УДК 004.056.55 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait.2021.1/3373
Поступила в редакцию 12.03.2021
Подписана в печать 26.04.2021

МЕТОД ОБРАБОТКИ ДАННЫХ С УЧЕТОМ ВЗАИМНОГО РАСПОЛОЖЕНИЯ ИНФОРМАЦИОННЫХ БЛОКОВ В МАСШТАБЕ ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА

© 2021 Е. А. Кулешова✉, А. Л. Марухленко, В. П. Добрица, М. О. Таныгин, А. В. Плугатарев

Юго-Западный государственный университет
ул. Челюскинцев, 19/Б, 305004 Курск, Российская Федерация

Аннотация. Развитие информационных технологий предполагает непрерывное совершенствование средств, обеспечивающих обработку и преобразование конфиденциальных данных. Решение подобных задач в масштабе реального времени предполагает совершенствование методов обработки потоковых данных, а также оценку их быстродействия. В статье рассмотрена математическая модель метода преобразования данных, основанная на идее клеточного автомата с плавающим окном. Для исследования быстродействия процесса обработки конфиденциальных данных разработан вариант организации структуры программного модуля с расширенным блоком настроечных параметров, включающим строку активации битовой окрестности обрабатываемых элементов, правило расширения граничных элементов матрицы, определяющее в зависимости от шага работы алгоритма положение соседей обрабатываемого элемента. В статье предложен метод формирования графической зависимости внесенных изменений, косвенно отражающий стойкость метода шифрования и выявляющий соответствие результатов доверительному интервалу. На основе проведенных исследований разработан программный модуль, на базе которого реализовано правило, учитывающее состояние соседних блоков данных для обрабатываемого элемента. Предложенный в статье вариант обработки потоковых данных на базе клеточных автоматов с использованием вычислительных кластеров позволяет оптимизировать скорость обработки потоков данных за счет предварительного этапа оценки соответствия текущего блока пользовательскому шаблону. Использование данного варианта обработки не привело к выходу из доверительного интервала, а распределение бит осталось близким к случайному. В статье проведены экспериментальные исследования на базе разработанного программного модуля, реализующего метод преобразования данных, основанный на идее клеточного автомата с плавающим окном, которые подтвердили полноту и корректность полученных результатов.

Ключевые слова: клеточный автомат, обработка потоков данных, вычислительный кластер, информационная безопасность, многопоточные вычисления, системы защиты конфиденциальной информации, преобразование данных.

ВВЕДЕНИЕ

На сегодняшний день, существует множество задач, требующих вычислительных мощностей для обеспечения быстрой обработки разнородного контента [1–3]. Как правило, подобный контент представлен в виде массивов данных, видеофайлов, корпоративных или личных данных [4, 5]. Стоит принимать во внимание то, что информационные технологии постоянно развиваются, что влечет за собой необходимость совершенствования средств защиты конфиденциальных данных [6, 7]. Согласно положениям международного права, при принятии подобных решений необходимо принимать во внимание правила разграничения доступа и методы оценки

✉ Кулешова Елена Александровна
e-mail: lena.kuleshova.94@mail.ru

защищенности информации [8, 9]. Одним из возможных решений подобной проблемы является построение системы, преобразование данных в которой будет осуществляться по средствам клеточных автоматов, что будет обеспечивать защищенность передаваемой информации, при этом существует возможность объединения высокоскоростных каналов связи.

Выбор клеточных автоматов в качестве платформы для преобразования данных обусловлен их простотой и потенциалом в выполнении сложных вычислений и способностью моделировать сложные системы. С момента появления клеточных автоматов в сфере защиты конфиденциальных данных клеточные автоматы нашли свое место во множестве схем шифрования, либо в качестве их основной основы, либо в качестве одного из компонентов. Это связано с тем, что они обладают той же выразительностью, что и машины Тьюринга.

В данной статье проведен краткий обзор наиболее актуальных систем преобразования данных на основе клеточных автоматов для определения возможных сфер применения и потенциальной полезности клеточных автоматов в области защиты информации. За последние годы простая и гибкая структура клеточных автоматов привлекла внимание исследователей из множества дисциплин, таких как биология, химия, физика, астрономия и математика. Из-за присущих им свойств, клеточные автоматы вполне подходят для аппаратной реализации. Клеточные автоматы были предложены для построения параллельных компьютеров [10, 11], в частности, для реализации некоторых вычислительных операций, таких как арифметика $GF(2^m)$ с конечным полем [12, 13], в качестве генераторов простых чисел [14] и быстрых односторонних хэш-функций [15]. Потенциал клеточных автоматов по созданию высокоскоростных приложений для преобразования данных огромен. Важной особенностью при этом будет являться то, что для пользователя данная система будет иметь вид единого аппаратного ресурса. Решение подобных задач в масштабе реального времени предполагает совершенствование методов обработки потоков конфиденциальных данных.

Целью работы является разработка варианта организации высокоскоростной обработки потоков конфиденциальных данных на основе применения метода обработки битовой последовательности на базе клеточных автоматов в масштабе вычислительного кластера и разработке программного модуля, учитывающего аппаратные особенности автоматизированной системы.

1. ПОСТАНОВКА ЗАДАЧИ

Задача преобразования конфиденциальных данных в контексте предлагаемого метода может быть описана последовательностью следующих этапов: поток необработанных данных построчно записывается в исходную текстовую матрицу; учитывая размер исходных данных, рассчитывается количество строк данной матрицы, расположение блока преобразования исходных данных задается в ручную на этапе выбора режима обработки; движение по матрице исходного текста происходит итеративно (по строкам или столбцам, в зависимости от режима обработки) по принципу плавающего окна до обработки всей матрицы данных.

Важно отметить, что в качестве исходной информации может выступать произвольный поток данных — пакеты сетевого взаимодействия, файлы мультимедиа, архивы или информация в текстовом виде. В случае работы с непрерывными потоками данных параметром можно пренебречь т.к. матрица формируется непрерывно в режиме буферного окна. В случае работы с конечным фрагментом (файлом данных) — требуется формирование матрицы до прямоугольного вида, таким образом, в случае если информационные биты закончились, а последняя строка матрицы сформирована не полностью — производится их заполнение хвостовой последовательностью. Это может быть некоторая функция или статичное заполнение нулями или единицами с учетом активной окрестности. Важным моментом является обеспечение доступа разрабатываемого модуля к носите-

лю данных или сетевому интерфейсу, при этом необходимо учитывать аппаратные особенности системы. Исходя из характеристик производительности современных процессоров, вытекает проблема обеспечения потребностей пользователей, которая не может быть решена при использовании всего одного сервера при его достаточной производительности. На основе вышесказанного, предлагается использовать кластеризацию, так как это позволит реализовать параллельное выполнения задач сразу в нескольких узлах.

2. МАТЕРИАЛЫ И МЕТОДЫ

Рассмотрим более подробно понятие кластеризации в контексте предлагаемого метода. При кластеризации используется несколько взаимосвязанных компьютерных систем. В тоже время они используют общие приложения и для пользователя выглядят как единая система. Слабым звеном при этом является время взаимодействия между узлами, что является причиной большой нагрузки на центральный процессор, что делает невозможным перемещение процессов между ядрами каждого сервера [16]. В качестве варианта решения данной проблемы может быть использован кластер распределения нагрузки (NLB), представленный на рис. 1.

Выбор данной технологии обоснован масштабируемостью и высоким коэффициент готовности. Основные функции кластерных систем — достижение большей вычислительной мощности и обеспечение повышенной надежности компьютерных систем. Анализ текущего состояния вычислительных систем подтверждает стремительное развитие в области многопоточных вычислений. Прослеживается позитивная динамика роста производительности вычислительных кластеров. В годы с 2006 по 2020 средняя мощность увеличилась более чем в 500 раз, то есть график роста мощности имеет экспоненциальный характер. Таким образом, данную технологию можно считать перспективной в условиях развития современных систем.

2.1. Математическая модель

Чтобы проиллюстрировать работу метода, рассмотрим работу с двумерной матрицей. Размер блока шифрования $(m_1 \times m_2)$ может быть установлен произвольно, а количество столбцов матрицы определяется количеством блоков, записанных в алфавите $A = \{0,1\}$. Количество строк матрицы определяется размером исходных данных, а в случае сетевого потока зависит от сеанса взаимодействия пользователей компьютерной сети [19]. Клеточным автоматом с плавающим окном называют совокупность (1):

$$CA_o = \langle Z^n, (N_1, ..., N_n), A, (m_1, ..., m_n), \Psi, R \rangle, \quad (1)$$

где Z^n — размерность клеточного автомата $n = 1, 2, 3$; $(N_1, ..., N_n)$ — размер таблицы; $(m_1, ..., m_n)$ — размер блока преобразования данных; Ψ — таблица функций переходов; R — маршрут обхода блока преобразования данных.

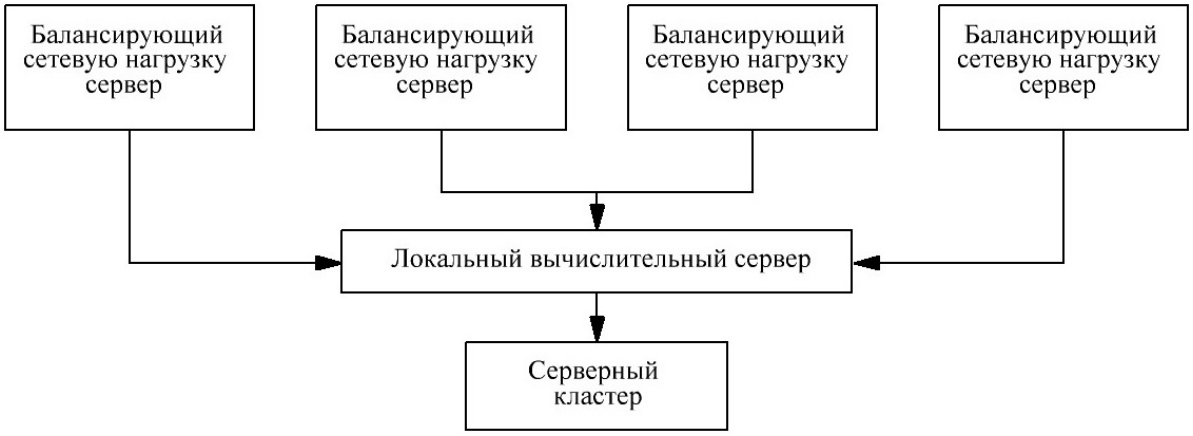

Рис. 1. Схема работы Network Load Balancing
[Fig. 1. Scheme of Network Load Balancing]

Обработка текущего элемента заключается в применении логической операции «исключающее или» к множеству ячеек, принадлежащих активной окрестности обрабатываемого элемента матрицы. Наиболее обычной окрестностью для одномерного КА является окрестность первых соседей, и она состоит просто из центральной клетки окрестности и ее правых и левых соседей: $\{-1,0,1\}$. Расширяя это обозначение для радиуса r, множество окрестностей задается как $\{-r,-r+1,...,0,...,r-1,r\}$. Для двумерного КА окрестности фон Неймана и Мура являются двумя наиболее распространенными формами соседства с радиусом 2 [17]. В данном случае мы рассматриваем окрестность Мура второго порядка.

Окрестность включает центральный (обрабатываемый) и 4 элемента из окружения. Для удобства ввода ключей и взаимодействия пользователей системы ключ задается последовательностью идентификаторов, определяющих расположение элемента в окрестности β. Допускается использование латинских букв и относительных индексов. Формальная запись β может быть упрощенно представлена последовательностью буквенных обозначений элементов окрестности или математически (пример математической записи представлен в виде формулы 2).

$$\beta = m_{x,y} \oplus m_{x-1,y-1} \oplus m_{x+1,y-2} \oplus \oplus m_{x+2,y+1} \oplus m_{x-1,y+2}. \quad (2)$$

Математическая запись является рекомендуемой т. к. учитывает относительное расположение элементов для окрестности большого порядка (удаленности от обрабатываемого элемента). Сеанс шифрования предполагает последовательный обход элементов матрицы, являющейся отражением данных в открытом виде и применение к ним правила β. Дешифрование включает в себя обратный обход матрицы. Это позволяет обеспечить обратимость преобразования и обеспечивает потенциальную возможность проведения нескольких раундов обработки для повышения стойкости [18–19]. Рассмотренная модель предполагает, что не все биты должны быть изменены т.к. это привело бы к инверсии содержимого файла. Важно обеспечивать равномерное внесение изменений в масштабах общего потока данных. Минимальный порог инверсий не должен быть менее 15 %. Это связано с абсолютным значением внутри байта информационного потока. Оптимальным (доверительным) интервалом считается диапазон 40–60 %, который дает изменения в масштабе всех байтов при поддержании равномерности распределения инвертированных бит. Превышение указанного интервала снижает стойкость предложенного метода т. к. инверсия входного потока покажет злоумышленнику исходные фрагменты данных.

2.2. Численное моделирование

Перейдем к рассмотрению программной реализации. На рис. 2 показана панель для выбора файла для обработки и задания правила дополнения (в верхней части окна программы). Слева расположены опции ведения протокола (в отладочных целях могут выводиться исходные байты в различных системах счисления, исходная и результирующая матрицы) и удобства тестирования модуля (возможность автоматического сохранения файла и подстановки результата в качестве входного потока данных после обработки). Справа расположены настроечные параметры клеточного автомата, которые включа-

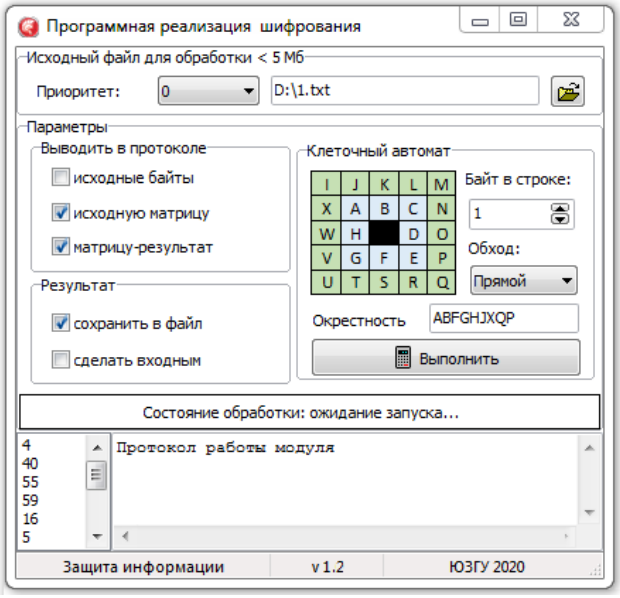

Рис. 2. Интерфейс программного модуля [Fig. 2. Program Module Interface]

ют задание активной окрестности, ширины матрицы и порядок обхода. Ниже находится индикатор прогресса текущей обработки. Актуален при работе с файлами, размер которых превышает 1 Мб.

Для анализа распределения измененных в результате шифрования на уровне бит сформируем матрицу, являющуюся разностью исходной и зашифрованной матрицы (рис. 3). На рис. 3 представлен фрагмент получившейся в результате суперпозиции исходной и обработанной матриц, содержащей 100 столбцов. Значением элемента матрицы будут значения $\{-1, 0, 1\}$. Белым клеткам соответствует 0 (значение бита данных в обработанной матрице не изменилось относительно исходной матрицы), горизонтальная и вертикальная штриховки соответствует значениям –1 и 1 соответственно (значение бита данных в обработанной матрице изменилось относительно исходной матрицы).

Упрощенная визуализация в виде поверхности, показана на рис. 4. Для наглядности мы взяли значение искажения по модулю. Здесь точки отличия приподнимаются от основного уровня и демонстрируют распределенность внесенных изменений. На схеме показано пятьдесят рядов, соответствующих строкам матрицы, ось из ста делений соответствует числу столбцов матрицы (ширина таблицы). Наклонные грани показывают равномерный переход между состояниями.

Взгляд на фрагмент графика «снизу» (горизонтальная проекция) совпадает с рис. 3. В соответствии с полученными результатами можно сделать вывод, что обработанная матрица содержит 50 % изменений на уровне битов и 100 % изменений на уровне байтов, т. е. результат обработки не схож с инверсией и не может быть понятен злоумышленнику без обратного преобразования, предполагающего знание или подбор ключевых параметров.

Полученные показатели подтверждают высокий уровень стойкости рассмотренного метода преобразования данных. В ходе дальнейших исследований планируется рассмотрение внесенных изменений не только с установлением факта изменения значения бита, но и с учетом граничных значений.

3. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

В результате проведенных исследований разработан вариант организации многопоточных вычислений, позволяющий при использовании вычислительных кластеров повысить скорость обработки непрерывных потоков

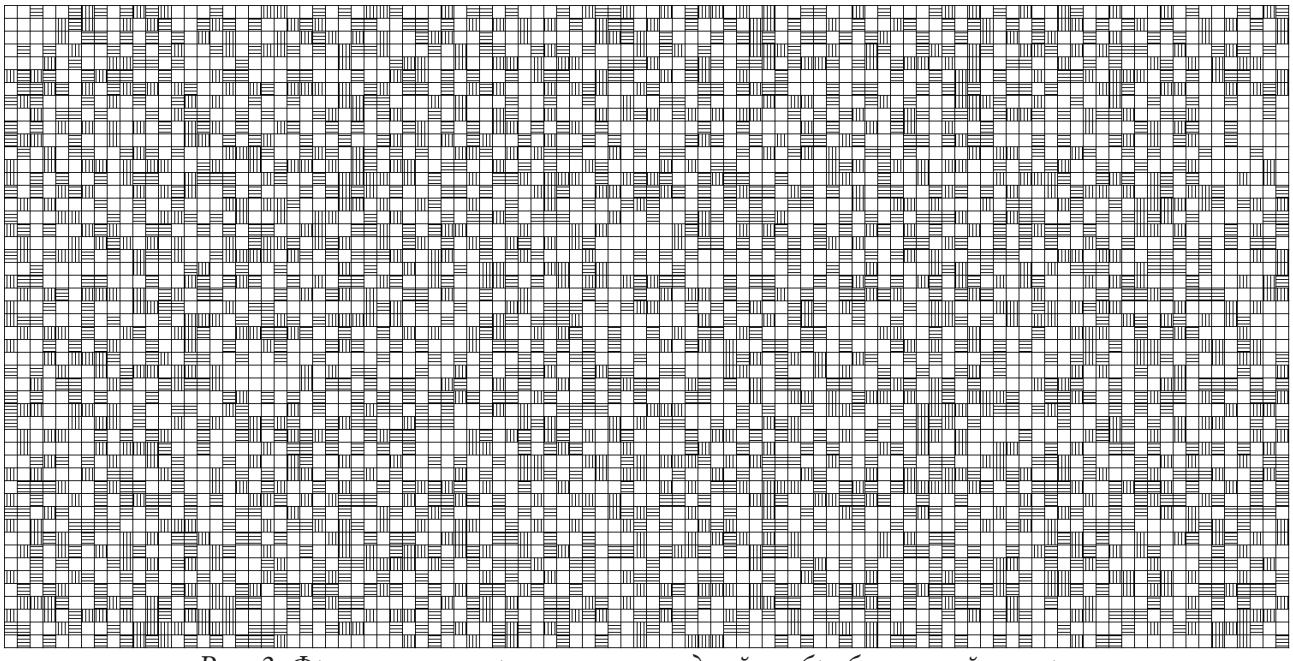

Рис. 3. Фрагмент суперпозиции исходной и обработанной матриц
[Fig. 3. Fragment of Superposition of the Original and Processed matrices]

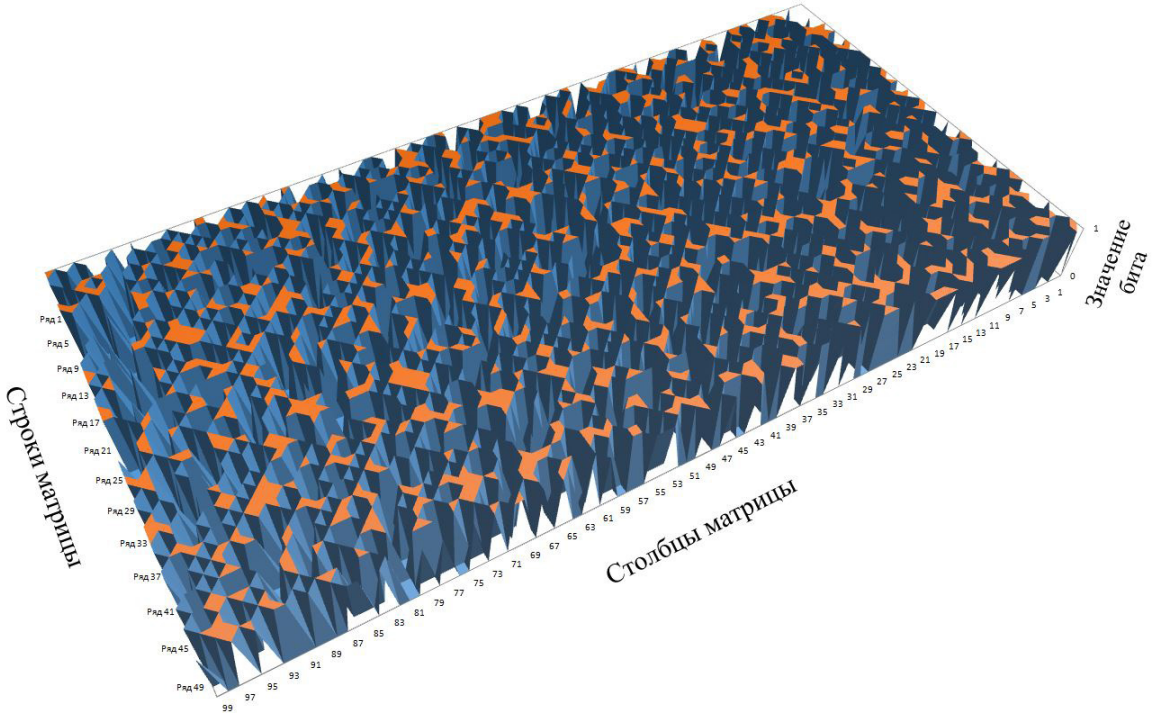

Рис. 4. Фрагмент суперпозиции матриц в виде поверхности
[Fig. 4. Fragment of the Superposition of Matrices in the Form of a Surface]

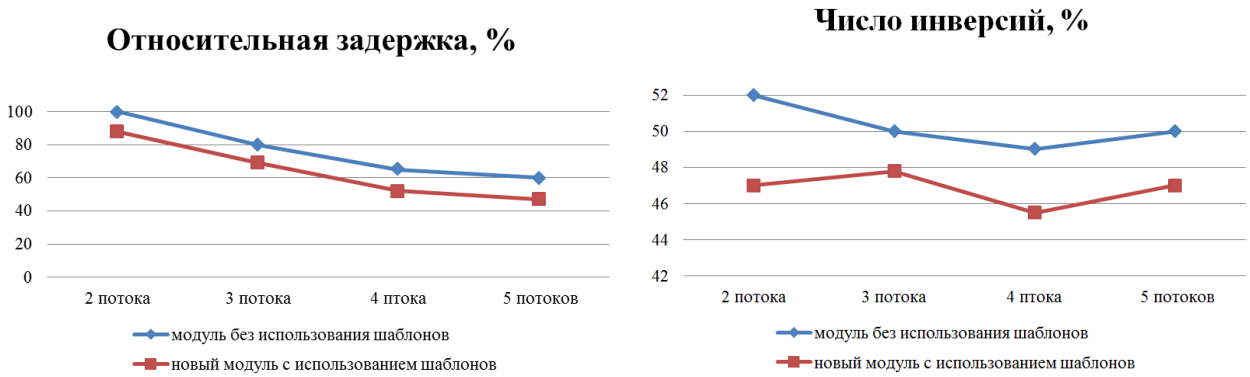

Рис. 5. Сравнительная диаграмма относительных скоростей и числа инверсий
[Fig. 5. Comparative Diagram of Relative Velocities and Number of Inversions]

данных, что особенно актуально при сетевом взаимодействии абонентов. На рис. 5 представлена сравнительная диаграмма относительных скоростей и числа инверсий (процент изменений) для программного модуля, рассмотренного в работе [20], и модуля, предлагаемого в данной работе. По оси ординат указана относительная задержка для различного числа потоков данных, при этом значение 100% соответствует максимальному времени преобразования данных. На второй части рис. 5 на оси ординат указано число инверсий данных, полученное в результате их преобразования для различного числа потоков.

Анализ полученных результатов показал, что в предложенном варианте увеличено быстродействие преобразований за счет предварительного этапа оценки соответствия текущего блока пользовательскому шаблону. Если к блоку применим шаблон, то активируется клеточная функция. Данная модификация позволила повысить быстродействие метода до 13 %, при этом распределение бит осталось близким к случайному и носит равномерный характер, а уменьшение процента инверсий не привело к выходу из доверительного интервала.

ЗАКЛЮЧЕНИЕ

При проведении данного исследования была разработана схема организации высокоскоростной обработки конфиденциальных данных, позволяющая при использовании вычислительных кластеров повысить скорость обработки непрерывных потоков данных, что особенно актуально при сетевом взаимодействии абонентов. Для реализации экспериментальной части исследования, полагаясь на разработанные теоретические положения, была проведена программная реализация модуля, реализующего правило, учитывающее состояние соседних блоков данных для обрабатываемого элемента. Предложен метод формирования графической зависимости внесенных изменений, косвенно отражающий стойкость метода шифрования и выявляющий соответствие результатов доверительному интервалу. Совокупность полученных практических результатов подтверждают полноту и корректность разработанных теоретических положений.

БЛАГОДАРНОСТИ

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-31-90069.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Марухленко, А. Л.* Программный комплекс для моделирования процесса передачи и обработки сетевых потоков данных / П. С. Мирзаханов, А. Л. Марухленко // Известия ЮЗГУ. – 2012. – № 2-3. – С. 175–180.

2. *Fúster, A.* Cellular Automata (CA) in Stream Ciphers / A. Fúster // Contemporary Mathemathics. – 2009. – V. 477. – P. 1–21. DOI: 10.1090 / conm / 477/09301.

3. *Tomeu, A.* Parallel Bit-Stream Cipher with Cellular Automata / A. Salguero, M. Capel // Annals of Multicore and GPU Programming. – 2014. – V. 1. – P. 10–18.

4. *Kale, V.* Parallel Computing Architectures and APIs: IoT Big Data Stream Processing / V. Kale. – 1st Edition. – Chapman and Hall : CRC, 2019. – 380 p.

5. *Muhammad, B.* Real-time data processing scheme using big data analytics in internet of things / B. Muhammad, F. Arif // J. of Ambient Intelligence and Humanized Computing. – 2019. – V. 10. – P. 4167–4177. DOI: 10.1007 / s12652-018-0820-5.

6. *Torisawa, T.* Nonlinear Psedurandom Sequences / T. Torisawa, T. Komatsuzaki, Y. Saikawa // Proceedings of 8th International Conference on CA for Research and Insdustry. – 2008. – P. 471–478.

7. *Petrica, L.* FPGA optimized CA random number generator / L. Petrica // J. Parallel and Distributed Computing. – 2018. – V. 111. – P. 251–259.

8. *Harper, A.* Gray Hat Hacking: The Ethical Hacker's Handbook / A. Harper, D. Regalado and oth. – McGraw-Hill Education. – 2018. – 640 p.

9. *Марухленко, Л. О.* Комплексная оценка информационной безопасности объекта с применением математической модели для расчета показателей риска / Л. О. Марухленко, М. А. Ефремов и др. // Известия ЮЗГУ. – 2018. – Т. 8, № 4 (29). – С. 34–40.

10. *Bandini, S.* Cellular Automata: from a theoretical parallel computational model / S. Bandini, G. Mauri, R. Serra // Parallel Computing. – 2001. – 27(5). – P. 539–553. DOI: 10.1016 / S0167-8191 (00) 00076-4.

11. *Maniatty, W. A.* Progress in computer research / W. A. Maniatty, B. K. Szymanski, T. Caraco // Parallel Computing with Generalized CA. – 2001. – Nova Science Publishers : USA. – 2001. – P. 51–75.

12. *Kim, H. S.* Cellular Automata based multiplier for public-key cryptosystem / H. S. Kim, K. Y. Yoo // Security in Pervasive Computing. – 2004. – V. 2802 of LNCS. – Springer Berlin Heidelberg. – P. 227–236.

13. *Li, H.* Efficient CA based versatile multiplier for GF(2m) / H. Li, C.N. Zhang //

Journal of Information Science and Engineering. – 2002. – V. 18. – P. 497–502. DOI: 10.1016/0012-365X(92)90582-Z.

14. *Sukhinin, B. M.* High-speed pseudorandom sequence generators based on Cellular Automata / B. M. Sukhinin // Prikl. Diskr. Mat. – 2010. –2(8). – P. 34–41.

15. *Tanygin, M. O.* A method of the transmitted blocks information integrity control / M. O. Tanygi, H. Y. Alshaeaa, E. A. Kuleshova // Radio Electronics, Computer Science, Control. – 2020. – № 1. – P. 181–189. DOI: https://doi.org/10.15588/1607-3274-2020-1-18.

16. *Franti, E.* Cellular Automata Encryption System / E. Franti, M. Dascalu // The Fifth International Conference on Engineering Computational Technology, 2021. – P. 283–297.

17. *Таныгин, М. О.* Установление доверительного канала обмена данными между источником и приёмником информации с помощью модифицированного метода одноразовых паролей / М. О. Таныгин, Х. Я. Алшаиа и др. // Известия ЮЗГУ. – 2018. – Т. 8, № 4 (29). – С. 63–71.

18. *Ключарёв, П. Г.* Блочные шифры, основанные на обобщённых клеточных автоматах / П. Г. Ключарёв // Наука и образование: электронное научно-техническое издание. – 2012. – № 12. – С. 361–374. DOI: 10.7463/0113.0517543.

19. *Росошек, С. К.* Криптосистемы клеточных автоматов / С. К. Росошек, С. И. Боровков, О. О. Евсютин // Прикладная дискретная математика. – 2008. – № 1. – С. 43–49. DOI 10.17223/20710410/1/8.

20. *Марухленко, А. Л.* Вариант организации многопоточной обработки конфиденциальных данных на базе клеточных автоматов / А. Л. Марухленко, А. В. Плугатарев, М. О. Таныгин, Л. О. Марухленко, Д. О. Бобынцев // Известия ЮЗГУ. – 2019. – Т. 23, № 3. – С. 100–112. DOI: 10.21869/2223-1560-2019-23-3-100-112.

Кулешова Елена Александровна — аспирант 4-го года обучения кафедры информацион-ной безопасности Юго-Западного государственного университета, г. Курск.
E-mail: lena.kuleshova.94@mail.ru
ORCID iD: https://orcid.org/0000–0002–8270–564X

Марухленко Анатолий Леонидович — канд. техн. наук, доц., доцент кафедры информаци-онной безопасности Юго-Западного государственного университета, г. Курск.
E-mail: proxy33@mail.ru
ORCID iD: https://orcid.org/0000–0002–3575–924X

Добрица Вячеслав Порфирьевич — д-р физ.-мат. наук, проф., профессор кафедры инфор-мационной безопасности Юго-Западного государственного университета, г. Курск.
E-mail: dobritsa@mail.ru
ORCID iD: https://orcid.org/0000–0001–7533–3684

Таныгин Максим Олегович — канд. техн. наук, доц., заведующий кафедрой кафедры ин-формационной безопасности Юго-Западного государственного университета, г. Курск.
E-mail: tanygin@yandex.ru
ORCID iD: https://orcid.org/0000–0002–4099–1414

Плугатарев Алексей Владимирович — аспирант 2-го года обучения кафедры информаци-онной безопасности Юго-Западного государственного университета, г. Курск.
E-mail: aplugatarev@bk.ru
ORCID iD: https://orcid.org/0000-0002-8549-4382

DOI: https://doi.org/10.17308/sait.2021.1/3373 ISSN 1995-5499
Received 12.03.2021
Accepted 26.04.2021

A DATA PROCESSING METHOD THAT TAKES INTO ACCOUNT THE MUTUAL ARRANGEMENT OF INFORMATION BLOCKS IN A COMPUTING CLUSTER

© 2021 E. A. Kuleshova✉, A. L. Marukhlenko, V. P. Dobritsa, M. O. Tanygin, A. V. Plugatarev

Southwest State University
19/B, Chelyuskintsev Street, 305004 Kursk, Russian Federation

Annotation. The development of information technologies entails the continuous improvement of tools that ensure the processing and transformation of confidential data. To solve such problems in real time, it is necessary to enhance the methods of data stream processing, as well as to evaluate their performance. The article discusses a mathematical model for the data transformation method based on the idea of cellular automaton with a floating window. To study the speed of the processing of confidential data, we developed a software module with a specific structure that includes a wider range of tuning parameters. These parameters include the activation string of the bit neighbourhood of the processed elements and the expansion rule for the boundary elements of the matrix which locates the neighbours of the processed element depending on the step of the algorithm. The article suggests a way to create a dependence diagram of the introduced changes, which indirectly reflects the strength of the encryption method and reveals whether the results comply with the confidence interval. Based on the results of the research, we developed a software unit implementing a rule that takes into account the state of the neighbouring data blocks for the processed element. The suggested approach to stream processing based on cellular automata and computing clusters allows us to optimise the speed of the data stream processing. This is possible due to the preliminary stage, when we assess the compliance of the current block to the custom template. At the same time, the results remained within the confidence interval and the bit distribution remained close to random. The article also describes the experiments we performed using the developed software unit that implements the data transformation method based on the idea of a cellular automaton with a floating window. The experimental study confirmed the completeness and correctness of the results obtained.
Keywords: cellular automaton, data stream processing, computing cluster, information security, multithreading, systems protecting the confidential information, data transformation.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. *Marukhlenko A. L. & Mirzakhanov P. S.* (2012) Software complex for modeling the process of transmission and processing of network data streams // Bulletin of the Southwest State University. 2-3. P. 175–180.

2. *Fúster A*. (2009) Cellular Automata (CA) in Stream Ciphers // Contemporary Mathemathics. 477. P. 1–21. DOI: 10.1090 / conm / 477/09301.

3. *Tomeu A., Salguero A. & Capel M.* (2014) Parallel Bit-Stream Cipher with Cellular Automata // Annals of Multicore and GPU Programming. 1. 10–18.

4. *Kale V.* (2019) Parallel Computing Architectures and APIs: IoT Big Data Stream Processing. Chapman and Hall, CRC.

5. *Muhammad B. & Arif F.* (2019) Real-time data pro-cessing scheme using big data analytics in inter-net of things // J. of Ambient In-telligence and Humanized Computing. 10. P. 4167–4177. DOI: 10.1007 / s12652-018-0820-5.

✉ Kuleshova Elena A.
e-mail: lena.kuleshova.94@mail.ru

6. *Torisawa T., Komatsuzaki T. & Saikawa Y.* (2008) Nonlinear Psedurandom Sequences // Proceedings of the 8th International Conference on CA for Research and Insdustry. P. 471–478.

7. *Petrica L.* (2018) FPGA optimized CA random number generator // J. Parallel and Distributed Computing. 111. P. 251–259.

8. *Harper A., Regalado D. [et al.]* (2018) Gray Hat Hacking: The Eth-ical Hacker's Handbook. McGraw-Hill Education.

9. *Marukhlenko L. O., Efremov M. A.* [et al.] (2018) Comprehensive assessment of the information security of an object using a mathematical model for calculating risk indicators // Bulletin of the Southwest State University. 4 (29). P. 34–40.

10. *Bandini S., Mauri G. & Serra R.* (2001) Cellular Automata: from a theoretical parallel computational model // Parallel Computing. 27 (5). P. 539–553. DOI: 10.1016 / S0167-8191 (00) 00076-4.

11. *Maniatty W. A., Szymanski B. K. & Caraco T.* (2001) Progress in computer research // Parallel Computing with Generalized CA. Nova Science Publishers: USA. P. 51–75.

12. *Kim H. S. & Yoo K. Y.* (2004) Cellu-lar Automata based multiplier for public-key cryptosystem // Security in Pervasive Computing. V. 2802 of LNCS. Springer Berlin Heidelberg. P. 227–236.

13. *Li H. & Zhang C. N.* (2002) Efficient CA based versatile multiplier for GF (2m) // Journal of Information Science and Engineering. 18. P. 497–502. DOI: 10.1016/0012-365X(92)90582-Z.

14. *Sukhinin B. M.* (2010) High-speed pseudorandom sequence generators based on Cellular Automata // Prikl. Diskr. Mat. 2 (8). P. 34–41.

15. *Tanygin M. O., Alshaeaa H. Y. & Kuleshova E. A.* (2020) A method of the transmitted blocks information integrity control // Radio Electronics, Computer Science, Control. 1. P. 181–189. DOI: https://doi.org/10.15588/1607-3274-2020-1-18.

16. *Franti E. & Dascalu M.* (2021) Cellular Automata Encryption System // The Fifth International Conference on Engineering Com-putational Technology. P. 283–297.

17. *Tanygin M. O., Alshaia H. Y. [et al.]* (2018) Establishing a trusted channel for data exchange between the source and the receiver of information using a modified method of one-time passwords // Bulletin of the Southwest State University. 4 (29). P. 63–71.

18. *Klyucharyov P. G.* (2012) Block ciphers based on generalized cellular automata // Science and education: electronic scientific and technical publication. 12. P. 361–374. DOI: 10.7463/0113.0517543.

19. *Rososhek S. K., Borovkov S. I. & Evsyutin O. O.* (2008) Cryptosystems of cellular automata // Applied Discrete Mathematics. 1. P. 43–49. DOI 10.17223/20710410/1/8.

20. *Marukhlenko A. L., Plugatarev A. V., Tanygin M. O., Marukhlenko L. O. & Bobyntsev D. O.* (2019) Variant of organizing multithreaded processing of confidential data based on cellular automata // Bulletin of the Southwest State University. 3. P. 100–112. DOI: 10.21869/2223-1560-2019-23-3-100-112.

Kuleshova Elena A. — 4th year postgraduate student, Department of Information Security, Southwest State University, Kursk.
E-mail: lena.kuleshova.94@mail.ru
ORCID iD: https://orcid.org/0000–0002–8270–564X

Marukhlenko Anatoly L. — PhD in Technical Sciences, Associate Professor, Department of Information Security, Southwest State University, Kursk.
E-mail: proxy33@mail.ru
ORCID iD: https://orcid.org/0000–0002–3575–924X

Dobritsa Vyacheslav P. — DSc in Physics and Mathematics, Professor, Department of Information Security, Southwest State University, Kursk.
E-mail: dobritsa@mail.ru
ORCID iD: https://orcid.org/0000–0001–7533–3684

Tanygin Maxim O. — PhD in Technical Sciences, Associate Professor, Head of the Department of Information Security, Southwest State University, Kursk.
E-mail: tanygin@yandex.ru
ORCID iD: https://orcid.org/0000–0002–4099–1414,

Plugatarev Alexey V. — 2nd year postgraduate student, Department of Information Security, Southwest State University, Kursk.
E-mail: aplugatarev@bk.ru
ORCID iD: https://orcid.org/0000-0002-8549-4382