УДК 004.056

05.00.00 Технические науки

ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ ДАННЫХ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ НА ОСНОВЕ ЛИНЕЙНЫХ СИСТЕМ ХЭШ-КОДОВ

Савин Сергей Владимирович
SPIN-код = 9468-6007
Краснодарское высшее военное училище имени генерала армии С.М. Штеменко, Краснодар, Россия

Финько Олег Анатольевич
доктор технических наук, профессор, советник РАРАН
SPIN-код = 8202-6046
Краснодарское высшее военное училище имени генерала армии С.М. Штеменко, Краснодар, Россия; Российская академия ракетных и артиллерийских наук, Россия

Решается задача уменьшения избыточности контрольной информации (сигнатур хэш-функций, электронных подписей) для обеспечения целостности записей данных. Накладываются ограничения на максимально допустимое количество записей с нарушением целостности. В известных решениях данной задачи с увеличением уровня защищенности данных увеличивается и количество контрольной информации (коэффициент избыточности). Введено понятие линейных систем хэш-кодов (ЛСХК). С помощью математического аппарата теории систем векторов выполнено обоснование и разработка алгоритма построения ЛСХК для обеспечения целостности данных в автоматизированных системах, который позволяет для заданного уровня защищенности данных (обеспечение целостности) уменьшить избыточность контрольной информации. Правила (принципы) построения ЛСХК аналогичны правилам построения линейных избыточных кодов, в частности кодов Хемминга. Предложен алгоритм контроля целостности данных в ЛСХК. Применение разработанных алгоритмов позволяет обеспечить необходимый уровень защищенности (целостность) данных в широком диапазоне требований технического задания заказчика

Ключевые слова: ЗАЩИТА ИНФОРМАЦИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ, ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ ДАННЫХ, ХЭШ-ФУНКЦИЯ, ХЭШ-КОД, ЭЛЕКТРОННАЯ ПОДПИСЬ

UDC 004.056

Technical sciences

DATA INTEGRITY IN THE AUTOMATED SYSTEM BASED ON LINEAR SYSTEMS HASHES

Savin Sergey Vladimirovich
RSCI SPIN-code = 9468-6007
Krasnodar Military School named S.M. Shtemenko, Krasnodar, Russia

Finko Oleg Anatolievich
Dr.Sci.Tech., Professor

RSCI SPIN-code = 8202-6046
Krasnodar Military School named S.M. Shtemenko, Krasnodar, Russia
Russian Academy of Missile and Artillery Sciences, Russia

To protect your data (data integrity) in the automated systems, we provide a solution of the problem, which is to reduce redundancy control of information (hash codes, electronic signatures). We impose restrictions on the maximum number of violations of the integrity of the records in the data block. It is known, that with an increase in data protection the amount of control information (coefficient of redundancy) also increases. We introduce the concept of linear systems of hash codes (LSHC). On the basis of the mathematical apparatus of the theory of systems of vectors we have developed an algorithm for constructing LSHC, which allows (for a given level of data protection, i.e. integrity) to reduce the redundancy of the control information. Rules (principles) of building LSHC comply with the rules of construction in coding theory (Hamming codes). The article provides an algorithm for data integrity in LSHC. The use of algorithms ensures the necessary level of data protection and the requirements specification of customers

Keywords: INFORMATION SECURITY IN AUTOMATED SYSTEM, DATA INTEGRITY, HASH FUNCTION, HASH CODE, ELECTRONIC SIGNATURE

1. Введение. Современный этап развития общества характеризуется возрастающей ролью информационной сферы, представляющей собой совокупность информации, информационной инфраструктуры, субъектов, осуществляющих сбор, формирование, распространение и использование информации, а также системы регулирования возникающих при этом общественных отношений. Национальные интересы Российской Федерации в информационной сфере включают в себя развитие современных информационных технологий, отечественной индустрии информации, в том числе индустрии средств информатизации, телекоммуникации и связи, а также защиту информационных ресурсов от несанкционированного доступа, обеспечение безопасности информационных и телекоммуникационных систем, как уже развернутых, так и создаваемых на территории России [1].

К развернутым и создаваемым автоматизированным системам (АС) предъявляются требования по защите (обеспечению конфиденциальности, доступности и целостности) информации (данных) [2, 3]. Выполненный анализ соответствующих руководящих документов и литературных источников выявил необходимость контроля целостности данных в условиях реализации, в том числе внутренних, угроз безопасности информации [4]. Так, например, более 90% компаний сталкивались с внутренними вторжениями, более половины сталкиваются с ними постоянно. Большая часть потерь связана с действиями сотрудников самих этих компаний, так как существующие методы обеспечения целостности данных контролируются самими держателями баз данных. Причинами нарушений целостности данных могут быть (внутренние угрозы): месть, корысть, страх, принуждение, вандализм, любопытство, тщеславие, самоутверждение, карьерные идеи, конкуренция и другие [5]. Это подтверждается отчетом компании InfoWatch за 2014 год [6]:

- нарушения внутренними пользователями – 54 %;

- внешний злоумышленник – 25,8 %;
- не определено – 12,5%;
- подрядчик – 4,2 %;
- руководитель – 1,5 %;
- системный администратор – 1,2 %;
- бывший сотрудник – 0,9 %.

Из перечня известных угроз безопасности данных [7] важными являются угрозы, основанные на злоупотреблении уполномоченными пользователями своими правами, которые приводят к уничтожению (модификации) отдельных областей хранения данных, относящихся к действиям администратора. Таким образом, злоумышленник, который проник в систему и получил привилегированные полномочия, может скрыть факт атаки [8].

В [9] для решения данной задачи было предложено использование так называемого метода «однократной записи» [10], сущность которого заключается в применении различных способов изготовления, изменения, копирования и размножения документов, которые позволяют обнаружить любое изменение в документе (запись не может быть заменена, вместо этого, в документе добавляется новая запись). Однако, с увеличением уровня защищенности данных (от внутреннего нарушителя) увеличивается и количество избыточной информации, необходимой для решения задачи защиты данных, что при заданных ограничениях в техническом задании заказчика не позволяет решить задачу защиты данных в полном объеме.

Цель статьи – обеспечение необходимого уровня защищенности (целостности) данных в АС на основе разработки алгоритма построения линейных систем хэш-кодов.

2. Известные решения задачи обеспечения целостности данных.

Наиболее распространенными методами решения задачи обеспечения целостности данных в АС являются:

- применение различных видов резервирования (RAID-массивы, методы дублирования, методы избыточного кодирование) [11 – 13];
- применение криптографических методов: ключевое и бесключевое хеширование, средства электронной подписи (ЭП) [14 – 18].

Использование систем обеспечения целостности данных на основе использования ключевых хэш-функций имеет ряд преимуществ:

- относительно невысокая избыточность;
- уменьшение количества криптографических преобразований;
- возможность регулировать длину хэш-кода.

Наиболее типичны следующие две схемы получения хэш-кодов (рис. 1, 2).

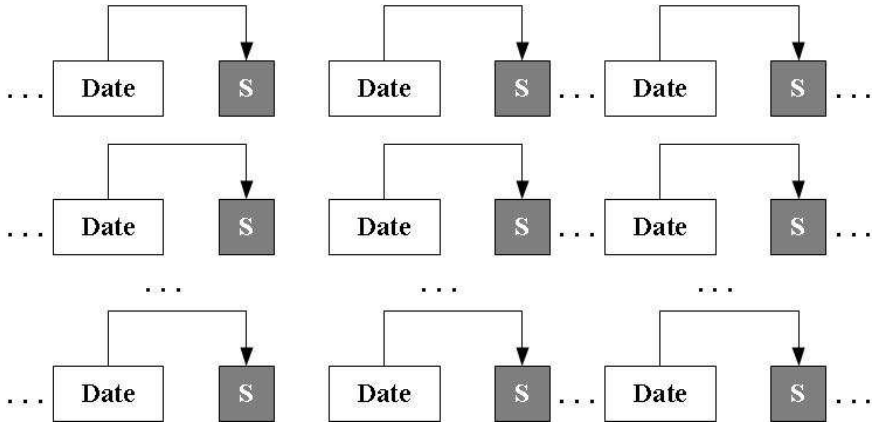

Рис. 1 – Схема получения хэш-кода для каждой записи в блоке данных

Недостаток данных схем – высокая избыточность при контроле целостности последовательностей записей небольшой размерности.

Пусть:

$M = \{\vec{m}_{t_i}, \vec{m}_{t_{i+1}}, \vec{m}_{t_{i+2}}, \ldots, \vec{m}_{t_{i+k}}\}$ – множество двоичных векторов произвольной конечной длины (блок данных);

$S = \{\vec{s}_{t_i}, \vec{s}_{t_{i+1}}, \vec{s}_{t_{i+2}}, \ldots, \vec{s}_{t_{i+k}}\}$ – множество двоичных векторов фиксированной конечной длины (сигнатуры хэш-функций);

$\vec{s}_{t_i} = h(\vec{m}_{t_i})$ – вычисление сигнатуры хэш-функции;

«||» – операция конкатенации двоичных векторов.

Контроль целостности данных, представленных двоичными векторами $\vec{m}_{t_i}, \vec{m}_{t_{i+1}}, \vec{m}_{t_{i+2}}, \ldots, \vec{m}_{t_{i+k}}$, состоит из двух этапов:

1) вычисление сигнатур хэш-функции для проверяемых записей $\vec{m}^*_{t_i}, \vec{m}^*_{t_{i+1}}, \vec{m}^*_{t_{i+2}}, \ldots, \vec{m}^*_{t_{i+k}}$:

$$\begin{cases} \vec{s}^*_{t_{i+1}} = h(\vec{m}^*_{t_{i+1}}), \\ \vec{s}^*_{t_{i+2}} = h(\vec{m}^*_{t_{i+2}}), \\ \ldots\ldots\ldots\ldots, \\ \vec{s}^*_{t_{i+k}} = h(\vec{m}^*_{t_{i+k}}); \end{cases}$$

2) проверка достоверности сигнатур хэш-функции под каждой записью, которая соответствует предикату:

$$P\left(\vec{m}_{t_{i+k}}\right) = \begin{cases} 1, & \text{if} \quad \vec{s}^*_{t_{i+k}} = \vec{s}_{t_{i+k}}; \\ 0, & \text{if} \quad \vec{s}^*_{t_{i+k}} \neq \vec{s}_{t_{i+k}}, \end{cases}$$

где «1» – означает, что искажений нет, «0» – есть.

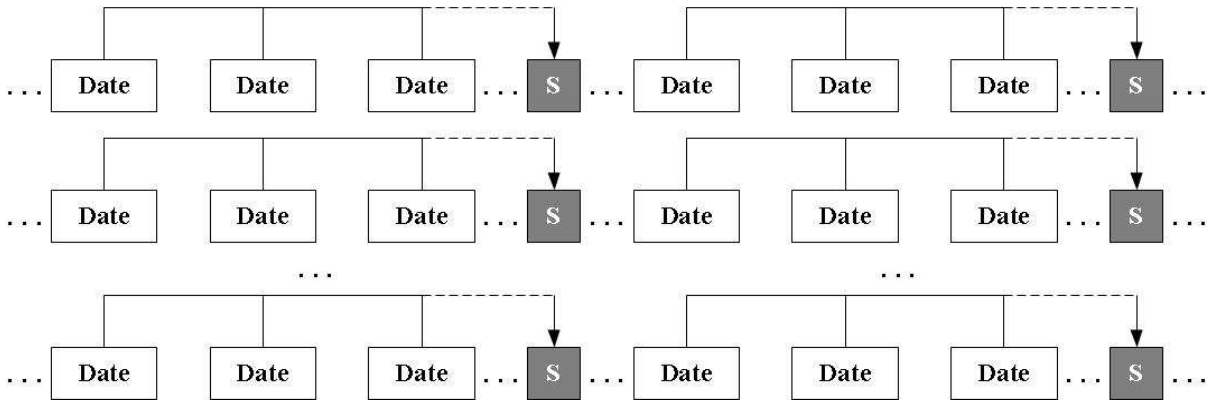

Рис. 2 – Схема получения хэш-кода для блока данных

Недостаток – отсутствие возможности локализации искаженных записей в каждом блоке данных.

3. Математическое обоснование разработки линейной системы хэш-кодов для обеспечения целостности данных в АС.

Введем параметры:

1) объем i-го фрагмента данных: V_i;

2) коэффициент избыточности: $K_{\text{изб}} = \frac{\sum_{i=1}^{d} V_i}{\sum_{i=1}^{d} V_i + Gr}$,

где d – общее количество фрагментов данных, G – объем данных, занимаемый одной сигнатурой хэш-функции, r – общее количество сигнатур хэш-функций;

3) уровень защищенности i-го фрагмента данных: $z_i = 2^{V_i}\left(2^r - 1\right)$;

4) средний уровень защищенности данных: $Z = \frac{\sum_{i=1}^{d} z_i}{d}$.

Пусть c_i – количество фрагментов данных, имеющих нарушение целостности, $P(c_i)$ – вероятность нарушения целостности i-го фрагмента данных.

Введем допущения:

1) $P(c_i) > P(c_j)$, если V_i записи c_i меньше V_j записи c_j;

2) для контроля целостности любых данных всегда используется однотипная хэш-функция;

3) объемы данных, заключенных в одной записи (рис. 1) и в произвольном блоке данных (рис. 2) равны, следовательно, и уровни защищенности одной записи данных и блока данных - одинаковы.

Таким образом, при данных допущениях и различных сочетаниях способов получения хэш-кодов (рис. 1, 2) уровень защищенности и коэффициент избыточности не меняется: $Z_{\forall i} = Z_{\forall j}$, $K_{\forall i} = K_{\forall j}$ (рис. 3).

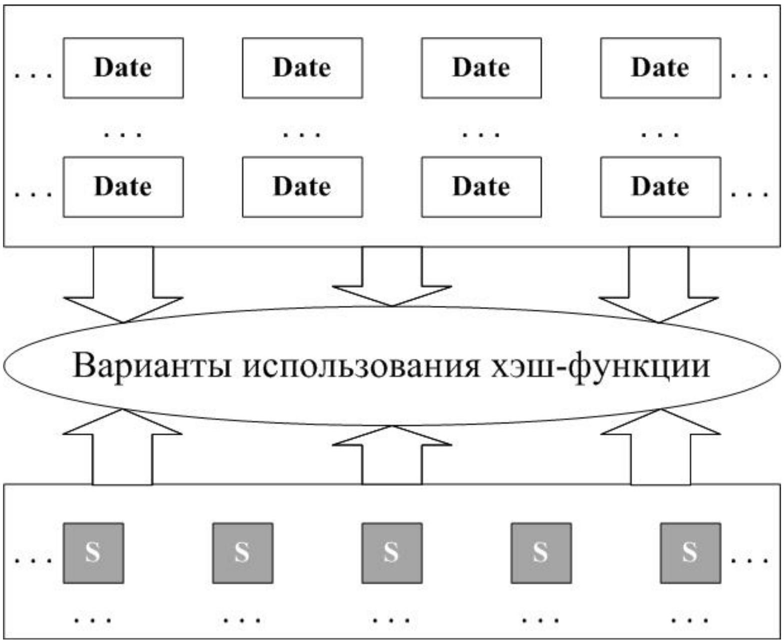

Рис. 3 – Обобщенная схема применения хэш-функций

Необходимо, для заданного уровня защищенности $Z_{\forall i} = \text{const}$ обеспечить уменьшение коэффициента избыточности $\downarrow K_{\text{изб}}$. Следовательно, для заданных значений n и k выполнить условие $n<k$:

$$\begin{cases} h(\vec{m}_{t_i}^{(1)} \| \vec{m}_{t_{i+1}}^{(1)} \| \vec{m}_{t_{i+2}}^{(1)} \| \ldots \| \vec{m}_{t_{i+d}}^{(1)}) = \vec{s}_{t_i}^{(1)}; \\ h(\vec{m}_{t_i}^{(2)} \| \vec{m}_{t_{i+1}}^{(2)} \| \vec{m}_{t_{i+2}}^{(2)} \| \ldots \| \vec{m}_{t_{i+x}}^{(2)}) = \vec{s}_{t_{i+1}}^{(2)}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{ю}\ldots\ldots; \\ h(\vec{m}_{t_i}^{(n)} \| \vec{m}_{t_{i+1}}^{(n)} \| \vec{m}_{t_{i+2}}^{(n)} \| \ldots \| \vec{m}_{t_{i+k}}^{(n)}) = \vec{s}_{t_{i+n}}^{(n)}. \end{cases}$$

Для решения данной задачи предлагается использовать математический аппарат теории систем векторов и линейных векторных пространств.

Множество двоичных векторов $\{\vec{m}_{t_i}, \vec{m}_{t_{i+1}}, \vec{m}_{t_{i+2}}, \ldots, \vec{m}_{t_{i+k}}\}$ можно рассматривать как систему линейно независимых векторов, т.к.:

$$x_1\vec{m}_{t_i} + x_2\vec{m}_{t_{i+1}} + x_3\vec{m}_{t_{i+2}} + \ldots + x_c\vec{m}_{t_{i+k}} = 0,$$

только при нулевом наборе коэффициентов: $x_1, x_2, x_3, \ldots, x_c$, где $x_c \in (0,1)$.

Данная система образует базис:

$$\vec{m}_{t_i} = E_1 = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \quad \vec{m}_{t_{i+1}} = E_2 = \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \end{pmatrix}, \quad \dots, \quad \vec{m}_{t_{i+k}} = E_k = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \end{pmatrix}.$$

Тогда множество записей и хэш-кодов: $\{\vec{m}_{t_i}, \vec{m}_{t_{i+1}}, \vec{m}_{t_{i+2}}, \dots, \vec{m}_{t_{i+k}}, \vec{s}_{t_i}, \vec{s}_{t_{i+1}}, \vec{s}_{t_{i+2}}, \dots, \vec{s}_{t_{i+n}}\}$ – также можно рассматривать как систему линейно независимых векторов, где базисом для них будет:

$$E_1, \quad E_2, \quad \dots, \quad E_k.$$

Множество всех возможных схем хеширования записей $\{\vec{m}_{t_i}, \vec{m}_{t_{i+1}}, \vec{m}_{t_{i+2}}, \dots, \vec{m}_{t_{i+k}}\}$ можно представить в виде двоичной матрицы, составленной из коэффициентов базиса:

$$n \cdot \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{pmatrix},$$

где каждая строка соответствует схеме хеширования и $n \le k$.

Строки матрицы обладают свойствами:

- являются различными и линейно-независимыми векторами;
- расстояние между векторами (по Хэммингу) $d_{\min} \ge 2$;
- каждый вектор имеет вес (в смысле Хэмминга) $\omega \ge d_{\min}$;
- нулевой вектор не входит в матрицу.

Аналогичными свойствами обладает порождающая матрица в теории линейных кодов [13], что позволяет сделать вывод о возможности использования правил (принципов) построения линейных кодов для построения линейных систем хэш-кодов.

Определение. Система хэш-кодов – множество хэш-кодов, полученных с помощью стандартной процедуры реализации хэш-функции от совокупностей данных (записей) в порядке, определенных специальной

процедурой выбора записей, основанной на математическом аппарате линейной алгебры.

Такие системы хэш-кодов в рамках настоящей работы будем назвать линейными (ЛСХК).

4. Алгоритм построения ЛСХК для обеспечения целостности данных в АС.

Хеширование исходного блока данных можно представить в виде следующего выражения:

$$(\vec{m}_{t_i}\ \vec{m}_{t_{i+1}}\ \vec{m}_{t_{i+2}}\ldots\vec{m}_{t_{i+k}})\rightarrow\ \ (\vec{m}_{t_i}\ \vec{m}_{t_{i+1}}\ \vec{m}_{t_{i+2}}\ldots\vec{m}_{t_{i+k}}\ \vec{s}_{t_{i+k+1}}\ \vec{s}_{t_{i+k+2}}\ldots\vec{s}_{t_{i+r}}),$$

где символ «$\rightarrow$» – означает отображение – специальная многомерная некоммутативная операция хеширования.

Тогда, получение защищенного блока данных с помощью хеширования можно представить в виде следующего выражения:

$$\left(\vec{m}_{t_i}\ \vec{m}_{t_{i+1}}\ldots\vec{m}_{t_{i+k}}\right)\otimes\begin{pmatrix}1\ 0\ldots0 & a_{11}a_{12}\ldots a_{1,r-k}\\ 0\ 1\ldots0 & a_{21}a_{22}\ldots a_{2,r-k}\\ \ldots\ldots\ldots & \ldots\ \ldots\ldots\ \ \ldots\\ 0\ 0\ldots1 & a_{k1}a_{k2}\ldots a_{k,r-k}\end{pmatrix}=\left(\vec{m}_{t_i}\ \vec{m}_{t_{i+1}}\ldots\vec{m}_{t_{i+k}}\ \vec{s}_{t_{i+k+1}}\ldots\vec{s}_{t_{i+r}}\right),$$

где $\vec{s}_{t_{i+r}}=h(a_0\vec{m}_{t_i}\|a_1\vec{m}_{t_{i+1}}\|\ldots\|a_k\vec{m}_{t_{i+k}}),\ \ a_k\in\{1,0\}$

или: $\overrightarrow{M}_v\otimes\mathrm{G}=\left(\vec{m}_{t_i}\ \vec{m}_{t_{i+1}}\ldots\vec{m}_{t_{i+k}}\ \vec{s}_{t_{i+k+1}}\ldots\vec{s}_{t_{i+r}}\right)$, где символ «$\otimes$» – означает специальную многомерную некоммутативную операцию хеширования записей $\vec{m}_{t_i}$, отмеченных единичным символом $a_{k,r-k}=1$ матрицы G; если же $a_k=0$, то $a_k\vec{m}_{t_{i+k}}=\varnothing$; $\overrightarrow{M}_v=\left(\vec{m}_{t_i}\quad\vec{m}_{t_{i+1}}\quad\ldots\quad\vec{m}_{t_{i+k}}\right)$ – информационный вектор (блок данных).

Блок-схема алгоритма построения ЛСХК представлена на рисунке 4.

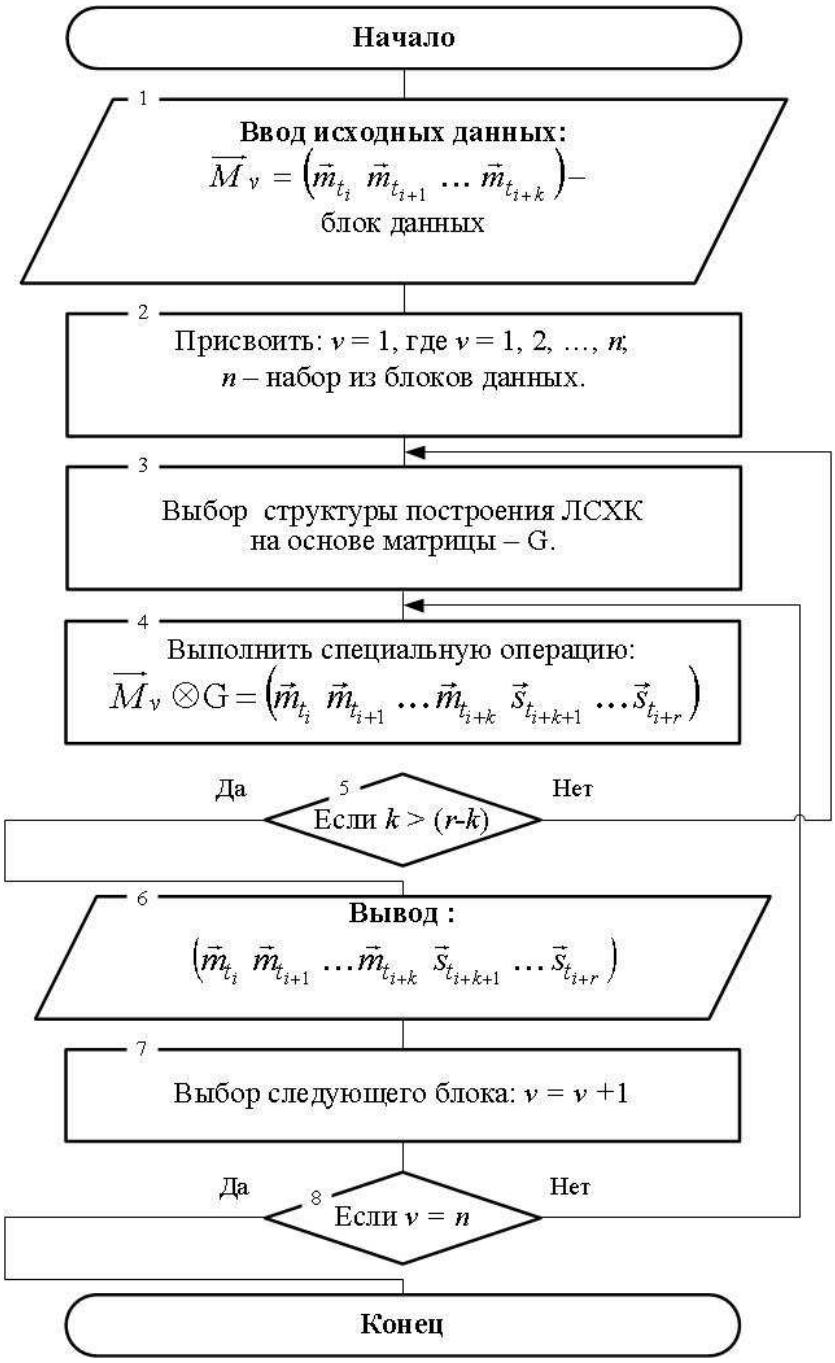

Рис. 4 – Алгоритм построения ЛСХК для обеспечения целостности данных

Для контроля целостности данных (обнаружение ошибки) в теории линейных кодов использует понятие синдром. Синдром $\vec{S}$ – это матрица-строка $(s_1\ \ s_2\ \ \dots\ \ s_L)$ с L элементами $s_L \in \{0,1\}$, по одному для каждого проверочного символа.

Под ошибкой в защищенном блоке $(\vec{m}_{t_i}\ \vec{m}_{t_{i+1}} \ldots \vec{m}_{t_{i+k}}\ \vec{s}_{t_{i+k+1}} \ldots \vec{s}_{t_{i+r}})$ будем понимать результат несоответствия двоичного вектора с результатом, полученным в результате проверки синдрома.

Алгоритм контроля целостности данных в ЛСХК.

Ввод: $(\vec{m}^*_{t_i}\ \vec{m}^*_{t_{i+1}} \ldots \vec{m}^*_{t_{i+k}}\ \vec{s}^*_{t_{i+k+1}} \ldots \vec{s}^*_{t_{i+r}})$ – контролируемый блок данных;

Шаг 1*:* Выполнить многомерную некоммутативную операцию хеширования записей $\vec{m}^*_{t_i}$, отмеченных единичным символом матрицы G:

$$\left(\vec{m}^*_{t_i}\ \vec{m}^*_{t_{i+1}} \ldots \vec{m}^*_{t_{i+k}}\right) \otimes \mathrm{G} = \left(\vec{m}^{**}_{t_i}\ \vec{m}^{**}_{t_{i+1}} \ldots \vec{m}^{**}_{t_{i+k}}\ \vec{s}^{**}_{t_{i+k+1}} \ldots \vec{s}^{**}_{t_{i+r}}\right).$$

Шаг 2: Вычислить синдром $\vec{\mathrm{S}} = (s_1\ s_2 \ldots s_{r-k})$, который соответствует значению предиката:

$$P\left(\vec{\mathrm{S}}\right) = \begin{cases} 0, & \text{if} \quad \vec{s}^*_{t_{i+r}} = \vec{s}^{**}_{t_{i+r}}; \\ 1, & \text{if} \quad \vec{s}^*_{t_{i+r}} \neq \vec{s}^{**}_{t_{i+r}}. \end{cases}.$$

Шаг 3: По таблице значений синдромов $\vec{\mathrm{S}}$ локализовать ошибку в блоке данных $\vec{m}_{t_i}\ \vec{m}_{t_{i+1}} \ldots \vec{m}_{t_{i+k}}\ \vec{s}_{t_{i+k+1}} \ldots \vec{s}_{t_{i+r}}$.

Вывод: данные о нарушении целостности в блоке данных.

Блок-схема алгоритма контроля целостности данных в ЛСХК представлена на рисунке 5.

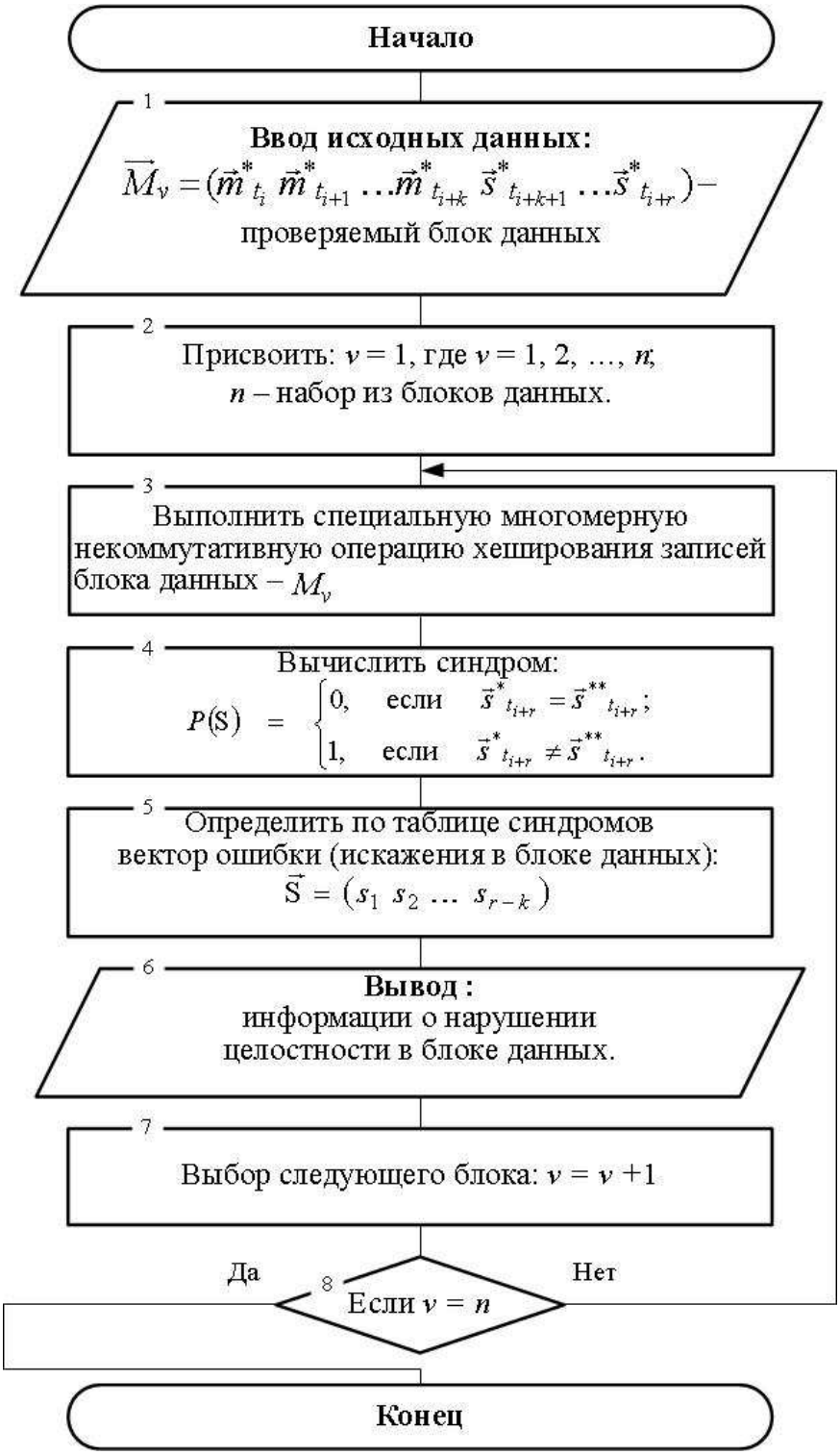

Рис. 5 – Алгоритм контроля целостности данных

Пример: Для построения ЛСХК $(7, 4)$ блока $\vec{M}_v = (\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4)$ используем систему линейно независимых векторов, которая в теории линейных кодов используется для построения кода Хемминга $(7, 4)$:

$$(\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4)\otimes\left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)=\left(\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4\ \vec{s}_1\ \vec{s}_2\ \vec{s}_3\right).$$

Полученная схема построения ЛСХК поясняется с помощью рисунка 6.

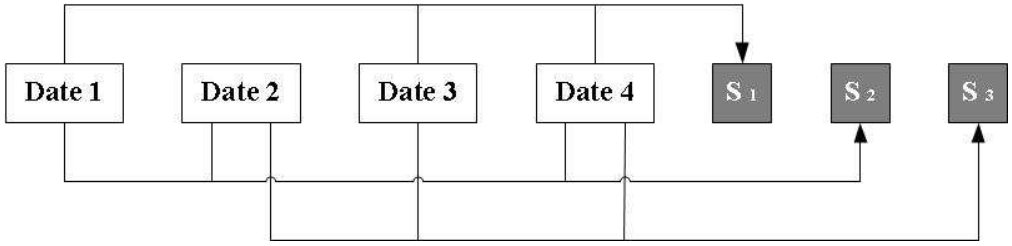

Рис. 6 – Схема получения ЛСХК $(7, 4)$

Контроль целостности защищенного блока данных $(\vec{m}_1^{\,*}\ \vec{m}_2^{\,*}\ \vec{m}_3^{\,*}\ \vec{m}_4^{\,*}\ \vec{s}_1^{\,*}\ \vec{s}_2^{\,*}\ \vec{s}_3^{\,*})$.

Вычисление:

$$(\vec{m}_1^{\,*}\ \vec{m}_2^{\,*}\ \vec{m}_3^{\,*}\ \vec{m}^{\,*})_4\otimes\left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)=\left(\vec{m}_1^{\,**}\ \vec{m}_2^{\,**}\ \vec{m}_3^{\,**}\ \vec{m}_4^{\,**}\ \vec{s}_1^{\,**}\ \vec{s}_2^{\,**}\ \vec{s}_3^{\,**}\right).$$

Вычислить синдром $\vec{\mathrm{S}}=(s_1, s_2, s_3)$, соответствующий предикату:

$$P\left(\vec{\mathrm{S}}\right)\ =\ \begin{cases}0, & \text{if}\quad \vec{s}_z^{\,*}=\vec{s}_z^{\,**};\\ 1, & \text{if}\quad \vec{s}_z^{\,*}\neq\vec{s}_z^{\,**},\end{cases}$$

где $z=1, 2, 3$.

По таблице синдромов определить нарушение целостности в защищенном блоке данных $(\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4\ \vec{s}_1\ \vec{s}_2\ \vec{s}_3)$:

Таблица 1 Пример таблицы синдромов ($[\vec{\bullet}_i]$– запись данных с нарушением целостности)

Синдром	Локализация ошибки
110	$[\vec{m}_1]\vec{m}_2\ \vec{m}_3\ \vec{m}_4\ \vec{s}_1\ \vec{s}_2\ \vec{s}_3$
011	$\vec{m}_1\ [\vec{m}_2]\vec{m}_3\ \vec{m}_4\ \vec{s}_1\ \vec{s}_2\ \vec{s}_3$
101	$\vec{m}_1\ \vec{m}_2\ [\vec{m}_3]\vec{m}_4\ \vec{s}_1\ \vec{s}_2\ \vec{s}_3$
111	$\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ [\vec{m}_4]\vec{s}_1\ \vec{s}_2\ \vec{s}_3$
100	$\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4\ [\vec{s}_1]\vec{s}_2\ \vec{s}_3$
010	$\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4\ \vec{s}_1\ [\vec{s}_2]\vec{s}_3$
001	$\vec{m}_1\ \vec{m}_2\ \vec{m}_3\ \vec{m}_4\ \vec{s}_1\ \vec{s}_2\ [\vec{s}_3]$
000	Ошибки нет

5. Вывод: Введено понятие ЛСХК. С помощью математического аппарата теории систем векторов выполнено обоснование и разработка алгоритма построения ЛСХК для обеспечения целостности данных в АС, который позволяет для заданного уровня защищенности данных уменьшить избыточность контрольной информации. Показано, что правила построения ЛСХК аналогичны правилам построения линейных избыточных кодов, в частности кодов Хемминга. Таким образом, хорошо разработанная в настоящее время теория линейных избыточных кодов может быть использована в новой для нее области – построения ЛСХК.

Литература:

1. Доктрина информационной безопасности Российской Федерации от 9 сентября 2000 г. № Пр-1895.

2. ГОСТ Р 50739 – 95 (переиздан 2006). «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования». – М.: Госстандарт России, 1996.

3. ГОСТ Р 50739 – 95. «Средства вычислительной техники. Защита от несанкционированного доступа к информации». – М.: Госстандарт России, 1995.

4. Савин, С.В. Защищенное хранение данных аудита безопасности АС, Сборник научных трудов Шестой международной научно – технической конференции (Инфоком – 6) / С.В. Савин – г. Ставрополь: Северо-Кавказский федеральный университет, 2014. Ч. 2. – С. 480-484.

5. Трошин, С. В. Мониторинг работы пользователей корпоративных сетей / С. В. Трошин. - Москва: Автореф. дисс. на соиск. уч. степени к.ф.-м.н., 2010.

6. Отчет компании InfoWatsh, Исследование утечек конфеденциальной информации в 2014 году, [Электронный ресурс] – www.infowatsh.ru/report2014 , 2015.

7. Руководящий документ ГосТехКомиссии. «Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных (выписка)». – М.: Госстандарт России, 2008.

8. Midsize Business Security Guidance. Microsoft Corporation. Security Monitoring and Attack Detection / Microsoft Corporation. – August 2006, [Электронный ресурс] – www.microsoft.com/technet/security/midsizebusiness/default.mspx, 2006.

9. Савин, С.В. Обеспечение целостности данных подсистемы регистрации и учета автоматизированных систем на основе метода «однократной записи» / С.В. Савин, О.А. Финько // Журнал «Известия Южного федерального университета (ЮФУ). Технические науки» №5/май 2015. – С. 64-77.

10. Atsushi Harada, Masakatsu Nishigaki, Masakazu Soga, Akio Takubo, Itsukazu Nakamura. A Write-Once Data Management System, ICITA 2002. –Shizuoka University, 3-5-1 Johoku, Hamamatsu, 432-8011, Japan, 2002.

11. Уоррен, Г. Подсчет битов: алгоритмические трюки для программистов (Hacker's Delight) / Генри С. Уоррен, мл. – М.: «Вильяме», 2007.

12. Морелос-Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса; перевод с англ. В. Б. Афанасьева. – М.: Техносфера, 2006.

13. Хемминг, Р.В. Теория кодирования и теория информации: Пер. с англ. – М.: «Радио и связь», 1983. – 176 с., ил.

14. Д. Кнут. Искусство программирования для ЭВМ, сортировка и поиск – М.: «Мир», 1978. – 844 с., ил.

15. Menezes, A. Handbook of Applied Criptography. / A. Menezes, P. van Oorschot, S. Vanstone. CRC Press, Inc., 1996.

16. Biham, E., Dunkelman, O. A framework for iterative hash functions / E. Biham, O. Dunkelman – HAIFA, ePrint Archive, Report 2007/278, [Электронный ресурс] – eprint.iacr.org/2007/278, July, 2007

17. Wang, X., Yu, H. How to Break MD5 and Other Hash Functions / X. Wang, H. Yu, EUROCRYPT 2005, LNCS 3494, pp. 19-35, Springer-Verlag, 2005.

18. Bellare, M. New Proofs for NMAC and HMAC: Security without Collision-Resistance / M. Bellare – CRYPTO 2006, ePrint Archive, Report 2006/043, [Электронный ресурс] – eprint.iacr.org/2006/043.pdf, 2006.

References

1. Doktrina informacionnoj bezopasnosti Rossijskoj Federacii ot 9 sentjabrja 2000 g. № Pr-1895.

2. GOST R 50739 – 95 (pereizdan 2006). «Sredstva vychislitel'noj tehniki. Zashhita ot nesankcionirovannogo dostupa k informacii. Obshhie tehnicheskie trebovanija». – M.: Gosstandart Rossii, 1996.

3. GOST R 50739 – 95. «Sredstva vychislitel'noj tehniki. Zashhita ot nesankcionirovannogo dostupa k informacii». – M.: Gosstandart Rossii, 1995.

4. Savin, S.V. Zashhishhennoe hranenie dannyh audita bezopasnosti AS, Sbornik nauchnyh trudov Shestoj mezhdunarodnoj nauchno – tehnicheskoj konferencii (Infokom – 6) / S.V. Savin – g. Stavropol': Severo-Kavkazskij federal'nyj universitet, 2014. Ch. 2. – S. 480-484.

5. Troshin, S. V. Monitoring raboty pol'zovatelej korporativnyh setej / S. V. Troshin. Moskva: Avtoref. diss. na soisk. uch. stepeni k.f.-m.n., 2010.

6. Otchet kompanii InfoWatsh, Issledovanie utechek konfedencial'noj informacii v 2014 godu, [Jelektronnyj resurs] – www.infowatsh.ru/report2014 , 2015.

7. Rukovodjashhij dokument GosTehKomissii. «Bazovaja model' ugroz bezopasnosti personal'nyh dannyh pri ih obrabotke v informacionnyh sistemah personal'nyh dannyh (vypiska)». – M.: Gosstandart Rossii, 2008.

8. Midsize Business Security Guidance. Microsoft Corporation. Security Monitoring and Attack Detection / Microsoft Corporation. – August 2006, [Jelektronnyj resurs] – www.microsoft.com/technet/security/midsizebusiness/default.mspx, 2006.

9. Savin, S.V. Obespechenie celostnosti dannyh podsistemy registracii i ucheta avtomatizirovannyh sistem na osnove metoda «odnokratnoj zapisi» / S.V. Savin, O.A. Fin'ko // Zhurnal «Izvestija Juzhnogo federal'nogo universiteta (JuFU). Tehnicheskie nauki» №5/maj 2015. – S. 64-77.

10. Atsushi Harada, Masakatsu Nishigaki, Masakazu Soga, Akio Takubo, Itsukazu Nakamura. A Write-Once Data Management System, ICITA 2002. –Shizuoka University, 3-5-1 Johoku, Hamamatsu, 432-8011, Japan, 2002.

11. Uorren, G. Podschet bitov: algoritmicheskie trjuki dlja programmistov (Hacker's Delight) / Genri S. Uorren, ml. – M.: «Vil'jame», 2007.

12. Morelos-Saragosa, R. Iskusstvo pomehoustojchivogo kodirovanija. Metody, algoritmy, primenenie / R. Morelos-Saragosa; perevod s angl. V. B. Afanas'eva. – M.: Tehnosfera, 2006.

13. Hemming, R.V. Teorija kodirovanija i teorija informacii: Per. s angl. – M.: «Radio i svjaz'», 1983. – 176 s., il.

14. D. Knut. Iskusstvo programmirovanija dlja JeVM, sortirovka i poisk – M.: «Mir», 1978. – 844 s., il.

15. Menezes, A. Handbook of Applied Criptography. / A. Menezes, P. van Oorschot, S. Vanstone. CRC Press, Inc., 1996.

16. Biham, E., Dunkelman, O. A framework for iterative hash functions / E. Biham, O. Dunkelman – HAIFA, ePrint Archive, Report 2007/278, [Jelektronnyj resurs] – eprint.iacr.org/2007/278, July, 2007

17. Wang, X., Yu, H. How to Break MD5 and Other Hash Functions / X. Wang, H. Yu, EUROCRYPT 2005, LNCS 3494, pp. 19-35, Springer-Verlag, 2005.

18. Bellare, M. New Proofs for NMAC and HMAC: Security without Collision-Resistance / M. Bellare – CRYPTO 2006, ePrint Archive, Report 2006/043, [Jelektronnyj resurs] – eprint.iacr.org/2006/043.pdf, 2006.