25. *Zhidyaev A.V., Kopysov A.N., Bogdanov A.A., Savel'ev A.V., Nikitin M.L*. Issledovanie energeticheskikh kharakteristik signalov, primenyaemykh dlya peredachi dannykh po dekametrovomu kanalu [Investigation of energy characteristics of signals used for data transmission over a decameter channel], *Vestnik IzhGTU im. M.T. Kalashnikova* [Bulletin of Kalashnikov ISTU], 2015, No. 3 (67), pp. 85-88.
26. *Bridger Wray W., Ruiz Mark D.* Advisors: Aruna Apte, James B. Greene. Naval Postgraduate School Monterey, California "Total Ownership Cost Reduction Case Study: AEGIS Radar Phase Shifters" December 2006.
27. *Stupnitskiy M.M., KHaritonov N.I., Devyatkin E.E*. Infokommunikatsionnaya infrastruktura tsifrovoy ekonomiki: zadachi otraslevogo instituta [Information and communication infrastructure of the digital economy: challenges for industry Institute], *Elektrosvyaz'* [Telecommunications], 2018, No. 4, pp. 70-76.

Статью рекомендовал к опубликованию к.т.н., профессор О.В. Воробьев.

Рыбаков Алексей Игоревич – Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича; e-mail: lexeus.r1@gmail.com; 193232, Санкт-Петербург, пр. Большевиков, 22; кафедра радиопередающих устройств и средств подвижной связи; аспирант.

Кротов Роман Евгеньевич – e-mail: ub1cag@yandex.ru; кафедра радиопередающих устройств и средств подвижной связи; аспирант.

Кокин СергейАлексеевич – e-mail: sergeikokins@gmail.com; кафедра радиопередающих устройств и средств подвижной связи; магистр.

Rybakov Aleksei Igorevich – St. Petersburg state University of telecommunications. prof. M.A. Bonch-Bruevich; e-mail: lexeus.r1@gmail.com; 22 Bolshevikov Ave., Saint Petersburg, 193232, Russia; the department of radio transmitting devices and means of mobile communication; post-graduate student.

Krotov Roman Evgen'evich – e-mail: ub1cag@yandex.ru; the department of radio transmitting devices and means of mobile communication; post-graduate student.

Kokin Sergey Alexandrovich – e-mail: sergeikokins@gmail.com; the department of radio transmitting devices and means of mobile communication; master.

УДК 004.422 DOI 10.18522/2311-3103-2020-2-218-227

Л.К. Бабенко, И.Д. Русаловский

БИБЛИОТЕКА ПОЛНОСТЬЮ ГОМОМОРФНОГО ШИФРОВАНИЯ ЦЕЛЫХ ЧИСЕЛ*

Рассматривается одно из новых направлений криптографии – гомоморфная криптография. 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. В работе приведены основные области применения гомоморфного шифрования. Выполнен анализ существующих разработок в области гомоморфного шифрования. Анализ показал, что существующие реализации библиотек позволяют обрабатывать только биты или массивы бит и не поддерживают операцию деления. Однако для решения прикладных задач необходима поддержка выполнения целочисленных операций. В результате анализа была выявлена необходимость реализация операции гомоморфного деления, а также актуальность разработки собственной реализации библиотеки гомоморфного шифрования над целыми числами. Возможность выполнения четырех операций (сложение, разность, умножение и деление) над

* Работа выполнена при финансовой поддержке РФФИ в рамках проекта № 18-07-01347.

зашифрованными данными позволит расширить области прикладного использования гомоморфного шифрования. Предложен метод гомоморфного деления, позволяющий выполнять операцию деления над гомоморфно зашифрованными данными. Предложена архитектура библиотеки полностью гомоморфных операций над целыми. Библиотека поддерживает основные гомоморфные операции над целыми числами, а также операцию деления, благодаря методу гомоморфного деления. На базе предложенных метода гомоморфного деления и архитектуры библиотеки была выполнена реализация библиотеки гомоморфных операций над целыми. В статье также приведены замеры времени, необходимого на совершение определенных операций над зашифрованными данными и выполняется анализ эффективности работы разработанной реализации библиотеки. Приводятся выводы и возможные пути дальнейшего развития.

Гомоморфное шифрование; криптографическая защита; криптографическая библиотека; C++.

L.K. Babenko, I.D. Rusalovsky

THE LIBRARY OF FULLY HOMOMORPHIC ENCRYPTION OVER THE INTEGERS

The article discusses one of the new directions of cryptography, a homomorphic cryptography. Its distinctive feature is that this type of cryptography allows you to process encrypted data without first decrypting it in such a way that the result of operations on encrypted data is equivalent after decryption to the result of operations on open data. The paper describes the main areas of application of homomorphic encryption. The analysis of existing developments in the field of homomorphic encryption is performed. The analysis showed that existing library implementations only allow processing of bits or arrays of bits and do not support the division operation. However, to solve applied problems, support for performing integer operations is necessary. The analysis revealed the need to implement the operation of homomorphic division, as well as the relevance of developing your own implementation of a library of homomorphic encryption over integers. The ability to perform four operations (addition, difference, multiplication and division) on encrypted data will expand the field of application of homomorphic encryption. A method of homomorphic division is proposed, which allows performing the division operation on homomorphically encrypted data. A library architecture of completely homomorphic operations on integers is proposed. The library supports the basic homomorphic operations on integers, as well as the division operation, thanks to the method of homomorphic division. Based on the proposed method of homomorphic division and library architecture, a library of homomorphic operations on integers was implemented. The article also provides measurements of the time required to perform certain operations on encrypted data and analyzes the effectiveness of the developed library implementation. Conclusions and possible ways of further development are given.

Homomorphic encryption; cryptographic protection; cryptographic library; C ++.

Введение. Криптография с незапамятных времен обеспечивают безопасную передачу информации в небезопасной среде, сохраняя пересылаемые данные в секрете. Эта наука непрерывно развивается. Не так давно зародилось новое направление – гомоморфная криптография. 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 [1–7]. При этом решается одна из проблем криптографии – генерации, хранения и распространения общих сеансовых ключей. При этом повышается уровень защищенности данных – сервер получает зашифрованные данные, обрабатывает их и возвращает зашифрованный результат, а открытые данные и ключи шифрования не покидают безопасный сегмент при сетевом взаимодействии.

Гомоморфное шифрование, несмотря на все свои достоинства, порождает ряд проблем, которые в будущем будет необходимо решить:

- необходимость обеспечения целостности пересылаемых данных;
- высокая трудоемкость операций над зашифрованными данными;
- быстрый рост коэффициентов после выполнения операций над данными.

Развитие и совершенствование гомоморфного шифрования актуально в наше время [8–10]. Улучшение алгоритмов и схем данного направления позволит заменить его во многих сферах, где в данное время используется симметричное шифрование:

- Облачные вычисления.
- Облачная обработка фотографий.
- Электронные голосование (выборы).
- Защищенный поиск информации.

Анализ актуальности. На данный момент уже выполнено несколько реализаций библиотек для гомоморфного шифрования [11–19]. Наиболее серьезными реализациями, доступными для общего пользования, можно считать две:

- библиотека HElib, сделанная Шаем Хавели и Виктор Шоуп которая реализует криптосистему BGV с GHS оптимизацией;
- библиотека FHEW сделанная Лео Дуглас и Даниэль Миккианакио которая является реализацией комбинации криптосистемы обучения с ошибками Регева и техники создания гибкой схемы Алперин-Шериффа и Пейкерта.

Библиотеки имеют высокую скорость работы, хорошую оптимизацию. Обе реализации выполнены на языке программирования C++. Однако, учитывая области применения гомоморфного шифрования, вышеперечисленные библиотеки имеют низкую практическую ценность, так как позволяют выполнять обработку только битов (либо массивов бит), но не целочисленных значений. Также в представленных реализациях отсутствует операция деления над зашифрованными данными. Опираясь на проведенный анализ, сделан вывод о необходимости разработки библиотеки полностью гомоморфного шифрования, позволяющей работать с целыми числами и выполнять над ними все математические операции (сложение, разность, умножение и деление).

Метод гомоморфного деления. Существующие схемы и алгоритмы гомоморфного шифрования не позволяют использовать операцию деления над зашифрованными данными. Для решения данной проблемы предлагается использовать некую абстракцию, построенную над шифротекстом и расширяющую возможности по выполнению математических операций над ним. Для этого необходимо выделить два уровня представления данных – криптографический и математический. На математическом уровне данные будут представлены в виде простых дробей, а операция деления будет реализована как операция деления простых дробей. Данное решение было разработано в результате анализа [20] и будет подробней описано далее в статье. При реализации библиотеки с предложенной архитектурой при соблюдении низкой связности модулей и общего интерфейса можно добиться того, чтобы без труда заменять алгоритмы шифрования криптографического уровня, но при этом сохранять весь функционал, который предоставляет математический уровень.

Криптографический уровень представляет собой модуль, основным типом данных которого является гомоморфно зашифрованное число. На данном уровне должны быть реализованы возможности по созданию новых шифротекстов на базе открытых текстов и ключей шифрования (операция шифрования данных), получению открытых данных из шифротекста на основании ключа шифрования (операция расшифрования данных), а также основные математические операции над шифротекстами – сложение (а также разность) и умножение.

Математический уровень является надстройкой над криптографическим. Для реализации операции деления математический уровень будет реализован как простая дробь – он содержит экземпляры объектов криптографического уровня в виде делимого и делителя и повторяет интерфейс криптографического уровня, дополняя его операцией деления. Все операции над объектом математического уровня выполняются как операции над простыми дробями и использует все возможности криптографического уровня.

Архитектура библиотеки. Реализация библиотеки выполнена на языке C++. Для поддержки выполнения множественных операций над зашифрованными числами и сведения к минимуму неточностей вычисления (округления значений), используется библиотека больших чисел NTL.

При реализации разработанной библиотеки перед ней стояли задачи:

- Возможность обработки целых чисел.
- Полностью гомоморфное шифрование.
- Поддержка всех математических операций, включая операцию деления.

Для поддержки операции деления архитектура библиотеки реализована на основе приведенного выше метода гомоморфного деления. Библиотека представлена классами криптографическим, математическим и классом, отвечающим за ключевую информацию. Архитектура библиотеки представлена на рис. 1.

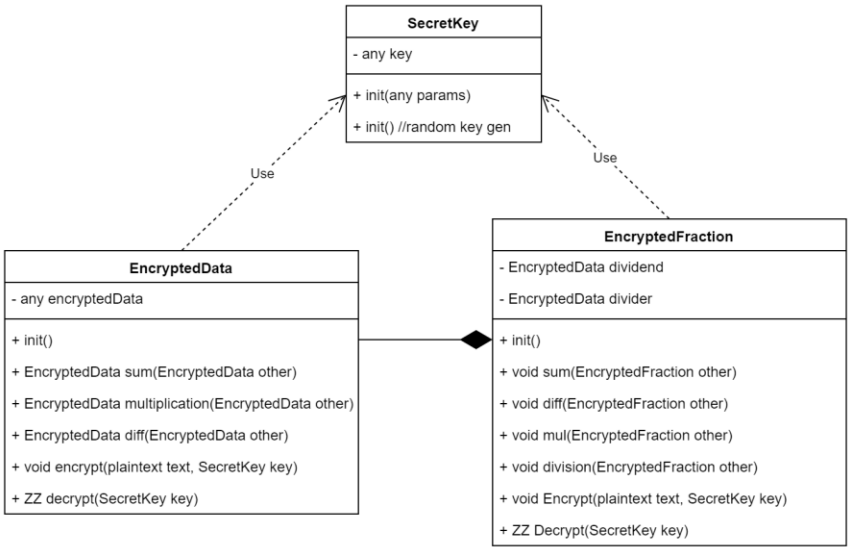

Рис. 1. Архитектура библиотеки

Secret Key. Класс оперирует информацией о секретном ключе, использующемся в криптографическом алгоритме. Предоставляет возможности по созданию, случайной генерации нового ключа и его использованию. В текущей реализации библиотеки для генерации ключей и полиномов используется генератор случайных чисел из стандартной библиотеки с автоматической рандомизацией относительно текущего времени. В статье приведено описание библиотеки относительно симметричного шифрования, однако небольшое изменение библиотеки позволит работать и с ассиметричным шифрованием. Интерфейс класса представлен на рис. 2.

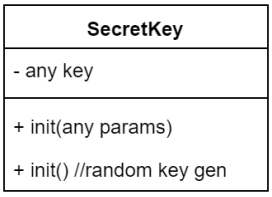

Рис. 2. Интерфейс класса SecretKey

Encrypted Data. Класс, определяющий основной тип данных криптографического уровня. Шифрование и расшифровка возможны с использованием заранее сгенерированного ключа, либо с помощью передачи секретных параметров. Также класс реализует все необходимые математические операции криптографического уровня – сумму, разность и умножение. Интерфейс класса представлен на рис. 3.

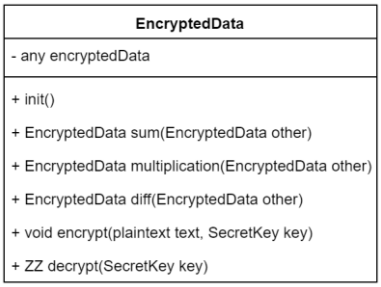

Рис. 3. Интерфейс класса EncryptedData

Encrypted Fraction. Зашифрованная дробь – основной объект данных математического уровня. Содержит в себе делимое и делитель – объекты данных криптографического уровня (EncryptedData). Реализует все основные операции над дробями – сумму, разность, умножение и деление. Позволяет шифровать данные и расшифровывать полученную дробь на секретном ключе (в ходе операции расшифровки расшифровываются делимое и делитель, а над результатами расшифровки производится операция деления). Архитектура класса представлена на рис. 4.

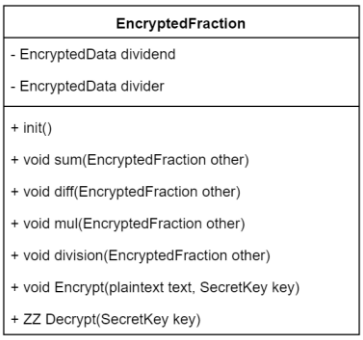

Рис. 4. Интерфейс класса EncryptedFraction

Примеры использования библиотеки. Для использования библиотеки необходимы базовые навыки программирования на языке C++. Ниже приведен пример использования библиотеки. В примере фигурируют полиномы, так как в на-

чальной реализации библиотеки именно в виде полиномов представлялись зашифрованные данные, но при замене криптографического уровня библиотеки изменится и представление зашифрованных данных.

Первоначально создается секретный ключ шифрования. Секретный ключ может быть создан на основе заданных параметров или сгенерирован случайным образом.

```
// Генерация секретного ключа шифрования
SecretKey secKey = SecretKey();
```

После создания секретного ключа необходимо создать объекты шифротекстов и выполнить их начальную инициализацию. В данном примере создаются начальные шифрующие полиномы 10 степени с коэффициентами в диапазоне (0, 10000).

```
// Начальная инициализация шифротекстов
int range = 10000;
int power = 10;
EncryptedFraction firstPoly = EncryptedFraction(power, range);
EncryptedFraction secondPoly = EncryptedFraction(power, range);
```

На начально сгенерированных объектах можно выполнить шифрование данных. Для этого необходимо вызвать соответствующий метод и передать ему в качестве параметров открытый текст и секретный ключ шифрования. В качестве примера приводится шифрование чисел 10 и 20.

```
// Шифрование данных
// ZZ – основной тип данных библиотеки NTL – большое целое
firstPoly.Encrypt(ZZ(10), secKey);
secondPoly.Encrypt(ZZ(20), secKey);
```

Над зашифрованными данными можно выполнять различные математические операции. После выполнения операций над данными можно их расшифровать на секретном ключе и получить результат операций над данными. В примере все операции выполняются последовательно.

```
// Математические операции над данными и последующая расшифровка
firstPoly.sum(secondPoly);
ZZ sum = firstPoly.Decrypt(secKey);
// sum = 30
firstPoly.diff(secondPoly); // Из первого полинома
ZZ diff = firstPoly.Decrypt(secKey);
// diff = 10
firstPoly.mul(secondPoly);
ZZ mul = firstPoly.Decrypt(secKey);
// mul = 200

firstPoly.division(secondPoly);
ZZ division = firstPoly.Decrypt(secKey);
// division = 10
```

Анализ эффективности. Разработанная библиотека тестировалась на производительность. Тесты проводились на ноутбуке со следующими характеристиками:

- ОС – Windows 10 Pro;

♦ процессор – Intel Core i5-3210M, 4 ядра, 2,5 ГГц (библиотека не имеет параллельных операций, все вычисления происходят на одном ядре);

♦ оперативная память – 8 Гб, DDR3.

В ходе тестов измерялось время, необходимое на последовательное умножение или сложение числа самого с собой. Тесты выполнялись следующим образом – выбирались начальные параметры шифрования (степень полинома и диапазон, в котором генерировались коэффициенты полинома), на основе выбранных параметров генерируется начальный полином, далее полученный полином складывался с самим собой или умножался сам на себя определенное число раз (1000 раз). Каждые 100 итераций производились замеры времени, необходимого на выполнение этих операций. На основе выполненных замеров оценивалась зависимость времени, необходимого на выполнение последовательных операций сложения или умножения для различного числа операций в серии.

Тесты умножения. На рис. 5 приведены графики зависимости времени обработки данных с помощью библиотеки от числа последовательных операций над данными для операции умножения.

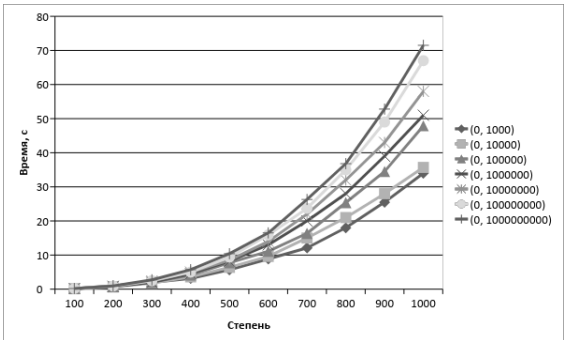

Рис. 5. Возведение в степень полинома 2-го порядка

Для операции умножения наблюдается экспоненциальная сложность выполнения операций. Это обусловлено быстрым ростом шума при выполнении данной операции (после каждой операции умножения объем шифротекста возрастает примерно в 2 раза).

Тесты сложения. На рис. 6 приведены графики зависимости времени обработки данных с помощью библиотеки от числа последовательных операций над данными для операции сложения.

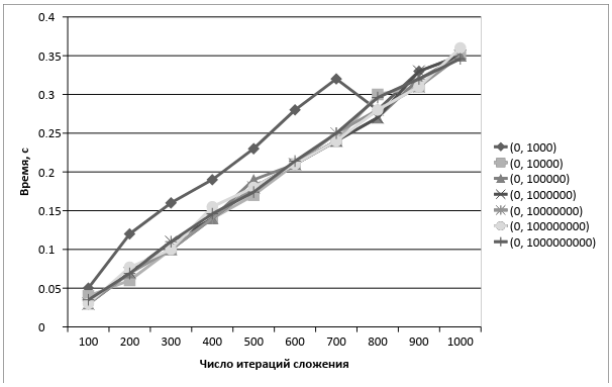

Рис. 6. Сложение полинома 2-го порядка с самим собой

Как видно из представленных графиков, операция сложения имеет линейную сложность от количества последовательных операций сложения в серии. Это связано с низкой скоростью увеличения размера шифротекста в результате операции сложения.

Направление дальнейшей работы. Результатом работы является функциональная библиотека полностью гомоморфного шифрования целых чисел, разработанная на языке C++. Библиотека имеет удобный интерфейс и неплохое быстродействие, как это показали проведенные тесты производительности. Библиотека может использоваться для разработки более сложных продуктов, которые в своем составе используют гомоморфное шифрование. В дальнейшем библиотека будет совершенствоваться и улучшаться. Планируется замена ее криптографического ядра на более защищенный и эффективный алгоритм – реализацию алгоритма Джентри для целых чисел.

Заключение. В ходе проделанной работы был произведен анализ гомоморфного шифрования, выявлены его слабые и сильные стороны и возможности, по его практическому использованию. Был выполнен анализ существующих реализаций библиотек гомоморфного шифрования, который показал, что на данный момент нет реализации, позволяющей работать с целыми числами и иметь полноценный математический аппарат для работы над ними. На основе данного анализа была выявлена необходимость спроектировать и разработать собственную библиотеку полностью гомоморфного шифрования целых чисел, которая позволила бы корректно выполнять криптографические преобразования над целочисленными данными и выполнять над зашифрованными данными все математические операции – сложения, умножения, разности и деления. Для реализации гомоморфного деления также был сформулирован и предложен метод, позволяющий реализовать данную операцию для любого полностью гомоморфного алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

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. *Бабенко Л.К., Буртыка Ф.Б., Макаревич О.Б., Трепачева А.В.* Защищенные вычисления и гомоморфное шифрование // III Национальный суперкомпьютерный форум (25-27 ноября 2014, г. Переславль-Залесский). ИПС имени А.К. Айламазяна РАН. – 2014.
4. *Макаревич О.Б., Буртыка Ф.Б.* Защищенная облачная база данных с применением гомоморфной криптографии // Тез. докладов 6-й Российской мультиконференции «Информационные технологии в управлении» (ИТУ–2014). – СПб., 2014. – С. 567-572.
5. *Буртыка Ф.Б.* Пакетное симметричное полностью гомоморфное шифрование на основе матричных полиномов // Тр. Института системного программирования РАН. – 2014. – Т. 26, № 5. – С. 99-116.
6. *Буртыка Ф.Б.* Симметричное полностью гомоморфное шифрование с использованием неприводимых матричных полиномов // Известия ЮФУ. Технические науки. – 2014. – № 8. – С. 107-122.
7. *Трепачева А.В.* Криптоанализ симметричных полностью гомоморфных линейных криптосистем на основе задачи факторизации чисел // Известия ЮФУ. Технические науки. – 2015. – № 5 (166). – С. 89-102.
8. *Diffie W. and Hellman M.* New directions in cryptography // IEEE Transactions on Information Theory. – 1976. – Vol. IT-22. – P. 644-654.
9. Гомоморфное шифрование. – URL: https://habrahabr.ru/post/255205/ (дата обращения: 01.06.2020).
10. Гомоморфное шифрование своими руками. – URL: https://habrahabr.ru/post/ 150067/ (дата обращения: 01.06.2020).

11. *Gentry.* Fully homomorphic encryption using ideal lattices // STOC. – 2009. – P. 169-178.
12. *Gentry Craig,* A fully homomorphic encryption scheme // A dissertation submitted to the department of computer science and the committee on graduate students of Standford University. – 2009.
13. *Regev O.* New lattice-based cryptographic constructions // J. ACM. – 2004. – Vol. 51, No. 6. – P. 899-942.
14. *Regev O.* On lattices, learning with errors, random linear codes, and cryptography // STOC. – 2005. – P. 84-93.
15. *Rao G.V., Kakulapati V., Purushoththaman M.* Privacy homomorphism in mobile ad hoc networks // International Journal of Research & Reviews in Computer Science. – 2011.
16. Helib. – URL: https://github.com/homenc/HElib (дата обращения: 01.06.2020).
17. FHEW. – URL: https://github.com/lducas/FHEW (дата обращения: 01.06.2020).
18. *Варновский Н.П., Шокуров А.В.* Гомоморфное шифрование // Тр. Института системного программирования РАН. – 2007. – № 12. – С. 27-36.
19. *Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В.* Пороговые системы гомоморфного шифрования и защита информации в облачных вычислениях // Программирование. – 2015. – № 4. – С. 47-51.
20. *Яковлев М.О.* Защищенный калькулятор. Разработка клиентского компонента. – URL: http://pdf.knigi-x.ru/21informatika/429422-1-kafedra-sistem-informatiki-vipusknaya-kvalifikacionnaya-rabota-bakalavra-yakovlev-mihail-olegovich-za.php (дата обращения: 01.06.2020).

REFERENCES

1. *Babenko L.K., Burtyka F.B., Makarevich O.B., Trepacheva A.V.* Metody polnost'yu gomomorfnogo shifrovaniya na osnove matrichnykh polinomov [Methods of fully homomorphic encryption based on matrix polynomials], *Voprosy kiberbezopasnosti* [Cybersecurity issues], 2015, No. 1, pp. 17-20.
2. *Babenko L.K., Burtyka F.B., Makarevich O.B., Trepacheva A.V.* Polnost'yu gomomorfnoe shifrovanie (obzor) [Fully Homomorphic Encryption (Overview)], *Voprosy zashchity informatsii* [Information Security Issues], 2015, No.. 3, pp. 3-26.
3. *Babenko L.K., Burtyka F.B., Makarevich O.B., Trepacheva A.V.* Zashchishchennye vychisleniya i gomomorfnoe shifrovanie [Secure computing and homomorphic encryption], *III Natsional'nyy superkomp'yuternyy forum (25-27 noyabrya 2014, g. Pereslavl'-Zalesskiy). IPS imeni A.K. Aylamazyana RAN* [III National supercomputer forum. November, 25-27 of 2014], 2014.
4. *Makarevich O.B., Burtyka F.B.* Zashchishchennaya oblachnaya baza dannykh s primeneniem gomomorfnoy kriptografii [Secure cloud database using homomorphic cryptography], *Tez. dokladov 6-y Rossiyskoy mul'tikonferentsii «Informatsionnye tekhnologii v upravlenii» (ITU–2014)* [Proceedings of 6th Russian multiconference «Information Technologies in Control» (ITU-2014)]. Saint Petersburg, 2014, pp. 567-572.
5. *Burtyka F.B.* Paketnoe simmetrichnoe polnost'yu gomomorfnoe shifrovanie na osnove matrichnykh polinomov [Batch symmetric fully homomorphic encryption based on matrix polynomials], *Tr. Instituta sistemnogo programmirovaniya RAN* [Proceedings of the Institute for System Programming RAS], 2014, Vol. 26, No. 5, pp. 99-116.
6. *Burtyka F.B.* Simmetrichnoe polnost'yu gomomorfnoe shifrovanie s ispol'zovaniem neprivodimykh matrichnykh polinomov [Symmetric fully homomorphic encryption using irreducible matrix polynomials], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 8, pp. 107-122.
7. *Trepacheva A.V.* Kriptoanaliz simmetrichnykh polnost'yu gomomorfnykh lineynykh kriptosistem na osnove zadachi faktorizatsii chisel [Cryptanalysis of symmetric fully homomorphic linear cryptosystems based on the number factorization problem], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 5 (166), pp. 89-102.
8. *Diffie W. and Hellman M.* New directions in cryptography, *IEEE Transactions on Information Theory*, 1976, Vol. IT-22, pp. 644-654.
9. Gomomorfnoe shifrovanie [Homomorphic encryption]. Available at: https://habrahabr.ru/post/255205/ (accessed 01 June 2020).

10. Gomomorfnoe shifrovanie svoimi rukami [DIY homomorphic encryption]. Available at: https://habrahabr.ru/post/ 150067/ (accessed 01 June 2020).
11. *Gentry.* Fully homomorphic encryption using ideal lattices, *STOC*, 2009, pp. 169-178.
12. *Gentry Craig.* A fully homomorphic encryption scheme, *A dissertation submitted to the department of computer science and the committee on graduate students of Standford University*, 2009.
13. *Regev O.* New lattice-based cryptographic constructions, *J. ACM*, 2004, Vol. 51, No. 6, pp. 899-942.
14. *Regev O.* On lattices, learning with errors, random linear codes, and cryptography, *STOC*, 2005, pp. 84-93.
15. *Rao G.V., Kakulapati V., Purushoththaman M.* Privacy homomorphism in mobile ad hoc networks, *International Journal of Research & Reviews in Computer Science*, 2011.
16. Helib. Available at: https://github.com/homenc/HElib (accessed 01 June 2020).
17. FHEW. Available at: https://github.com/lducas/FHEW (accessed 01 June 2020).
18. *Varnovskiy N.P., Shokurov A.V.* Gomomorfnoe shifrovanie [Homomorphic encryption], *Tr. Instituta sistemnogo programmirovaniya RAN* [Proceedings of the Institute for System Programming RAS], 2007, No. 12, pp. 27-36.
19. *Varnovskiy N.P., Martishin S.A., Khrapchenko M.V., Shokurov A.V.* Porogovye sistemy gomomorfnogo shifrovaniya i zashchita informatsii v oblachnykh vychisleniyakh [Threshold systems of homomorphic encryption and information security in cloud computing], *Programmirovanie* [Programming], 2015, No. 4, pp. 47-51.
20. *Yakovlev M.O.* Zashchishchennyy kal'kulyator. Razrabotka klientskogo komponenta [Protected calculator. Client component development]. Available at: http://pdf.knigi-x.ru/21informatika/ 429422-1-kafedra-sistem-informatiki-vipusknaya-kvalifikacionnaya-rabota-bakalavra-yakovlev-mihail-olegovich-za.php (accessed 01 June 2020).

Статью рекомендовал к опубликованию д.э.н., профессор Е.Н. Тищенко

Бабенко Людмила Климентьевна – Южный федеральный университет; e-mail: blk@tsure.ru; 347928, г. Таганрог, ул. Чехова, 2, корпус "И"; тел.: +79054530191; кафедра безопасности информационных технологий; д.т.н.; профессор.

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

Babenko Lyudmila Kliment'evna – Southern Federal University; e-mail: blk@tsure.ru; Block “I”, 2, Chekhov street, Taganrog, 347928, Russia; phone: +79054530191; the department of information technologies security; dr. of eng. sc.; professor.

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