МАСШТАБИРОВАНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ С ПРИМЕНЕНИЕМ ГОМОМОРФНОГО ШИФРОВАНИЯ

Бабенко Л.К.[1], Русаловский И.Д.[2]

Аннотация.

Криптография с незапамятных времен обеспечивает безопасную передачу информации в небезопасной среде, сохраняя пересылаемые данные в секрете. Не так давно начало активно развиваться направление гомоморфной криптографии. Его отличительной особенностью является то, что данный вид криптографии позволяет обрабатывать зашифрованные данные без их предварительной расшифровки таким образом, что результат операций над зашифрованными данными эквивалентен после расшифровки результату операции над открытыми данными. Благодаря этим особенностям гомоморфное шифрование может эффективно использоваться в различных облачных сервисах для выполнения безопасных вычислений и безопасной обработки изображений. При этом гарантируется, что открытых данных не будет ни у кого, даже у сервиса, который выполняет вычисления.

Цель работы: *разработка методов и средств гомоморфного шифрования, позволяющих выполнить гомоморфную реализацию алгоритмов масштабирования изображений.*

Метод исследования: *анализ возможных реализаций обработки цифровых изображений с использованием гомоморфного шифрования, анализ существующих проблем выполнения гомоморфной реализации для алгоритмов обработки изображений.*

Результаты: *предложен метод гомоморфного сравнения битов и чисел, представленных в виде массива битов; выполнен анализ применимости гомоморфного шифрования к методу «ближайшего соседа» для масштабирования изображения; предложена гомоморфная реализация алгоритма увеличения изображения EPX; выполнен анализ сложности выполнения операции при увеличении одного пикселя исходного изображения при использовании предложенного метода, приведены результаты анализа.*

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

DOI: 10.21681/2311-3456-2021-3-2-10

Введение

На протяжении веков криптография помогает обеспечивать безопасную передачу данных в небезопасной среде, сохраняя пересылаемые данные в секрете. В настоящее время активно развивается направление гомоморфной криптографии. Впервые понятие гомоморфной криптографии было сформировано в 1978 году[3]. Однако дальнейшее развитие это направление получило только в 2009 году, когда Крейг Джентри теоретически доказал возможность создания полностью гомоморфных криптосистем и предложил свой вариант реализации такой криптосистемы[4].

Отличительная особенность гомоморфной криптографии – возможность обрабатывать зашифрованные данные без их предварительной расшифровки. В рамках гомоморфной криптографии можно выполнять над зашифрованными данными некоторые операции таким образом, что результат операций над зашифрованными данными после расшифровки будет эквивалентен результату соответствующей операции над открытыми 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:

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

---

1 Бабенко Людмила Климентьевна, доктор технических наук, профессор, профессор кафедры БИТ Института компьютерных технологий и информационной безопасности Южного Федерального Университета «ЮФУ», г. Таганрог, Россия. E-mail: lkbabenko@sfedu.ru

2 Русаловский Илья Дмитриевич, аспирант кафедры БИТ Института компьютерных технологий и информационной безопасности, Южного Федерального Университета «ЮФУ», г. Таганрог, Россия. E-mail: ilya.rusalovskiy@mail.ru

3 R.L. Rivest, L. Adleman, M.L. Dertouzos. On data banks and privacy homomorphisms // Foundations of secure computation. – 1978. – Vol. 32, no. 4. – pp. 169–178.

4 C. Gentry, A fully homomorphic encryption scheme // A dissertation submitted to the department of computer science and the committee on graduate students of Standford University. – 2009.

Обычно рассматривается гомоморфизм относительно операций сложения и умножения. Система шифрования является гомоморфной относительно операции сложения, если

$$m_1 + m_2 = D\left(E\left(m_1\right) \oplus E\left(m_2\right)\right) \quad (2)$$

Система шифрования гомоморфна относительно операции умножения, если

$$m_1 * m_2 = D\left(E\left(m_1\right) \otimes E\left(m_2\right)\right) \quad (3)$$

где $\oplus, \otimes$ – операции сложения и умножения над зашифрованными данными, которые соответствуют операциям сложения и умножения над открытыми данными; D – функция расшифрования; E – функция шифрования.

Гомоморфные криптосистемы подразделяются на частично и полностью гомоморфные. Частично гомоморфные криптосистемы проявляют гомоморфизм только относительно одного оператора – сложения или умножения. Полностью гомоморфные криптосистемы проявляют гомоморфизм относительно и сложения, и умножения. В настоящее время ведутся активные разработки в области гомоморфной криптографии [1-12].

Благодаря вышеописанным особенностям гомоморфное шифрование может эффективно использоваться в следующих сферах:

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

Применение гомоморфного шифрования в облачных сервисах[13-17] гарантирует, что данные не будут 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. В данной статье будет рассмотрен вопрос безопасной обработки изображений с использованием облачных сервисов, а частности, методов масштабирования изображений.

Облачная обработка изображений

В настоящее время популярны облачные сервисы удаленной обработки изображений. Они позволяют выполнять обработку изображений на удаленном сервере. При этом все вычисления выполняются на самом сервисе, и пользователю не требуется устанавливать специальные программы. Однако, пользователю требуется отправить изображение на удаленный сервер для обработки. В случае подмены или неблагонадежности сервера изображение может быть передано третьим лицам, что вызывает недоверие к облачным сервисам. Для обеспечения конфиденциальности данных можно использовать гомоморфное шифрование. Это позволит обрабатывать изображение на облачном сервисе без раскрытия данных. Процесс обработки изображения на облачном сервисе с применением гомоморфного шифрования можно представить следующим образом (рис. 1).

В данном примере в качестве доверенной среды рассматривается некоторый персональный компьютер, локальная или корпоративная сеть или сегмент сети, в пределах которых пользователь может безопасно передавать и хранить данные. Как только данные покидают доверенную среду, они попадают в недоверенную среду (каналы передачи данных, облачные сервисы, INTERNET), они могут быть перехвачены злоумышленником. В данном примере не рассматриваются такие вопросы, как защита доверенной среды, защита целостности данных при попадании в недоверенную среду и прочие. Рассматриваются только проблемы обработки

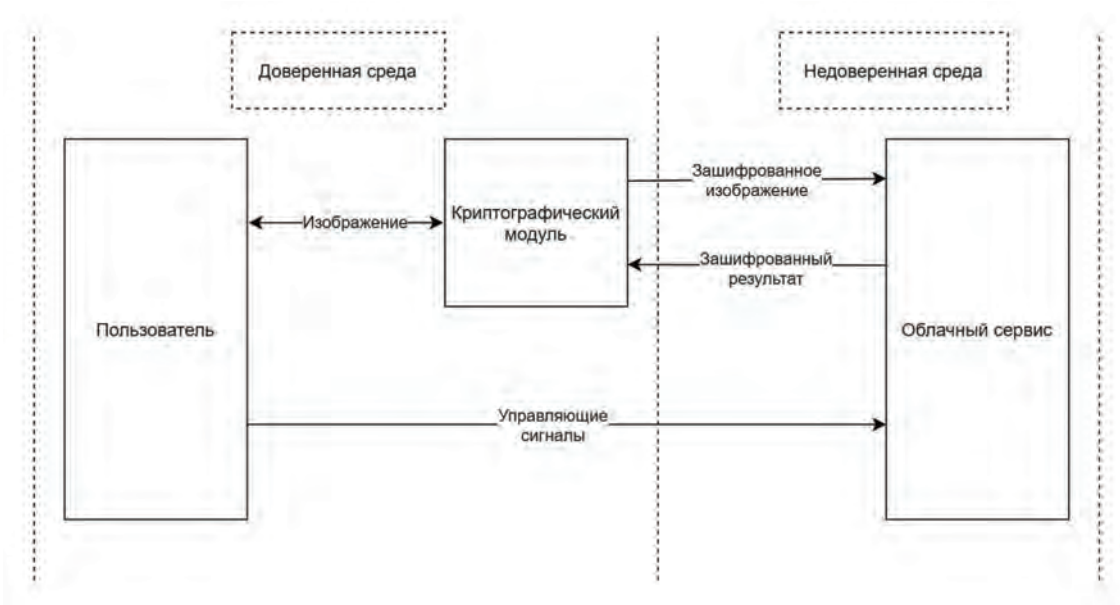

Рис.1. Схема облачной обработки изображения

DOI: 10.21681/2311-3456-2021-3-2-10

и защиты конфиденциальности данных. Процесс обработки изображения с использованием облачного сервиса может быть построен следующим образом:

1. Пользователь выбирает изображение для обработки
2. Изображение предварительно шифруется гомоморфно и отправляется на сервер облачного сервиса. Генерация ключей шифрования и шифрование выполняются на стороне пользователя в доверенной среде. Ключ расшифрования никуда не пересылается и хранится в доверенной среде. Ключ шифрования, в случае использования асимметричного шифрования, может передаваться на сервер облачного сервиса и использоваться для шифрования дополнительных данных, например, некоторых коэффициентов.
3. Пользователь отправляет сервису управляющие сигналы. Например, «Увеличить в 2 раза», «Инвертировать цвета» и тд.
4. Облачный сервис обрабатывает полученные сигналы. После выполнения каждой операции сервис возвращает зашифрованный результат назад, чтобы пользователь мог видеть изменения в режиме реального времени.

Криптографический модуль в предложенной схеме должен быть реализован посредством некоторого полностью гомоморфного алгоритма шифрования и обеспечивать генерацию и хранение ключей, а также операции шифрования и расшифрования данных. Доступ к открытым данным и ключу расшифрования будет только у пользователя, а передаваться по каналу связи и обрабатываться на облачном сервисе будут только зашифрованные гомоморфно данные. Облачный сервис должен поддерживать тот же самый гомоморфный алгоритм шифрования, что и криптографический модуль. Также на облачном сервисе должны быть реализованы гомоморфные математические операции и, при необходимости, функция шифрования. На базе этих гомоморфных операций должны быть выполнены гомоморфные реализации алгоритмов обработки цифровых изображений. Чем большее число различных операций над изображениями поддерживает облачный сервис, тем большую практическую ценность он имеет.

Обработка цифровых изображений

Цифровые изображения – это двумерные изображения, представленные в цифровом виде. Цифровые изображения, в зависимости от способа их представления, делятся на растровые и векторные. Растровые изображения представляют собой двумерный массив, каждый элемент которого соответствует одному минимальному элементу – пикселю. Векторные изображения создаются путем математического описания геометрических примитивов – точки, линии, окружности и так далее.

Обработка и редактирование цифровых изображений – это процесс изменения оригинального изображения для достижения определенных целей[5]. В основном редактирование изображений требуется для достижения таких целей, как устранение дефектов изображения, структурное редактирование изображения, подготовка изображения к отображению на цифровых носителях или к печати. Можно выделить следующие виды обработки изображений:

- редактирование яркости;
- редактирование контрастности;
- устранение нерезкости;
- замена и инверсия цветов;
- изменение размеров изображения;
- кадрирование;
- дорисовка, добавление надписей, символов;
- добавление спецэффектов, фильтров, теней.

В данной статье будут более подробно рассмотрены методы изменения размеров цифровых изображений и их возможная гомоморфная реализация. Методы масштабирования изображения позволяют получить изображение нового размера на основе исходного изображения. В настоящее время используются такие методы, как метод «ближайшего соседа», аффинные преобразования, свертки, EPX и прочие. Следует отметить, что все методы обеспечивают разное качество конечного изображения и имеют разную скорость работы. Обычно качество и производительность любого метода можно оценить по отношению числа пикселей, которые участвовали в формировании конечного изображения, к общему числу пикселей исходного изображения.

Увеличение изображения методом ближайшего соседа

Метод ближайшего соседа позволяет увеличить или уменьшить растровое изображение на любой коэффициент. При этом новые размеры изображения будут округлены до целого числа пикселей. Алгоритм состоит из двух этапов. Сначала вычисляется размер нового изображения:

$$\begin{aligned} width' &= width \times scale_w \\ height' &= height \times scale_h \end{aligned} \qquad (4)$$

где $width, height$ – ширина и высота исходного изображения, $width', height'$ – ширина и высота результирующего изображения, $scale_w, scale_h$ – коэффициенты увеличения изображения по горизонтали и вертикали соответственно; в случае увеличения с сохранением пропорций эти коэффициенты будут равны.

После этого каждый пиксель нового изображения заполняется «ближайшим» пикселем исходного изображения (рис. 2). Координаты «ближайшего» пикселя для подстановки вычисляются по формулам:

$$x = \frac{(x'-1)\times(width-1)}{width'-1} + 1 \qquad (5)$$

$$y = \frac{(y'-1)\times(height-1)}{height'-1} + 1 \qquad (6)$$

5 Потапов А. А., Пахомов А. А., Никитин С. А., Гуляев Ю. В., Новейшие методы обработки изображений. — М.: Физматлит, 2008. — 496 с.

где x, y – координаты пикселя исходного изображения, который будет подставлен в результирующее изображение; x', y' – координаты результирующего пикселя, для которого выполняется поиск «ближайшего» исходного.

Таким образом, для всех пикселей результирующего изображения (x';y') вычисляются координаты «ближайшего» пикселя исходного изображения (x;y). Значение пикселя (x';y') заполняется значением пикселя (x;y). В случае получения дробного числа в качестве координаты пикселя, оно округляется вниз до целого.

Плюсы этого метода – простота реализации и скорость вычислений. Однако при увеличении изображения методом ближайшего соседа появляется «ступенчатость», а при уменьшении часть цветов и важных деталей могут быть утеряны, так как при уменьшении будут задействованы не все пиксели исходного изображения. Данный метод часто используется для предварительной обработки изображения, пока финальное изображение вычисляется с помощью другого метода, обеспечивающего более высокое качество и требующего больших временных затрат. Метод ближайшего соседа можно реализовать без применения гомоморфного шифрования, так как вычисления производятся над индексами пикселей, а не над самими пикселями исходного изображения.

Алгоритм масштабирования пиксельной графики

EPX/Scale2x/AdvMAME2x.EPX («Eric's Pixel eXpansion», пиксельное увеличение Эрика) – алгоритм масштабирования пиксельной графики, разработанный Эриком Джонстоном из LucasArts при портировании игрового движка с IBMPC, где разрешение было 320 на 200 пикселей, на первые цветные компьютеры Macintosh, где разрешение экрана было больше примерно вдвое. AdvMAME2x и Scale2x – это разработанные позднее модификации алгоритма EPX; они имеют более эффективную, но функционально идентичную реализацию. Данные алгоритмы масштабирования цифровых изображений созданы специально для увеличения графических изображений низкого качества. Они дают в качестве результата менее размытую картинку, чем традиционные алгоритмы и позволяют избежать ступенчатости, как при использовании метода ближайшего соседа. Следует отметить, что данные алгоритмы позволяют увеличивать изображение только в кратное двум число раз (2x, 4x) и не позволяют уменьшать изображение.

Вышеперечисленные алгоритмы немного отличаются реализацией метода корректировки цвета результирующих пикселей, но имеют схожий принцип работы (рис. 3):

- для каждого пикселя исходного изображения генерируется 4 пикселя результирующего изображения;
- все пиксели результирующего изображения красятся в цвет исходного пикселя;
- цвет результирующих пикселей корректируется в зависимости от цветов пикселей, пограничных с исходным пикселем;
- если у исходного пикселя нет соседних пикселей с одной или нескольких сторон (пиксель на краю изображения), то значения недостающих соседних пикселей принимаются за значение исходного пикселя.

Корректировка цвета результирующих пикселей в алгоритме EPX выполняется следующим образом:

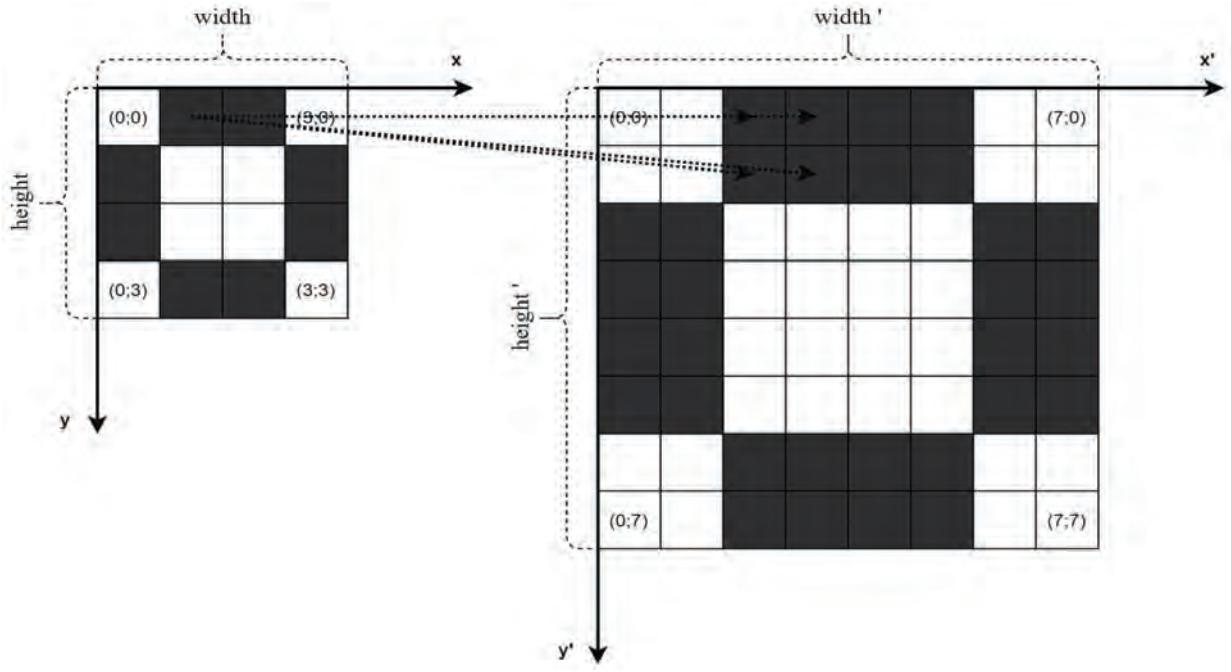

Рис.2. Метод ближайшего соседа

DOI: 10.21681/2311-3456-2021-3-2-10

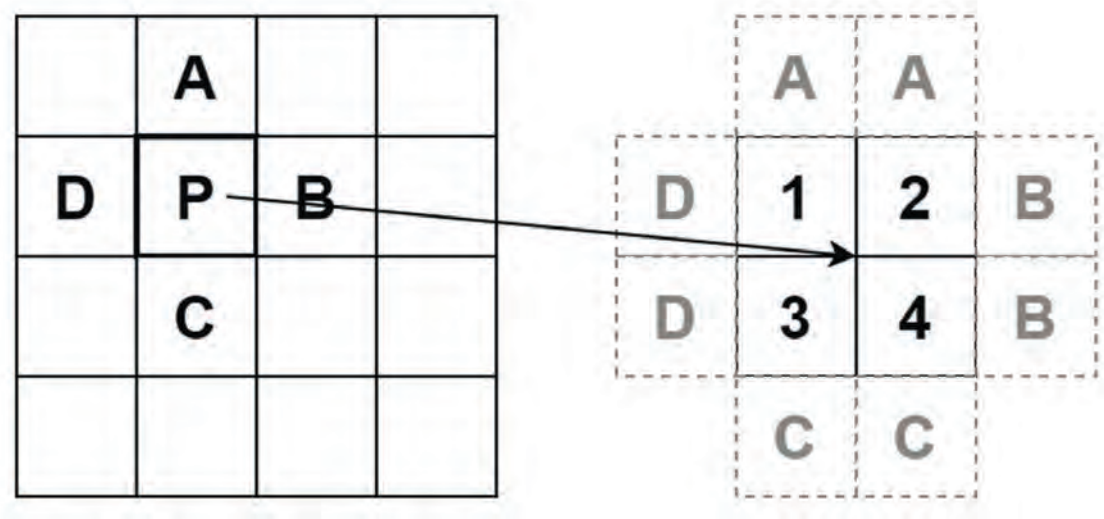

Рис.3. Алгоритм EPX

$$\begin{aligned}&\text{Если A == B, то 1 = A, иначе 1 = P}\\&\text{Если A == C, то 2 = A, иначе 2 = P}\\&\text{Если D == B, то 3 = D, иначе 3 = P}\\&\text{Если D == C, то 4 = D, иначе 4 = P}\end{aligned} \quad (7)$$

Если значения пикселей A и B исходного изображения равны, то пиксель результирующего изображения 1 красится в цвет пикселя A, иначе цвет пикселя 1 остается равен P (начальному заполнению). Аналогичные операции (формула 7) выполняются для остальных пикселей результирующего изображения.

Для корректировки цветов необходимо выполнить сравнение двух пикселей. Следовательно, для выполнения гомоморфной реализации алгоритма EPX нам понадобится некоторый метод выбора одного из двух значений на основе результата сравнения. Этот метод должен выполняться над гомоморфно зашифрованными данными.

Метод сравнения двух битов

Разработка метода сравнения битов актуальна тем, что применение этого метода позволит выполнить гомоморфную реализацию многих алгоритмов, имеющих нелинейную структуру (содержащих в себе условия вида «Если {условие} то {действие$_1$}, иначе {действие$_2$}»). К примеру, как было ранее рассмотрено в тезисе [18], для гомоморфной реализации алгоритма Гаусса требуется операция сравнения. Наличие подобной операции позволило бы полноценно реализовать алгоритм Гаусса на облачном сервере. Операция сравнения требуется и во многих других математических алгоритмах. Основная проблема, которая возникает на данном этапе – проблема поддержки и математических, и логических операций над зашифрованными данными. Гомоморфные криптосистемы над целыми числами поддерживают только математические операции, в то время как криптосистемы над битами и массивами бит поддерживают только логические операции. Вероятно, для решения этой проблемы потребуется реализация математических операций в гомоморфных криптосистемах над битами, однако в этом случае сильно возрастет сложность вычислений. Данная проблема будет подробней рассмотрена в другой статье. На основе вышесказанного, на данном этапе разработка метода сравнения двух битов позволит выполнить гомоморфную реализацию нелинейных алгоритмов, в которых не требуются математические вычисления, например, алгоритма EPX.

Предлагаемый метод применим к гомоморфно зашифрованным битам. В результате выполнения метода будет получен бит, значение которого указывает на результат сравнения.

Пусть даны два бита a, b, зашифрованных гомоморфно. Функция сравнения будет иметь следующий вид:

$$e = \overline{a \oplus b} \quad (8)$$

где $\oplus$ – некоторая гомоморфная операция, эквивалентная операции «исключающее ИЛИ» над открытыми данными.

Мы применяем инверсию к результату для удобства, чтобы в случае равенства чисел получать в результате 1, а в обратном случае – 0. Данная функция позволяет только выявить, равны ли биты или нет. Вычислить, является ли бит a больше бита b, или bбольше или равен a, можно по следующей формуле:

$$e' = (a \oplus b) \wedge a \quad (9)$$

где $\wedge$ – некоторая гомоморфная операция, эквивалентная операции конъюнкции над открытыми данными.

В результате вычисления формулы 9 мы получим 1, при a=1, b=0 и 0 во всех других случаях.

Метод сравнения двух чисел

Предлагаемый метод применим к гомоморфно зашифрованным числам при условии их побитового представления. В результате выполнения метода будет получен бит, значение которого указывает на результат сравнения. Метод позволяет определить, равны числа или нет, но не позволяет выявить большее из них.

Пусть даны два целых числа A, B, которые могут быть представлены побитно в виде $a_1...a_n$ и $b_1...b_n$ соответственно. Тогда для сравнения этих чисел необходимо сравнить поочередно соответствующие биты. Функция сравнения этих чисел будет иметь вид:

$$e = \overline{(a_1 \oplus b_1) \vee (a_2 \oplus b_2) \vee \ldots \vee (a_n \oplus b_n)} \quad (10)$$

Таким образом мы используем метод сравнения двух бит и применяем его поочередно ко всем парам бит $(a_n;b_n)$. Мы применяем инверсию к результату для удобства, чтобы в случае равенства чисел получать в результате 1, а в обратном случае – 0. Данный метод может быть применен к гомоморфно зашифрованным данным при условии, что гомоморфный алгоритм шифрования поддерживает следующие логические операции: конъюнкция, дизъюнкция, исключающее ИЛИ, инверсия. В том случае, если гомоморфный алгоритм поддерживает следующие операции: исключающее ИЛИ и конъюнкция, можно применить к формуле законы де Моргана, чтобы заменить дизъюнкцию на конъюнкцию и упростить вычисления[6].

Проблема поддержки логических операций

Для поддержки предложенных методов сравнения гомоморфный алгоритм должен поддерживать гомоморфные операции, эквивалентные следующим операциям: конъюнкция, дизъюнкция, исключающее ИЛИ, инверсия. Однако необязательно поддерживать все перечисленные операции. Достаточным условием будет поддержка функционально полной системы логических функций. Функционально полная система логических функций представляет собой набор логических функций, с помощью которых можно записать любую, сколь угодно сложную функцию. В этом случае говорят, что этот набор образует базис. Функционально полными являются 3 базиса:

- «И-ИЛИ-НЕ» – базис конъюнкции, дизъюнкции, инверсии
- «И-НЕ» – функцию «ИЛИ” можно реализовать через обратное преобразование
- «ИЛИ-НЕ» – функцию «И” можно реализовать через обратное преобразование

Напомним, как можно реализовать все логические операции с использованием базиса «И-НЕ” (рис. 4).

Гомоморфная реализация алгоритма EPX

Использование предложенного метода для сравнения чисел позволяет выполнить гомоморфную реализацию алгоритма EPX. В случае обработки цветных цифровых изображений каждый пиксель состоит из 3-х цветовых компонент RGB модели. Каждый компонент представляет собой целое число в диапазоне [0; 255] и может быть представлен 8-ю битами. Следовательно, каждый пиксель исходного изображения будет представлен 24 битами, то есть 24 шифротекстами. Для сравнения двух пикселей необходимо выполнить сравнение соответствующих цветовых компонент. Адаптируем предложенный метод для сравнения двух пикселей цифрового изображения.

Пусть даны два пикселя C, D, которые представлены цветовыми компонентами $\{R_c, G_c, B_c\}$ и $\{R_d, G_d, B_d\}$ соответственно. Тогда результат сравнения этих пикселей можно получить по формулам:

$$e_r = \left(r_1^c \oplus r_1^d\right) \vee \left(r_2^c \oplus r_2^d\right) \vee \ldots \vee \left(r_n^c \oplus r_n^d\right) \quad (11)$$

$$e_g = \left(g_1^c \oplus g_1^d\right) \vee \left(g_2^c \oplus g_2^d\right) \vee \ldots \vee \left(g_n^c \oplus g_n^d\right) \quad (12)$$

$$e_b = \left(b_1^c \oplus b_1^d\right) \vee \left(b_2^c \oplus b_2^d\right) \vee \ldots \vee \left(b_n^c \oplus b_n^d\right) \quad (13)$$

$$e = \overline{e_r \vee e_g \vee e_b} \quad (14)$$

Для удобства будем обозначать предложенный метод как «eq», а операцию сравнения двух пикселей A,B как «A eq B». Используя этот метод, выполним гомоморфную адаптацию алгоритма EPX. Начальное заполнение выполняется на основании размеров изображения и позиций пикселей, этот процесс будет одинаково выполняться как над открытыми, так и над зашифрованными данными. Гомоморфная реализация корректировки цветов после начального заполнения будет выполняться следующим образом:

$$E_1 = A\,eq\,B;1 = \left(E_1 \wedge A\right) \vee \left(\overline{E_1} \wedge P\right) \quad (15)$$

$$E_2 = A\,eq\,C;2 = \left(E_2 \wedge A\right) \vee \left(\overline{E_2} \wedge P\right) \quad (16)$$

$$E_3 = D\,eq\,B;3 = \left(E_3 \wedge D\right) \vee \left(\overline{E_3} \wedge P\right) \quad (17)$$

$$E_4 = D\,eq\,C;4 = \left(E_4 \wedge D\right) \vee \left(\overline{E_4} \wedge P\right) \quad (18)$$

Основным минусом данной реализации является ее высокая ресурсоемкость. Рассмотрим в качестве примера реализации полностью гомоморфный алгоритм Джентри. Он поддерживает следующие логические операции: конъюнкция, исключающее ИЛИ, инверсия. Дизъюнкция может быть выражена следующим образом:

$$c_1 \vee c_2 = \left(c_1 \wedge c_2\right) \oplus c_1 \oplus c_2 \quad (19)$$

Если предположить, что пиксели исходного изображения представлены 3-мя цветовыми компонентами, каждая из которых представлена числом в диапазоне [0; 255], то для представления каждого пикселя потребуется минимум 24 бита. Тогда, чтобы выполнить увеличение всего одного пикселя исходного изображения, потребуется выполнить 4 операции сравнения пикселей, а потом для каждой операции сравнения нужно будет выполнить операцию инверсии пикселя, две опе-

6 Пухальский Г. И., Новосельцева Т. Я. Цифровые устройства: Учебное пособие для вузов. — СПб.: Политехника, 1996. — 885 с.

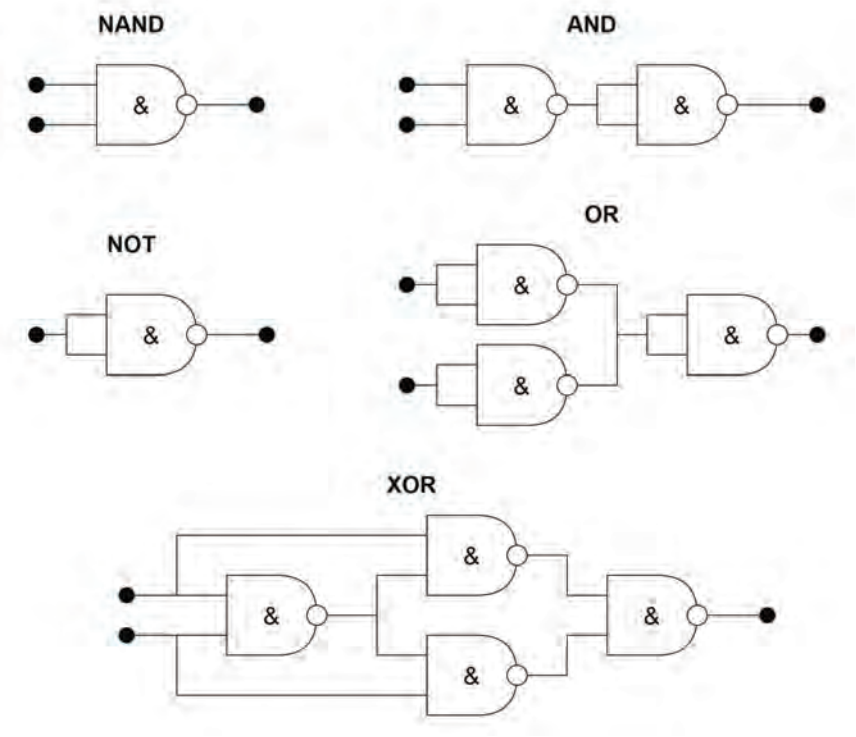

Рис.4. Базис «И-НЕ»

рации конъюнкции между пикселями и операцию дизъюнкции между пикселями. Для упрощения вычислений воспользуемся операцией сравнения, реализованной через конъюнкцию. Каждая операция между пикселями – это аналогичная операция над битами, выполненная 24 раза. Каждая операция сравнения пикселей – это 24 операции исключающего ИЛИ, 23 конъюнкции и 24 инверсии. Таким образом, для увеличения одного пикселя исходного изображения методом EPX потребуется:96 операций исключающего ИЛИ; 100 операций инверсии; 100 операций конъюнкции 4 операции дизъюнкции. Если раскрыть операцию дизъюнкции, то получится: 104 операции исключающего ИЛИ; 100 операций инверсии; 104 операции конъюнкции над битами.

При обработке изображения можно будет несколько оптимизировать число выполняемых операций за счет кэширования части результатов предыдущих операций, однако сложность вычислений все равно останется достаточно высокой.

Вывод. В работе рассмотрена проблема облачной обработки растровых изображений с применением гомоморфной криптографии. Приведено примерное описание схемы работы облачного сервиса, выполняющую обработку изображений. Рассмотрены некоторые простейшие методы масштабирования изображений, такие как метод «ближайшего соседа», EPX. Данные методы были проанализированы на возможность их гомоморфной реализации, была выявлена потребность в гомоморфном методе сравнения двух чисел, позволяющем выбрать одно из двух значений на основе результата сравнения. Для решения этой проблемы были предложены методы гомоморфного сравнения битов и чисел. На основе метода гомоморфного сравнения была предложена гомоморфная реализация алгоритма EPX. Предложенный метод позволяет выполнить увеличение зашифрованного гомоморфно изображения. Однако предложенная реализация обладает высокой ресурсоемкостью. Приведен приблизительный анализ числа необходимых логических операций над битами для выполнения увеличения одного пикселя исходного изображения алгоритмом EPX. Используя предложенный метод, можно выполнить увеличение изображение на облачном сервисе с использованием полностью гомоморфного шифрования. Также предложенный метод гомоморфного сравнения может быть использован для выполнения гомоморфной реализации других алгоритмов, в которых необходима операция сравнения.

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

Литература

1. 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.
2. Бабенко Л.К., Буртыка Ф.Б., Макаревич О.Б., Трепачева А.В. Полностью гомоморфное шифрование (обзор) // Вопросы защиты информации. 2015. №. 3. С. 3–26.
3. Егорова В.В., Чечулина Д.К. Построение криптосистемы с открытым ключом на основе полностью гомоморфного шифрования // Прикладная дискретная математика. Приложение, 2015, выпуск 8, С. 59–61.
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. DOI: 10.21681/2311-3456-2019-5-70-74
6. Трубей А.И. Гомоморфное шифрование: безопасность облачных вычислений и другие приложения (обзор) // Информатика, Минск. 2015. С. 90-101.
7. Буртыка Ф.Б. Симметричное полностью гомоморфное шифрование с использованием неприводимых матричных полиномов // Известия ЮФУ. Технические науки. 2014. № 8. С.107–122.
8. Трепачева А.В. Криптоанализ симметричных полностью гомоморфных линейных криптосистем на основе задачи факторизации чисел // Известия ЮФУ. Технические науки. 2015. № 5 (166). С. 89–102.
9. C. Gentry, S. Halevi, Implementing gentry's fully-homomorphic encryption scheme // EUROCRYPT, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed. – vol. 6632. – Springer. 2011. pp. 129–148.
10. Бабенко Л.К., Русаловский И.Д., Библиотека полностью гомоморфного шифрования целых чисел // Известия ЮФУ. Технические науки. 2020. №2. С. 79-88.
11. Бабенко Л.К., Русаловский И.Д., Метод реализации гомоморфного деления // Известия ЮФУ. Технические науки. 2020. №4. С. 212-221.
12. Буртыка Ф.Б. Пакетное симметричное полностью гомоморфное шифрование на основе матричных полиномов // Труды Института системного программирования РАН. 2014. Т. 26. № 5. С. 99–116.
13. Бабенко Л.К., Буртыка Ф.Б., Макаревич О.Б., Трепачева А.В. Защищенные вычисления и гомоморфное шифрование // Программные системы: теория и приложения. 2014. 25 с.
14. Макаревич О. Б., Буртыка Ф. Б. Защищенная облачная база данных с применением гомоморфной криптографии. Тез.докл. 6-й Росс. мультиконференции «Информационные технологии в управлении» (ИТУ–2014). СПб, 2014. С. 567-572.
15. Минаков С.С. Основные криптографические механизмы защиты данных, передаваемых в облачные сервисы и сети хранения данных // Вопросы кибербезопасности. 2020. № 3(37). С. 66-75. DOI: 10.21681/2311-3456-2020-05-66-75
16. Варновский Н.П., Захаров В.А., Шокуров А.В. К вопросу о существовании доказуемо стойких систем облачных вычислений // Вестник Московского университета, Серия 15, Вычислительная математика и кибернетика. 2016. № 2. С. 32-46.
17. Астахова Л.В., Султанов Д.Р., Ашихмин Н.А. Защита облачной базы персональных данных с использованием гомоморфного шифрования // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2016. Т. 16, №3. С. 52-61.
18. 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.

SCALING DIGITAL IMAGES USING HOMOMORPHIC ENCRYPTION

Babenko L.K.[7], Rusalovsky I.D.[8]

Abstract. *Since time immemorial, cryptography has provided secure transmission of information in an insecure environment, keeping the data secret. Not so long ago the homomorphic cryptography began to actively develop. Its distinctive feature is that this type of cryptography allows you to process encrypted data without their preliminary decryption in such a way that the result of operations on encrypted data is equivalent, after decryption, to the result of operations on open data. Because of these features, homomorphic encryption can be effectively used in various cloud services to perform secure computing and secure image processing. At the same time, it is guaranteed that no one will have open data, even the service that performs the calculations.*

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

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

Purpose of the work: *development of methods and tools for homomorphic encryption that allow performing homomorphic implementation of image processing algorithms.*

Research methods: *analysis of possible implementations of digital image processing using homomorphic encryption, analysis of existing problems of performing a homomorphic implementation for image processing algorithms.*

Results*: a method for homomorphic comparison of bits and numbers presented as an array of bits is proposed; a homomorphic implementation of the EPX image resizing algorithm is proposed; the complexity of the operation is analyzed when one pixel of the original image is enlarged using the proposed method; the analysis results are presented.*

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

References

1. Babenko L.K., Burty`ka F.B., Makarevich O.B., Trepacheva A.V. Metody` polnost`iu gomomorfnogo shifrovaniia na osnove matrichny`kh polinomov // Voprosy` kiberbezopasnosti. 2015. №1. S. 17–20.
2. Babenko L.K., Burty`ka F.B., Makarevich O.B., Trepacheva A.V. Polnost`iu gomomorfnoe shifrovanie (obzor) // Voprosy` zashchity` informatcii. 2015. №. 3. S. 3–26.
3. Egorova V.V., Chechulina D.K. Postroenie kriptosistemy` s otkry`ty`m cliuchom na osnove polnost`iu gomomorfnogo shifrovaniia // Pricladnaia diskretnaia matematika. Prilozhenie, 2015, vy`pusk 8, S. 59–61.
4. Babenko L.K., Trepacheva A.V. O nestoi`kosti dvukh simmetrichny`kh gomomorfny`kh kriptosistem, osnovanny`kh na sisteme ostatochny`kh classov // Trudy` Instituta sistemnogo programmirovaniia RAN. 2019. T. 18. № 1. S. 230-262.
5. Arakelov G.G. Voprosy` primeneniia pricladnoi` gomomorfnoi` kriptografii // Voprosy` kiberbezopasnosti. 2019. № 5(33). S. 70-74. DOI: 10.21681/2311-3456-2019-5-70-74
6. Trubei` A.I. Gomomorfnoe shifrovanie: bezopasnost` oblachny`kh vy`chislenii` i drugie prilozheniia (obzor) // Informatika, Minsk. 2015. S. 90-101.
7. Burty`ka F.B. Simmetrichnoe polnost`iu gomomorfnoe shifrovanie s ispol`zovaniem neprivodimy`kh matrichny`kh polinomov // Izvestiia IUFU. Tekhnicheskie nauki. 2014. № 8. S.107–122.
8. Trepacheva A.V. Kriptoanaliz simmetrichny`kh polnost`iu gomomorfny`kh linei`ny`kh kriptosistem na osnove zadachi faktorizatcii chisel // Izvestiia IUFU. Tekhnicheskie nauki. 2015. № 5 (166). S. 89–102.
9. C. Gentry, S. Halevi, Implementing gentry's fully-homomorphic encryption scheme // EUROCRYPT, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed. – vol. 6632. – Springer. 2011. pp. 129–148.
10. Babenko L.K., Rusalovskii` I.D., Biblioteka polnost`iu gomomorfnogo shifrovaniia tcely`kh chisel // Izvestiia IUFU. Tekhnicheskie nauki. 2020. №2. S. 79-88.
11. Babenko L.K., Rusalovskii` I.D., Metod realizatcii gomomorfnogo deleniia // Izvestiia IUFU. Tekhnicheskie nauki. 2020. №4. S. 212-221.
12. Burty`ka F.B. Paketnoe simmetrichnoe polnost`iu gomomorfnoe shifrovanie na osnove matrichny`kh polinomov // Trudy` Instituta sistemnogo programmirovaniia RAN. 2014. T. 26. № 5. S. 99–116.
13. Babenko L.K., Burty`ka F.B., Makarevich O.B., Trepacheva A.V. Zashchishchenny`e vy`chisleniia i gomomorfnoe shifrovanie // Programmny`e sistemy`: teoriia i prilozheniia. 2014. 25 s.
14. Makarevich O. B., Burty`ka F. B. Zashchishchennaia oblachnaia baza danny`kh s primeneniem gomomorfnoi` kriptografii. Tez.docl. 6-i` Ross. mul`tikonferentcii «Informatcionny`e tekhnologii v upravlenii» (ITU–2014). SPb, 2014. S. 567-572.
15. Minakov S.S. Osnovny`e kriptograficheskie mehanizmy` zashchity` danny`kh, peredavaemy`kh v oblachny`e servisy` i seti khraneniia danny`kh // Voprosy` kiberbezopasnosti. 2020. № 3(37). S. 66-75. DOI: 10.21681/2311-3456-2020-05-66-75
16. Varnovskii` N.P., Zaharov V.A., Shokurov A.V. K voprosu o sushchestvovanii dokazuemo stoi`kikh sistem oblachny`kh vy`chislenii` // Vestneyk Moskovskogo universiteta, Seriia 15, Vy`chislitel`naia matematika i kibernetika. 2016. № 2. S. 32-46.
17. Astahova L.V., Sultanov D.R., Ashikhmin N.A. Zashchita oblachnoi` bazy` personal`ny`kh danny`kh s ispol`zovaniem gomomorfnogo shifrovaniia // Vestneyk IUUrGU. Seriia "Komp`iuterny`e tekhnologii, upravlenie, radioe`lektronika". 2016. T. 16, №3. S. 52-61.
18. Rusalovskii` I.D. Gomomorfnaia realizatciia algoritma Gaussa // Sbornik statei` IV Vserossii`skoi` nauchno-tekhnicheskoi` konferentcii molody`kh ucheny`kh, aspirantov i studentov «Fundamental`ny`e i pricladny`e aspekty` komp`iuterny`kh tekhnologii` i informatcionnoi` bezopasnosti». 2014. S. 364-367.