неизвестных. Проведен анализ предложенного алгоритма и приведены возможные адаптации, позволяющие повысить скорость обработки данных. Предложенная гомоморфная реализация метода Гаусса может использоваться в облачных сервисах для защищенных облачных вычислений, чтобы находить решение СЛАУ.

Возможные направления дальнейших исследований:

- рассмотреть возможность реализации арифметики над числами с плавающей точкой через операции над гомоморфно зашифрованными битами
- рассмотреть другие алгоритмы для решения СЛАУ и сравнить их быстродействие и вычислительную сложность для систем разной размерности

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

**Литература**


1. Бабенко М.Г., Голимблевская Е.И, Ширяев Е.М. Сравнительный анализ алгоритмов гомоморфного шифрования на основе обучения с ошибками // Труды 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: Материалы 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: Сборник 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. Минаков С.С. Основные криптографические механизмы защиты данных, передаваемых в облачные сервисы и сети хранения данных // Вопросы кибербезопасности. – 2020. – № 3(37). – С. 66-75.
12. Дерябин М. А., Кучеров Н. Н. Обзор безопасных методов шифрования для облачных вычислений // Новости науки в АПК. – 2019. – № 3(12). – С. 298-303.
13. 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.
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 М. А. Обзор безопасных методов шифрования для облачных вычислений // Новости науки в АПК. – 2019. – № 3(12). – С. 298-303.
16. Русаловский И. Д., Бабенко Л.К., Макаревич О.Б. Разработка методов гомоморфного деления // Известия ЮФУ. Технические науки. – 2022. – № 4(228). – С. 212-221.
17. 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.


# HOMOMORPHIC REALIZATION OF THE GAUSS ELIMINATION METHOD

***Babenko L.K.[4], Rusalovsky I.D.[5]***

***Purpose of the work:*** *improving the security of data processing in cloud services through the development and application of methods and algorithms for homomorphic encryption.*

***Research methods:*** *analysis of a possible implementation of the Gaussian elimination method using homomorphic encryption, analysis of existing problems in implementing a homomorphic implementation for the Gaussian elimination method.*

***Research results:*** *analysis of the possibility of performing a homomorphic implementation of the solution of a system of linear algebraic equations (SLAE) by the Gaussian elimination method was performed; problems that*

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

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

*arise due to the specifics of the task are noted - the processed data is encrypted and the control algorithm does not have access to it; the solution of the noted problems is proposed; an algorithm for the homomorphic implementation of the Gaussian elimination method is proposed, which allows solving the SLAE over homomorphically encrypted data and obtaining a homomorphically encrypted result containing the numerical solution of the SLAE, as well as an error bit indicating the incompatibility of the system or an infinite number of solutions; the analysis is carried out and the best variant of the ciphertext representation is proposed, which allows avoiding the growth of the ciphertext when solving SLAE containing a large number of variables; the analysis of the proposed implementation is carried out and possible improvements that increase the speed of data processing are considered.*

***Scientific novelty:*** *an algorithm for the homomorphic implementation of the Gaussian elimination method for solving SLAE is proposed, which can be used in cloud services for secure data processing. The algorithm can be used to solve SLAE in its pure form, or as a step of another algorithm.*

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

**References**

1. Babenko, M. G., Golimblevskaia E. I., Shiriaev E. M. Comparative Analysis of Homomorphic Encryption Algorithms Based on Learning with Errors // Proceedings of the Institute for System Programming of the RAS. – 2020. – Vol. 32, No. 2. – P. 37-52.
2. L. Babenko, I. Rusalovsky Biblioteka polnost'yu gomomorfnogo shifrovaniya celyh chisel // Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki [Proceedings of Southern Federal University. Engineering sciences]. – 2020. – №2. – pp. 79-88.
3. L. Babenko, I. Rusalovsky Metod realizacii gomomorfnogo deleniya chisel // Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki [Proceedings of Southern Federal University. Engineering sciences]. – 2020. – №4. – pp. 212-221.
4. L. Babenko, A. Trepacheva O nestojkosti dvuh simmetrichnyh gomomorfnyh kriptosistem, osnovannyh na sisteme ostatochnyh klassov // Trudy Instituta sistemnogo programmirovaniya RAN. – 2019. – Vol. 18, № 1. – pp. 230-262.
5. Arakelov G.G. Voprosy primeneniya prikladnoj gomomorfnoj kriptografii // Voprosy kiberbezopasnosti [Cybersecurity issues]. – 2019. – № 5(33). – pp. 70-74.
6. SHachina, V. A. Gomomorfnaya kriptografiya v bazah dannyh // Prikladnaya matematika i informatika: sovremennye issledovaniya v oblasti estestvennyh i tekhnicheskih nauk: Materialy V Mezhdunarodnoj nauchno-prakticheskoj konferencii (shkoly-seminara) molodyh uchenyh, Tol'yatti, 22–24 aprelya 2019 goda. – 2019. – pp. 468-473.
7. Trusova YU. O., Vovk N. N., Anisimov YU. A. Uvelichenie skorosti gomomorfnogo shifrovaniya na osnove kriptosistemy El'-Gamalya // Matematika i matematicheskoe modelirovanie: Sbornik materialov XIII Vserossijskoj molodezhnoj nauchno-innovacionnoj shkoly, Sarov, 02–04 aprelya 2019 goda. – 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 // International Journal of Open Information Technologies. – 2021. – Vol. 9, № 3. – pp. 11-22.
9. Volyanskij YU. Usovershenstvovanie sistemy poiska opasnyh slov s ispol'zovaniem gomomorfnogo shifrovaniya // Innovacii. Nauka. Obrazovanie. – 2021. – № 38. – pp. 687-695.
10. Arakelov G. G., Mihalev A. V. Kombinaciya chastichno gomomorfnyh skhem // Elektronnye informacionnye sistemy. – 2020. – № 3(26). – pp. 83-92.
11. Minakov S.S. Osnovnye kriptograficheskie mexanizmy zashhity dannyx, peredavaemyx v oblachnye servisy i seti xraneniya dannyx // Voprosy kiberbezopasnosti [Cybersecurity issues]. – 2020. – № 3(37). – pp. 66-75.
12. Deryabin M. A., Kucherov N. N. Obzor bezopasnyx metodov shifrovaniya dlya oblachnyx vychislenij // Novosti nauki v APK. – 2019. – № 3(12). – pp. 298-303.
13. Babenko L. K., SHumilin A. S., Alekseev D. M. Algoritm obespecheniya zashchity konfidencial'nyh dannyh oblachnoj medicinskoj informacionnoj sistemy // Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki [Proceedings of Southern Federal University. Engineering sciences]. – 2021. – № 5(222). – pp. 120-134.
14. Minakov S. S. Osnovnye kriptograficheskie mekhanizmy zashchity dannyh, peredavaemyh v oblachnye servisy i seti hraneniya dannyh // Voprosy kiberbezopasnosti [Cybersecurity issues]. – 2020. – № 3(37). – pp. 66-75.
15. Deryabin M. A., Kucherov M. A. Obzor bezopasnyh metodov shifrovaniya dlya oblachnyh vychislenij // Novosti nauki v APK. – 2019. – № 3(12). – pp. 298-303.
16. Rusalovskij I. D., Babenko L.K., Makarevich O.B. Razrabotka metodov gomomorfnogo deleniya // Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki [Proceedings of Southern Federal University. Engineering sciences]. – 2022. – № 4(228). – pp. 212-221.
17. 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.