Оригинальная статья / Original article

https://doi.org/10.21869/2223-1560-2021-25-2-51-64

Рекурсивный алгоритм формирования структурированных множеств информационных блоков для повышения скорости выполнения процедур определения их источника

М. О. Таныгин [1] ✉, Х. Я. А. Алшаиа [1], В. П. Добрица [1], О. Г. Добросердов [1]

[1] Юго-Западный государственный университет
ул. 50 лет Октября 94, г. Курск 305040, Российская Федерация

✉ e-mail: tanygin@yandex.com

Резюме

Цель исследования. *Для некоторых классов современных информационных систем выполнение известных алгоритмов преобразования данных при проведении процедур идентификации и аутентификации источника данных затруднено из-за ограничений на длительность полного цикла обработки данных. В статье рассматривается алгоритм выполнения процедуры определения источников для группы блоков данных, позволяющий за счёт изменения порядка выполнения типовых операций обнаруживать возникающие ошибки раньше, чем обычные итерационные алгоритмы формирования древовидных структур. Целью работы является снижение вычислительных и ресурсных затрат на выполнение приёмником процедур идентификации источников групп блоков данных, каждый из которых обладает размером, не превышающим несколько байтов .*

Методы. *В основе рассматриваемого метода идентификации лежит рекурсивный алгоритм формирования древовидной структуры из блоков поступающей информации и последующий анализ ветвей такой структуры. Это позволяет определить подмножество блоков, сформированных целевым источником. В статье приведено формальное математическое описание алгоритма, а также представлены результаты проведённого имитационного моделирования процедур определения источника. При этом характеристики рекурсивного алгоритма сравнивались с аналогичными, полученными для известного итерационного.*

Результаты. *Полученные в результате имитационного моделирования зависимости между средним числом типовых операций сравнения хешей, средним числом сформированных ветвей древовидной структуры блоков и числом принятых блоков позволили сформулировать критерии применения рекурсивного и итерационного алгоритмов.*

Заключение. *В работе показано, что применение рекурсивного алгоритма формирования древовидной структуры кадров позволяет снизить среднее число выполняемых приёмником типовых операций на 5 – 10 % и уменьшить затраты памяти для хранения ветвей такой структуры на величину до 30% .*

Ключевые слова: *передача данных; приёмник сообщений; идентификация; рекурсия; вычислительная сложность; быстродействие; имитационное моделирование.*

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

Для цитирования: Рекурсивный алгоритм формирования структурированных множеств информационных блоков для повышения скорости выполнения процедур определения их источника / М.О. Таныгин, Х.Я.А. Алшаиа, В.П. Добрица, О.Г. Добросердов // Известия Юго-Западного государственного университета. 2021; 25(2): 51-64. https://doi.org/10.21869/2223-1560-2021-25-2-51-64.

Поступила в редакцию 02.04.2021 *Подписана в печать 27.04.2021* *Опубликована 24.08.2021*

Recursive Algorithm for Forming Structured Sets of Information Blocks to Increase the Speed of Their Source Determination Procedures

Maxim O. Tanygin [1] ✉, Hayder Y. A. Alshaea [1], Vyacheslav P. Dobritsa [1], Oleg G. Dobroserdov [1]

[1] Southwest State University
50 Let Oktyabrya str. 94, Kursk 305040, Russian Federation

✉ e-mail: tanygin@yandex.com

Abstract

Purpose of research. *In some classes of information systems, it is impossible to use well-known algorithms for the identification and the authentication the data blocks sources. The reason for this is the duration of the full data processing cycle. The article considers the original algorithm for determining sources for a group of data blocks. It allows you to detect errors faster than the usual iterative algorithm for forming tree structures of blocks by changing the order of base operations. The purpose of the work is to reduce the computational and resource costs for the receiver to perform identification the sources of blocks, each of them has a size not exceeding a few bytes.*

Methods. *The identification method based on the forming a tree structure of incoming information blocks and subsequent analysis of tree branches. It allows selecting a chain of blocks formed by the target source. In the article the formal description of the algorithm is given. The results of the simulation of the procedures for determining the source are presented. In this case, the characteristics of the recursive algorithm were compared with those obtained for the known iterative one.*

Results. *The relationships between the average number of typical hash comparison operations, the average number of tree structure branches, and the number of accepted blocks obtained as a result of simulation modeling. This made it possible to determine the conditions for applying the recursive and iterative algorithms.*

Conclusion. *It is shown that the using of a recursive algorithm for forming a tree structure of frames can reduce the average number of base operations performed by the receiver by 5-10 % and reduce the memory cost for storing tree structure branches by up to 30%.*

Keywords: *data transmission; message receiver; identification; recursion; computational complexity; performance; simulation.*

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

For citation: Tanygin M. O., Alshaea H. Y. A., Dobritsa V. P., Dobroserdov O. G. Recursive Algorithm for Forming Structured Sets of Information Blocks to Increase the Speed of Their Source Determination Procedures. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2021; 25(2): 51-64 (In Russ.). https://doi.org/ 10.21869/2223-1560-2021-25-2-51-64.

Received 02.04.2021 *Accepted 27.04.2021* *Published 24.08.2021*

***

Введение

Взаимная идентификация и аутентификация является неотъемлемой частью любого протокола информационного обмена по открытым каналам связи. Алгоритмы, используемые в этих процедурах, подразумевают выполнение множества арифметико – логических операций над передаваемыми данными для обеспечения требуемой достоверности [1]. При этом зачастую требование по достоверности аутентификации вступает в противоречие с требованием по скорости её выполнения [2, 3]. Особенно это актуально для современных роботизированных автономных систем, где важна скорость обработки управляющей информации и энергетические затраты, которыми сопровождается такая обработка. Последнее ограничение, например, существенно влияет на выбор алгоритмов криптографических преобразований в современных беспилотных летающих аппаратах [4], где чрезвычайно важны весовые характеристики вычислительного блока и элементов питания [5]. Кроме того, Это обуславливает актуальность задачи поиска новых методов и алгоритмов и технических решений для выполнения процедур идентификации и аутентификации, которые бы обеспечивали высокую скорость обработки информации, и, что особенно важно, возможность адаптировать целевые характеристики процедур информационного обмена в зависимости от реальной интенсивности информационного потока между удалёнными субъектами.

Материалы и методы

Исследуемым методом аутентификации удалённых субъектов, обменивающихся данными по протоколам с ограниченным размером кадра информации, является метод, основанный на формировании из кадров, обрабатываемых приёмником, цепочек, связанных друг с другом результатом выполнения над их данными различных операций [6 – 9]. Так как в реальных условиях приёмник обрабатывает данные не только от целевого источника, для которого выполняется аутентификация, но и данные от десятков и даже сотен других источников, то возможно возникновение ошибок, связанных с возможностью случайного совпадения хешей блоков разных источников [10, 11]. При этом формируется множество $W = \{w_1, \dots w_N\}$ цепочек, лишь одна из которых представлена блоками целевого источника, а остальные – блоками произвольного множества источников.

Если объединить повторяющиеся элементы таких N цепочек, то можно получить древовидную структуру, показанную на рис. 1.

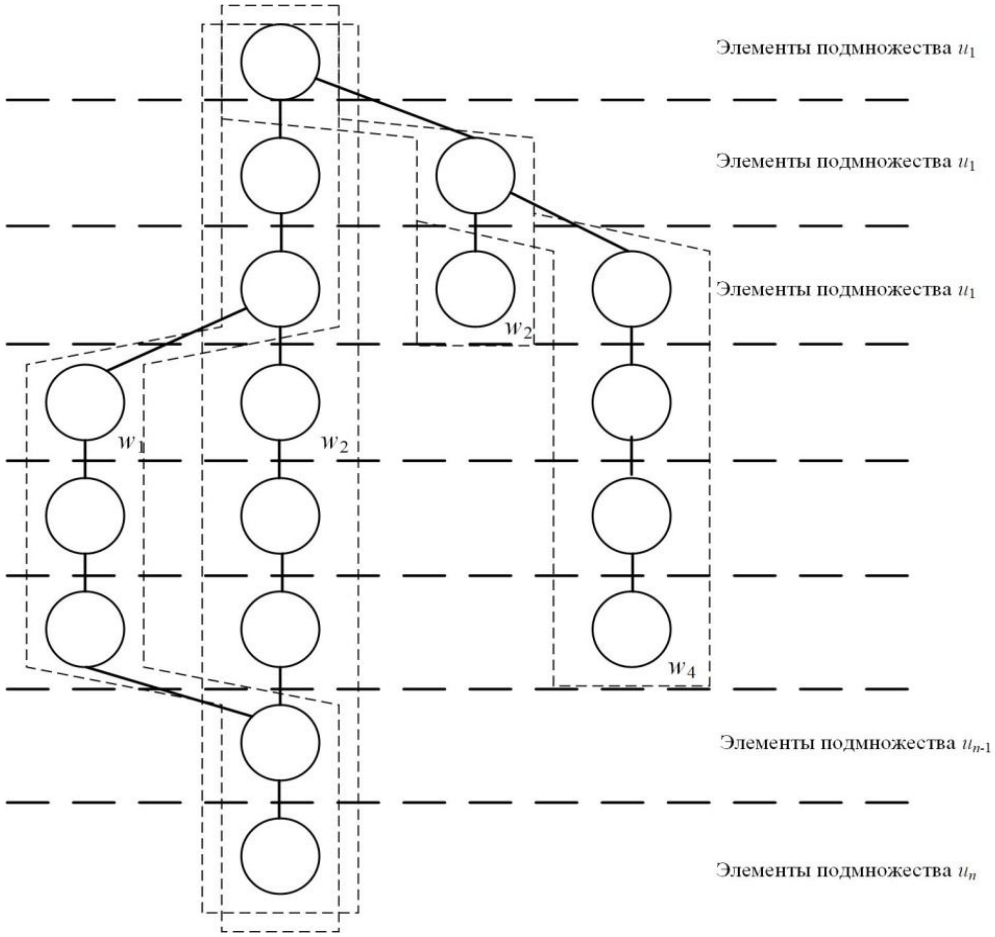

Рис. 1. Схема древовидной структуры блоков данных

Fig. 1. Tree structure of data blocs

Критерием того, что цепочка w_i сформирована целевым источником, является то, что её длина равна некоторому значению n – параметру, который определяется протоколом передачи блоков данных и известен и приёмнику и отправителю. В случае единственности такой ветви, она признаётся множеством блоков от целевого источника. Во всех остальных случаях (несколько цепочек удовлетворяют указанному выше условию, или не удовлетворяет ни одна) считается, что блоки переданы с ошибкой.

Предваряет процедуру анализа разбиение множества U блоков данных на непересекающиеся подмножества u_1, u_2, …, u_n блоков, с различным индексом – позицией блока в цепочке сообщений:

$$\forall s \in U, f^{\text{ind}}(s) = i \to s \in u_i, i = \overline{1,n} \quad (1)$$

где f^{ind} – операция выделения индекса блока.

Кроме вышеописанной, базовыми для выполнения процедуры построения цепочек информационных блоков являются следующие операции: f^{sh} – выделения хеша пакета и F^{hash} – вычисления хеш–функции из данных.

Разбиение всего множества буферизированных блоков на подмножества можно выполнять на этапе предобработки данных блока, проверки их целостности, и она не оказывает значи-

тельного воздействия на трудоёмкость процедуры определения источника [12].

Дальнейшая процедура формирования древовидной структуры может быть реализована двумя способами:

– выполнением итераций по добавлению в структуру элементов последовательно из множеств u_1, u_2, …, u_n , в результате формирование дерева происходит «сверху вниз»;

– прохождением каждой ветви древовидной структуры до её окончания, путём последовательного формирования каждого подмножества множества W, в результате формирование дерева происходит «слева направо» [13].

Для реализации первого метода можно использовать последовательный перебор всего множества блоков данных с включением очередного блока на соответствующую позицию в дереве. Его особенностью является то, что проверка длины ветвей значению n возможна только после окончания формирования древовидной структуры информационных блоков, описываемой множеством W.

Для реализации второго метода требуется рекурсивный алгоритм прохода дерева в глубь, который сложнее в аппаратной реализации, но обладает преимуществами, которые будут рассмотрены ниже. В его реализации подмножества u_1, u_2, …, u_n передаются в качестве параметров в рекурсивную процедуру формирования цепочек. В качестве параметров такой процедуры выступают также:

i_{chain} – текущий номер ветви;

i_{ind} – текущий анализируемый индекс;

k – номер анализируемого блока в каком-либо из подмножеств u_1, u_2, …, u_n.

На начальном этапе N=1, $i_{\text{chain}} = 1$, $i_{\text{ind}} = 1$, $k = 1$ $w_1 = \{s_{\text{start}}\}$, где s_{start} – стартовый блок цепочки блоков источника. Считаем, что он однозначно выделяется из множества анализируемых приёмником блоков [14].

Сам рекурсивный алгоритм состоит из следующих основных этапов:

1. Получаем значения $\{i_{\text{chain}}, i_{\text{ind}}, k\}$. Обнуляем счётчик совпадений m.

2. Повторяем цикл от $j = 0$ до $j = \left|u_{i_{\text{ind}}}\right|$ с шагом 1. По окончании переходим к пункту 10.

3. Если $F^{\text{hash}}\left(u^{\langle k\rangle}_{i_{\text{ind}}+1}\right) = f^{\text{hash}}\left(u^{\langle j\rangle}_{i_{\text{ind}}}\right)$, то переходим пункту 4, иначе к пункту 9.

4. Инкрементируем m.

5. Если m>1, то переходим к пункту 6, иначе – к пункту 8

6. Инкрементируем N, формируем множество $w_N = \{ w^{\langle 1\rangle}_{i_{\text{chain}}}, \ldots, w^{\langle l\rangle}_{i_{\text{chain}}} \}$, $l = \left|w_{i_{\text{chain}}}\right|$, $w^{\langle l+1\rangle}_N = u^{\langle k\rangle}_{i_{\text{ind}}+1}$.

7. Если $i_{\text{ind}} > 1$, то вызываем рекурсивную процедуру с параметрами $\{N, i_{\text{ind}} + 1, j\}$. Переходим к пункту 9.

8. Вызываем рекурсивную процедуру с параметрами $\{i_{\text{chain}}, i_{\text{ind}} + 1, j\}$. Переходим к пункту 9.

9. Конец цикла. Переход к пункту 2.

10. Конец процедуры.

Трудоёмкость обоих рассматриваемых методов одинакова, так как для формирования множества W количество операций сравнения хешей определяется только числом элементов множества U, которые войдут в состав древовидной структуры (см. рис. 1).

В соответствии с описанным выше алгоритмом формирования древовидной структуры информационных блоков, ошибки передачи блоков возможны, когда как минимум два сформированных подмножества w_i и w_j, $i \neq j$, $|w_i| = |w_j|$, отличаются одним или несколькими блоками, расположенными в середине:

$$w_i \cap w_j = \left\{ w_i^{\langle k \rangle}, \ldots, w_j^{\langle k+L \rangle} \right\}, 1 < k < n, \quad (2)$$

где L – длина отличающегося фрагмента двух ветвей.

Предположим, подмножество w_i сформировано полностью из блоков целевого устройства. Тогда подмножество w_j будем называть посторонним относительно w_i. Пример подобных ветвей, отличающихся лишь фрагментами, показан на рис. 2, где серым тоном выделены участки из блоков посторонних источников.

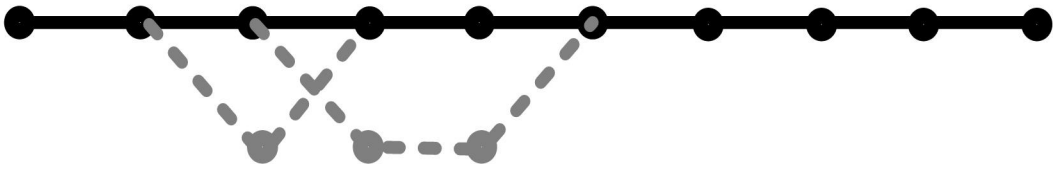

Рис. 2. Пример формирования посторонних ветвей

Fig. 2. Example of forming extraneous branches

Таким образом, если к произвольному этапу реализации рекурсивного алгоритма сформировано более чем одно подмножество длины n, и их количество не уменьшается (альтернативный алгоритм формирования графа «сверху вниз» может предусматривать отбрасывание ветвей, которые не дополняются блоками очередного подмножества u_i, $i = \overline{0\text{K}\ n-1}$), то ошибка уже возникла, и дальнейшая реализация алгоритма не имеет смысла. Это является теоретической предпосылкой к увеличению скорости обнаружения ошибочной передачи цепочки блоков данных. Кроме того, при формировании древовидной структуры (см. рис. 1) необходимо организовывать хранение сформированных цепочек блоков во внутренней памяти приёмника [15]. При использовании алгоритма формирования цепочек «сверху вниз» [16] необходимо хранить все цепочки до момента окончательного построения всего дерева, тогда как при рекурсивном алгоритме среднее число хранящихся цепочек будет меньше, так как после обнаружения ошибки новые цепочки формироваться не будут.

Для исследования эффекта, заключающегося в снижении среднего числа элементарных операций сравнения хешей и числа задействованных блоков памяти, оба алгоритма были реализованы на языке высокого уровня. Разрабо-

танное программное средство формировало статистику по исследуемым параметрам на основе 10^4 циклов реализации алгоритма над случайно сформированным в каждом цикле множеством информационных блоков U. Такое количество опытов позволяет судить по среднему значению наблюдаемых параметров об их математическом ожидании с точностью 10^{-3} [17, 18]. Изменяемым параметром при имитационном моделировании, помимо мощности множества U и числа блоков в сообщении целевого источника n, был размер H поля хеша, выделяемого операцией $f^{\text{hash}}(u)$.

Результаты и их обсуждение

Полученные в результате имитационного моделирования зависимости между средним числом операций сравнения хешей T, средним числом сформированных в приёмнике ветвей древовидной структуры N и числом анализируемых блоков $|U|$ для обоих рассматриваемых алгоритмов приведены на рис. 3 и 4. Графики построены для одного значения длины поля хеша H, но аналогичная картина наблюдается и при других значениях параметра H: при небольшом числе анализируемых блоков разница в вычислительных и ресурсных затратах для алгоритма незначительна, тогда как с ростом значения $|U|$ наблюдается всё больший выигрыш при использовании рекурсивного алгоритма. Более ярко он выражен в числе сформированных ветвей древовидной структуры. Это объясняется тем, что рекурсивный алгоритм прекращает свою работу при формировании двух ветвей длиной n, тогда как алгоритм формирования древовидной структуры «сверху вниз» продолжает свою работу

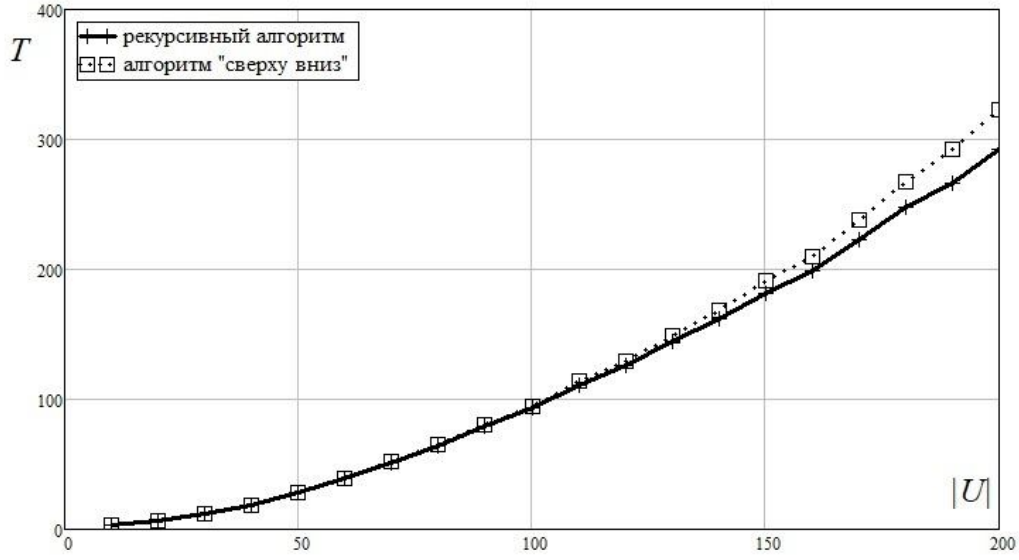

Рис. 3. Зависимость T среднего числа операций сравнения хешей от мощности множества анализируемых блоков U

Fig. 3. Dependence of the average number of hash comparison operations T on the power of the analyzed blocks set U

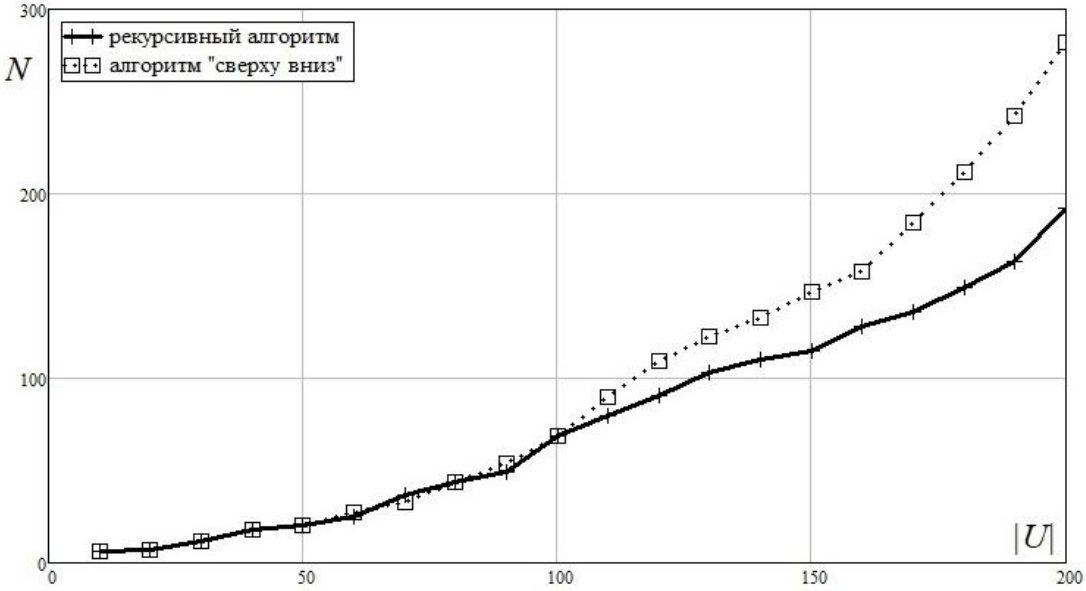

Рис. 4. Зависимость *N* среднего числа ветвей древовидной структуры от мощности множества анализируемых блоков *U*

Fig. 4. Dependence *N* of the average number of branches of the tree structure on the power of the analyzed blocks set *U*

Так как между достоверностью выполнения процедур идентификации источников и числом анализируемых блоков $|U|$ существует зависимость, то с точки зрения практического использования представляет интерес зависимость между затратами на реализацию алгоритмов и наблюдаемой частотой возникновения ошибок. Данная зависимость (в части затрат памяти на хранение множества ветвей древовидной структуры) приведена на рис. 5.

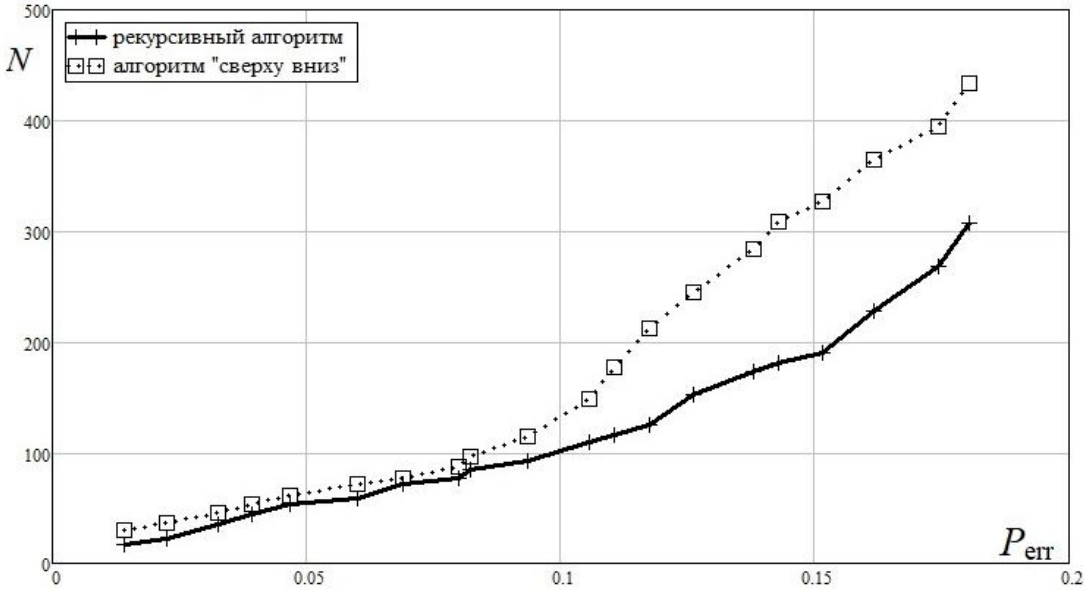

Рис. 5. Зависимость *N* среднего числа ветвей древовидной структуры от вероятности ошибки определения источника данных P_{err}

Fig. 5. Dependence *N* of the average number of branches of the tree structure on the probability of the data source identification error P_{err}

Видно, что эффект от применения рекурсивного алгоритма начинает проявляться с вероятности ошибки $P_{err} > 0,07$. В диапазоне $P_{err} > 0,15$ число формируемых посторонних ветвей в рекурсивном алгоритме не превышает 70% от аналогичного числа ветвей при использовании стандартного итерационного алгоритма.

Выводы

Результаты проведённых исследований позволяют утверждать, что использование описанного рекурсивного алгоритма формирования древовидных структур данных в системах определения источников информации для групп блоков целесообразно при большом числе анализируемых блоков и при больших значениях наблюдаемых ошибок идентификации (превышающих 0,07). При этом среднее число выполняемых типовых операций снижается на 5 – 10 % по сравнению с обычным итерационным алгоритмом формирования древовидных структур, а число ветвей в таких структурах снижается на 20–30%. Для меньшей интенсивности возникновения ошибок его использование нецелесообразно, так как реализация рекурсии по сравнению с реализацией обычного итерационного алгоритма потребует усложнения структуры микропрограммного устройства управления в приёмнике. Это усложнение будет выражено в необходимости определения дополнительных команд микропрограммы, обеспечивающих вызов подпрограмм [19], а также организацию дополнительной стековой памяти в устройстве управления [20]. При этом размер такого стека должен быть достаточен для рекурсии максимальной глубиной n – длине множества блоков данных, для которых производится определение источника. Поэтому, если приёмник проектируется для работы на протоколах, предусматривающих высокую достоверность определения источника (вероятность ошибки передачи меньше 0,05) [21], то в нём целесообразно использовать обычный итерационный алгоритм формирования древовидных структур. Сопоставление характеристик итерационного и рекурсивного алгоритма при различных значениях вероятности ошибки идентификации приведено в табл. 1. За единицу принято число операций сравнения хеша и количество блоков памяти, требуемых для хранения древовидной структуры при использовании итерационного алгоритма.

Кроме этого, полученные результаты подтверждает тезис о зависимости между сложностью древовидной структуры, формируемой в приемнике при проведении процедуры идентификации источника, и вероятностью возникновения ошибок. Это является экспериментальным обоснованием возможности синтеза алгоритмов обнаружения ошибок на основе анализа лишь числа ветвей древовидной структуры до момента построения ветвей длиной n.

Таблица 1. Сравнение алгоритмов формирования древовидной структуры

Table 1. Comparison of algorithms for forming a tree structure

Тип алгоритма формирования древовидной структуры блоков данных / Type of algorithm for forming a tree structure of data blocks	Вероятность возникновения ошибки определения источника данных P_{err} / The probability of the data source determining error P_{err}				Дополнительные требования к организации преёмника данных / Additional requirements for the data receiver organization
	$P_{err} \leq 0{,}07$		$P_{err} > 0{,}07$		
	Число сравнений хешей / Number of hash comparisons	Число блоков памяти для хранения древовидной структуры / The number of memory blocks for storing the tree structure	Число сравнений хешей / Number of hash comparisons	Число блоков памяти для хранения древовидной структуры / The number of memory blocks for storing the tree structure	
Итерационный	T	N	T	N	отсутствуют
Рекурсивный	$\approx T$	$<0.9 \cdot N$	от $0.9 \cdot T$ до $0.95 \cdot T$	$<0.8 \cdot N$	требуется организация вызова подпрограмм в микропрограммном устройстве управления

Реализация таких алгоритмов позволит дополнительно повысить скорость обнаружения ошибок, снизив число выполняемых приёмником типовых операций сравнения хешей.

Список литературы

1. Мальчуков А.Н., Осокин А.Н. Система автоматизированного проектирования кодеков помехоустойчивых кодов короткой длины // Известия Томского политехнического университета. 2008. Т. 312. № 5. С. 70-75.

2. Мыцко Е.А., Мальчуков А.Н., Иванов С.Д. Исследование алгоритмов вычисления контрольной суммы CRC8 в микропроцессорных системах при дефиците ресурсов // Приборы и системы. Управление, контроль, диагностика. 2018. № 6. С. 22-29.

3. Xie J., Pan X. An improved rc4 stream cipher // International Conference on Computer Application and System Modeling.2010. https://doi.org: 10.1109/IC-CASM.2010.5620800

4. Zhang W., Gong X., Han G. An improved ant colony algorithm for path planning in one scenic area with many spots // Mathematical Problems in Engineering. 2016. https://doi.org:10.1155/2016/7672839

5. Allouch A., Cheikhrouhou O., Koubaa A. MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems // International Wireless Communications and Mobile Computing Conference (IWCMC). Morrocco. 2019. https://doi.org: 10.1109/IWCMC.2019.8766667

6. Iwata T., Kurosawa K. OMAC: one-key CBC MAC // Fast Software Encryption. 2003. P. 129 – 53.

7. Liu C., Ji J., Liu Z. Implementation of DES Encryption Arithmetic based on FPGA // AASRI Procedia. 2013. Vol. 5. P. 209–213.

8. Black J., Rogaway P. CBC MACs for arbitrary-length messages: The three-key constructions // J. Crypto. 2015. Vol. 18. №2. P. 111–131.

9. Stallings W. NIST Block Cipher Modes of Operation for Confidentiality // Cryptologia. 2010. № 34(2). P. 163 – 175.

10. Ben Othman S., Alzaid H., Trad A., & Youssef, H. An efficient secure data aggregation scheme for wireless sensor networks. IISA 2013. https://doi.org: 10.1109/iisa.2013.6623701

11. Fangfang Dai, Yue Shi, Nan Meng, Liang Wei and Zhiguo Ye From Bitcoin to Cybersecurity: a Comparative Study of Blockchain Application and Security Issues // The 2017 4th International Conference on Systems and Informatics (ICSAI 2017). Hangzhou, China 2017.

12. Левитин А. В. Алгоритмы. Введение в разработку и анализ. М.: Вильямс, 2006. 576 с.

13. Алгоритмы. Построение и анализ / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. М.: Вильямс, 2019. 1328 с.

14. Таныгин М.О., Алшаиа Х.Я., Алтухова В.А. Об одном методе контроля целостности передаваемой поблоково информации // Телекоммуникации. 2019. № 3. С. 12-21.

15. Таныгин М.О., Алшаиа Х.А., Добрица В.П. Оценка влияния организации буферной памяти на скорость выполнения процедур определения источника сообщений // Труды МАИ. 2020. № 5(114). С.15.

16. Таныгин М.О. Теоретические основы идентификации источников информации, передаваемой блоками ограниченного размера. Курск, 2020. 198 с.

17. Кельтон В., Лоу А. Имитационное моделирование. СПб.: Питер, 2004. 874 с.

18. Строгалев В.П., Толкачева И.О. Имитационное моделирование. М.: МГТУ им. Баумана, 2008. 279 с.

19. Хоровиц П. Искусство схемотехники. Т. 1. М.: Медиа, 2012. 600 с.

20. Таненбаум Э., Остин Т. Архитектура компьютера. СПб. : Питер, 2012. 816 с.

21. Предварительный национальный стандарт РФ. Информационные технологии. Интернет вещей. Протокол обмена для высокоемких сетей с большим радиусом действия и низким энергопотреблением. URL: http://docs.cntd.ru/document/554596382 (дата обращения 15.03.2021).

References

1. Malchukov A.N., Osokin A.N. Sistema avtomatizirovannogo proektirovaniya kodekov pomekhoustojchivyh kodov korotkoj dliny [Computer-aided design system for short-length noise-resistant codecs]. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2008, vol. 312, no. 5, pp. 70-75 (In Russ.).

2. Mycko E.A., Mal'chukov A.N., Ivanov S.D. Issledovanie algoritmov vychisleniya kontrol'noj summy CRC8 v mikroprocessornyh sistemah pri deficite resursov [Investigation of algorithms for calculating the CRC8 checksum in microprocessor systems with a shortage of resources]. *Pribory i sistemy. Upravlenie, kontrol', diagnostika = Instrements and Systems: Monitoring, Control and Diagnostics*, 2018, no. 6, pp. 22-29 (In Russ.).

3. Xie J., Pan X. An improved rc4 stream cipher. *International Conference on Computer Application and System Modeling*. 2010. https://doi.org: 10.1109/IC-CASM.2010.5620800

4. Zhang W., Gong X., Han G. An improved ant colony algorithm for path planning in one scenic area with many spots. *Mathematical Problems in Engineering*, 2016. https://doi.org: 10.1155/2016/7672839

5. Allouch A., Cheikhrouhou O., Koubaa A. MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems. *International Wireless Communications and Mobile Computing Conference (IWCMC)*. Morrocco, 2019. https://doi.org:10.1109/IWCMC.2019.8766667

6. Iwata T., Kurosawa K. OMAC: one-key CBC MAC. *Fast Software Encryption*. 2003, pp. 129 – 53.

7. Liu C., Ji J., Liu Z. Implementation of DES Encryption Arithmetic based on FPGA. *AASRI Procedia*, 2013, vol. 5, pp. 209–213.

8. Black J., Rogaway P. CBC MACs for arbitrary-length messages: The three-key constructions. *J. Cryptol*, 2015, vol. 18, no.2, pp. 111–131.

9. Stallings W. NIST Block Cipher Modes of Operation for Confidentiality. *Cryptologia*, 2010, no. 34(2), pp. 163 – 175.

10. Ben Othman S., Alzaid H., Trad A., Youssef H. An efficient secure data aggregation scheme for wireless sensor networks. IISA, 2013. https://doi.org: 10.1109/ iisa.2013.6623701

11. Fangfang Dai, Yue Shi, Nan Meng, Liang Wei and Zhiguo Ye From Bitcoin to Cybersecurity: a Comparative Study of Blockchain Application and Security Issues. *The 2017 4th International Conference on Systems and Informatics (ICSAI 2017).* Hangzhou, China 2017.

12. Levitin A. V. *Algoritmy. Vvedenie v razrabotku i analiz* [Introduction to development and analysis]. Moscow, Vil'yams Publ., 2006. 576 p. (In Russ.).

13. Kormen T. H., Lejzerson CH. I., Rivest R. L., SHtajn K. *Algoritmy. Postroenie i analiz* [Algorithms. Construction and analysis]. Moscow, Vil'yams Publ., 2019. 1328 p. (In Russ.).

14. Tanygin M.O., Alshaia H.YA., Altuhova V.A. Ob odnom metode kontrolya celostnosti peredavae-moj poblokovo informacii [On one method of controlling the integrity of the information transmitted by the block]. *Telekommunikacii = Telecommunications*, 2019, no. 3, pp. 12-21 (In Russ.).

15. Tanygin M.O., Alshaia H.A., Dobrica V.P. Ocenka vliyaniya organizacii bufernoj pamyati na skorost' vypolneniya procedur opredeleniya istochnika soobshchenij [Assessment of the impact of the organization of the buffer memory to the speed of the procedures for determining the source of the message]. Trudy MAI = Proceedings of MAI, 2020, no. 5(114), p.15 (In Russ.).

16. Tanygin M.O. *Teoreticheskie osnovy identifikacii istochnikov informacii, peredavaemoj blokami ogranichennogo razmera* [Theoretical bases of identification of sources of information transmitted blocks of limited size]. Kursk, 2020. 198 p. (In Russ.).

17. Kelton V., Lou A. *Imitacionnoe modelirovanie* [Simulation modeling]. Saint Petersburg, Piter Publ., 2004. 874 p. (In Russ.).

18. Strogalev V.P., Tolkacheva I.O. *Imitacionnoe modelirovanie* [Simulation modeling]. Moscow, MGTU im. Baumana Publ., 2008. 279 p. (In Russ.).

19. Horovitz P. *Iskusstvo skhemotekhniki* [The art of circuit engineering]. Moscow, 2012. 600 p. (In Russ.).

20. Tanenbaum E., Ostin T. *Arhitektura komp'yutera* [Computer Architecture]. Saint Petersburg, Piter Publ., 2012. 816 p. (In Russ.).

21. Predvaritel'nyj nacionalnyj standart RF. Informacionnye tekhnologii. Internet veshchej. Protokol obmena dlya vysokoemkih setej s bolshim radiusom dejstviya i nizkim energopotrebleniem [Preliminary National Standard of the Russian Federation. Information technologies. The Internet of Things. Exchange protocol for high-capacity networks with a long range and low power consumption]. Available at: http://docs.cntd.ru/document/554596382 (accessed 15.03.2021) (In Russ.).

Информация об авторах / Information about the Authors

Таныгин Максим Олегович, кандидат технических наук, доцент, заведующий кафедрой «Информационная безопасность», Юго-Западный государственный университет, г. Курск, Российская Федерация.
e-mail: tanygin@yandex.ru

Maxim O. Tanygin, Cand. of Sci. (Engineering), Associate Professor Head of Information Security Department, Southwest State University, Kursk, Russian Federation,
e-mail: tanygin@yandex.ru

Хайдер Яхья Алшаиа, аспирант кафедры «Информационная безопасность», Юго-Западный государственный университет, г. Курск, Российская Федерация.
e-mail: haideryhy7@gmail.com

Hayder Yahja Alshaeaa, Post-Graduate of Information Security Department, Southwest State University, Kursk, Russian Federation,
e-mail: haideryhy7@gmail.com

Добрица Вячеслав Порфирьевич, доктор физико-математических наук, профессор, профессор кафедры «Информационная безопасность», Юго-Западный государственный университет, г. Курск, Российская Федерация,
e-mail: dobritsa@mail.ru

Vjacheslav P. Dobritsa, Dr. of Sci. (Engineering), Professor, Information Security Department, Southwest State University, Kursk, Russian Federation,
e-mail: dobritsa@mail.ru

Добросердов Олег Гурьевич, доктор технических наук, старший научный сотрудник, Юго-Западный государственный университет, г. Курск, Российская Федерация,
e-mail: serfingk@yandex.ru

Oleg G. Dobroserdov, Dr. of Sci. (Engineering), Senior Research Associate, Southwest State University, Kursk, Russian Federation,
e-mail: serfingk@yandex.ru