ГИБРИДНЫЙ КРИПТО-КОДОВЫЙ МЕТОД КОНТРОЛЯ И ВОССТАНОВЛЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ ДЛЯ ЗАЩИЩЁННЫХ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИХ СИСТЕМ

Диченко С.А.¹, Финько О.А.²

В работе представлено решение задачи контроля и восстановления целостности информации в современных системах хранения данных защищенных информационно-аналитических систем с учетом особенностей их функционирования в условиях деструктивных воздействий злоумышленника и среды.

Цель исследования – повышение отказоустойчивости систем хранения данных защищенных информационно-аналитических систем с одновременным снижением вводимой избыточности.

Метод исследования – агрегирование методов контроля целостности данных, основанных на использовании функции хэширования, и восстановления целостности данных за счет применения избыточных модулярных кодов, учитывающие структуру многомерного представления информации в современных системах хранения данных защищенных информационно-аналитических систем.

Результат исследования – оригинальные схемы совместного использования известных решений, учитывающие структуру многомерного представления информации в современных системах хранения данных защищенных информационно-аналитических системах. Разработанный метод в сравнении с наиболее популярными из существующих решений комплексного обеспечения целостности информации позволил расширить функциональные возможности систем хранения данных по обеспечению проверки достоверности данных после их восстановления. Полученные результаты позволяют обеспечить защищенность информационно-аналитических систем в условиях деструктивных воздействий злоумышленника и среды на уровне аппаратных и программных затрат, соответствующих наиболее эффективным методам обеспечения надежности.

Ключевые слова: системы в защищенном исполнении, системы хранения данных, контроль целостности информации, восстановление целостности информации, хэш-функция, многозначные избыточные коды, коды, контролирующие ошибки.

DOI: 10.21681/2311-3456-2019-6-17-36

Введение

Системы хранения данных (СХД) в современных условиях эволюционного роста объема и ценности информации, обрабатываемой в информационных системах различного назначения, получили свое широкое распространение и стали их неотъемлемой частью. В то же время, наибольшую значимость СХД приобретают при обеспечении принятия решения пользователями защищенных информационно-аналитических систем (ИАС), центральной подсистемой которых они являются. Это обуславливается тем, что правильность принимаемого пользователем ИАС решения определяется, в том числе, и достоверностью исходных данных – информации, хранящейся в СХД ИАС. Поэтому, защита информации, хранящейся в СХД ИАС, является актуальной задачей. Особую актуальность решение задачи защиты информации, в частности, контроля и обеспечения ее целостности, приобретает в условиях деструктивных воздействий на СХД ИАС злоумышленника и среды.

Представление информации в современных системах хранения данных

Для ИАС, наряду с информационными системами другого назначения, критичным является время получения ответов на запросы пользователей системы, поэтому для них характерно представление информации не в виде реляционных таблиц, а в виде упорядоченных многомерных массивов в виде гиперкубов [1]. Такое построение СХД позволяет лучше справляться с выполнением сложных нерегламентированных запросов и обеспечивает поддержку систем, ориентированных на аналитическую обработку информации.

Многомерная модель данных, лежащая в основе построения СХД ИАС, опирается на концепцию многомерных кубов, или гиперкубов. Они представляют собой упорядоченные многомерные массивы, которые также часто называют OLAP-кубами (On-Line Analytical Processing – оперативная аналитическая обработка).

1 Диченко Сергей Александрович, кандидат технических наук, старший преподаватель кафедры, Краснодарское высшее военное училище, г. Краснодар, Россия. E-mail: dichenko.sa@yandex.ru

2 Финько Олег Анатольевич, доктор технических наук, профессор, профессор кафедры, академический советник Российской академии ракетных и артиллерийских наук (РАРАН), Краснодарское высшее военное училище, г. Краснодар, Россия. E-mail: ofinko@yandex.ru.
Web: <http://www.mathnet.ru/person40004>.

Рис. 1. Принцип организации многомерного куба данных

Технология OLAP представляет собой методику оперативного извлечения нужной информации из больших массивов данных и формирования соответствующих отчетов [2].

В соответствии с теорией многомерных матриц [3] данные, хранящиеся в многомерном кубе, могут быть представлены P -мерной матрицей (многомерной матрицей), под которой понимается совокупность элементов $C_{i_1 \dots i_p}$, где индексы $i_1 \dots i_p$ принимают значения от

1 до n_a соответственно, где $a = 1, \dots, p$. P -мерная матрица содержит $n_1 \times n_2 \times \dots \times n_p$ элементов и обозначается как $C_M = [c_{i_1 \dots i_p}]$.

Многомерная модель данных, хранящихся в СХД ИАС, представлена на рисунке 1.

Многомерный куб данных можно рассматривать как систему координат с осями, например, для трехмерного куба: x, y, z , по которым откладываются подблоки:

$$\begin{aligned}
 &m_{1,1,1}, m_{1,1,2}, \dots, m_{1,1,k}; m_{1,2,1}, m_{1,2,2}, \dots, m_{1,2,k}; \dots; m_{1,n,1}, m_{1,n,2}, \dots, m_{1,n,k}; \\
 &m_{2,1,1}, m_{2,1,2}, \dots, m_{2,1,t}; m_{2,2,1}, m_{2,2,2}, \dots, m_{2,2,t}; \dots; m_{2,n,1}, m_{2,n,2}, \dots, m_{2,n,t}; \dots; \\
 &m_{k,1,1}, m_{k,1,2}, \dots, m_{k,1,t}; m_{k,2,1}, m_{k,2,2}, \dots, m_{k,2,t}; \dots; m_{k,n,1}, m_{k,n,2}, \dots, m_{k,n,t}
 \end{aligned}$$

В такой системе каждому подблоку $m_{r,p,q}$ ($r = 1, \dots, k$; $p = 1, \dots, n$; $q = 1, \dots, t$) блока данных, хранящегося в СХД ИАС соответствует определенная ячейка, в которой размещается его содержимое. В классических OLAP-кубах, содержимое называется числовыми показателями (фактами), связанными с данным набором, образа между объектами и их числовыми характеристиками устанавливается однозначная связь.

Таким образом, информация в СХД ИАС является логически целостной. Это уже не просто наборы строковых и числовых значений, которые в случае реляционной модели нужно получать из различных таблиц, а целостные структуры с однозначными связями, что делает преимущества многомерного подхода очевидными.

Особенности обеспечения целостности информации

Известно [1], что основная цель функционирования любой ИАС – обеспечение доступа к ее данным в любое время для проведения требуемого анализа с высокой степенью наглядности и достоверности. В результате пользователи ИАС получают возможность самостоятельно, а не через программистов, извлекать необходимую им информацию при помощи достаточно простого и понятного интерфейса.

Классические подходы к обеспечению целостности информационных массивов основаны на методах идентичной или неидентичной избыточности (соответственно хранение копий либо хранение предыстории), в соответствии с которыми, как правило, для конкретной системы выбирается эффективная стратегия резервирования [4].

Безусловным достоинством большинства классических методов резервирования является их простота, недостатками – высокая степень дублирования информации, неэффективное использование емкости носителей информации, в ряде случаев – неудовлетворительно большое время восстановления информации [5].

Исследование классических способов контроля и обеспечения целостности информации

Задача обеспечения целостности данных является сложной, ввиду своей комплексности, так как включает в себя не только контроль целостности данных, но и ее обеспечение, что подразумевает восстановление данных, целостность которых была нарушена по каким-либо причинам.

Существуют различные способы решения задачи контроля и обеспечения целостности данных, среди которых наибольший интерес представляют следующие.

Известны способы контроля целостности данных за счет вычисления значений контрольных сумм и сравнения их со значениями эталонных [6, 7], а также способы, основанные на применении криптографических методов: ключевое и бесключевое хэширование, сред-

ства электронной подписи [8-11]. Недостатком данных способов является отсутствие возможности без ввода дополнительного механизма восстановления данных обеспечить их целостность.

Известны способы обеспечения целостности данных за счет применения различных видов резервирования (с использованием программно-аппаратной или программной реализации технологии RAID (Redundant Array of Independent Disks) (RAID-массивы)) [12, 13], методы дублирования, методы избыточного кодирования [14-18]. Недостатком данных способов является неспособность контроля целостности данных с криптографической достоверностью, а также высокая избыточность.

Представленные решения показывают, что часть способов позволяет осуществить контроль целостности данных путем сравнения значений эталонных и вычисляемых хэш-кодов хэш-функции (контрольных сумм) при запросе на использование хранящихся в памяти данных (рис. 2). Однако отсутствие механизмов их восстановления не позволяет обеспечить их целостность.

Другие способы, напротив, позволяют обеспечить целостность данных путем их восстановления, к примеру, из резервной копии (рис. 3). Однако их практическая реализация без возможности осуществления предварительного контроля целостности данных является неэффективной.

Отдельные способы позволяют осуществить контроль и обеспечить целостность данных, однако, ценой высокой избыточности (рис. 4).

Наиболее популярными являются решения комплексного обеспечения целостности данных, где после осуществления контроля целостности данных возможно восстановление их целостности, что достигается за счет последовательного применения сначала криптографических преобразований, а затем – технологии резервного копирования (рис. 5).

В этом случае контроль и обеспечение целостности данных актуальны как для систем типа RAID, где все

Рис. 2. Схема, поясняющая процедуру контроля целостности данных

Рис. 3. Схема, поясняющая процедуру восстановления данных из резервной копии

Рис. 4. Схема, поясняющая процедуру контроля и обеспечения целостности данных (ценой высокой избыточности)

Рис. 5. Схема, поясняющая процедуру контроля и обеспечения целостности данных за счет комбинирования известных способов

носители размещены в одном конструктивном блоке, так и для распределенных систем хранения, то есть для сетевых хранилищ.

В настоящее время продолжается работа по разработке новых способов контроля и обеспечения целостности данных [19-25], в том числе и для облачных хранилищ [26].

Таким образом, для комплексного обеспечения целостности данных в СХД ИАС необходимо осуществить агрегирование известных классических решений. Однако, выбор известных способов для последовательного осуществления контроля целостности данных, а затем, в случае необходимости, восстановления их целостности является сложной задачей, так как совместное их использование должно в полной мере учитывать особенности функционирования ИАС в условиях деструктивных воздействий злоумышленника и среды.

Анализ и выбор существующих решений

Известен способ, где перед записью в RAID-массив (после чтения) данные зашифровываются (расшифровываются) специально выделенным устройством, подключенным к PCI-шине. При этом ключ зашифрования (расшифрования) считывается с внешнего запоминающего устройства и/или запрашивается у пользователя³. В другом способе перед записью в массив данные разбиваются на несколько сегментов, после чего от данных из каждого сегмента отдельно вычисляются контрольные суммы. Сегменты данных и контрольные суммы далее распределяются по дискам RAID-массива⁴.

Предложен способ защиты данных в сетевом хранилище, где пользовательский запрос на чтение (запись) данных сначала проходит процедуру авторизации, и только в случае разрешения операции выполняется соответственно расшифрование (зашифрование) данных

на сетевом хранилище⁵. Ключи зашифрования (расшифрования) хранятся на стороне клиента.

Предложен другой вариант комбинированной защиты, где данные хранятся в облаке, а модуль шифрования хранится не на стороне клиента, а на стороне провайдера облачного хранилища⁶. Это решение предназначено, как правило, для защиты резервных копий данных в облаке, хотя оригинальные данные хранятся на стороне клиента в исходном виде. При этом для обеспечения защиты данных файл с данными сначала разбивается на части, а затем каждая часть преобразуется с помощью криптографического алгоритма и записывается на один или несколько носителей в облаке. Защита обеспечивается в случае, когда данные утрачиваются на стороне клиента. В этом случае резервная копия восстанавливается из облака.

Недостатками представленных решений является высокая избыточность и отсутствие возможности без ввода дополнительного механизма контроля осуществить проверку достоверности восстановленных данных в случае нарушения их целостности, что не учитывает в полной мере всех особенностей функционирования ИАС в условиях деструктивных воздействий, в первую очередь, злоумышленника.

Для устранения указанных недостатков известных решений предлагается для осуществления контроля целостности данных, а также проверки их достоверности после восстановления в случае нарушения целостности использовать криптографические методы, в частности, функцию хэширования, разработанную специально для этих целей. В свою очередь, процедуру восстановления данных можно выполнять посредством избыточных модульных кодов (МК), применение математического аппарата которых позволит обеспечить минимальную избыточность.

Совместное использование функции хэширования и избыточных МК обеспечит построение оригинальных

3 US Patent N8209551 publ. 26/06/2012.

4 US Patent Application N20050081048 publ. 14/04/2005.

5 US Patent N7752676 publ. 06/07/2010.

6 US Patent Application N20110107103 publ. 05/05/2011.

схем, учитывающих структуру многомерного представления данных в современных СХД ИАС, что является важным при учете уровня аппаратных и программных затрат, соответствующих наиболее эффективным методам обеспечения целостности хранящейся в них информации.

Декомпозиция многомерного массива информации, хранящейся в системах хранения данных

В процессе поиска и извлечения из гиперкуба необходимой информации над его измерениями производится ряд действий, наиболее типичными из которых являются: сечение (срез); транспонирование; свертка; детализация [2].

Для декомпозиции многомерного массива информации, хранящейся в СХД ИАС, используя известные правила, выполним его расчленение на сечения (сечения гиперкуба данных), которое заключается в выделении подмножества ячеек гиперкуба при фиксировании значения одного (рис. 6а) или нескольких измерений (рис. 6б).

В результате расчленения на сечения получается срез или несколько срезов, каждый из которых содержит информацию, связанную со значением измерения, по которому он был построен.

Структурно-параметрический синтез системы параллельного контроля и обеспечения целостности данных

Рассмотрим сечение гиперкуба данных, полученное при фиксировании значения одного измерения (рис. 6а), где подблоки блоков данных в соответствии с известными правилами [3] нумеруются по двум осям системы координат.

Для осуществления контроля и обеспечения целостности рассмотрим блоки данных M_i ($i = 1, 2, \dots, n$), подлежащие защите, представленные в виде подблоков фиксированной длины

$$M_i = \{ m_{i,1} \parallel m_{i,2} \parallel \dots \parallel m_{i,n} \},$$

где $m_{i,j} \in Z_{p_{i,j}}$; $i, j = 1, \dots, n$; $p_{i,j} \in N$, « $\parallel$ » – операция конкатенации, n – количество блоков данных M_i , подлежащих защите, а также подблоков фиксированной длины в каждом рассматриваемом блоке данных M_i длиной 512 бит.

Рис. 6. Сечения гиперкуба данных: а) при фиксировании значения одного измерения, б) при фиксировании – двух измерений

Получим матрицу:

$$W = \left[\begin{array}{cccc} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{array} \right] \left. \vphantom{\begin{array}{cccc} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{array}} \right\} \begin{array}{l} 512 \text{ бит} \\ 512 \text{ бит} \end{array}.$$

Пример 1

При представлении блоков данных M_i подблоками $m_{i,1}, m_{i,2}, \dots, m_{i,n}$ размером по 64 бит их количество равняется $n = 8$.

Матрица W примет вид:

$$W = \left[\begin{array}{ccccccccc} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} & m_{1,7} & m_{1,8} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} & m_{2,7} & m_{2,8} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} & m_{3,6} & m_{3,7} & m_{3,8} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} & m_{4,6} & m_{4,7} & m_{4,8} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} & m_{5,6} & m_{5,7} & m_{5,8} \\ m_{6,1} & m_{6,2} & m_{6,3} & m_{6,4} & m_{6,5} & m_{6,6} & m_{6,7} & m_{6,8} \\ m_{7,1} & m_{7,2} & m_{7,3} & m_{7,4} & m_{7,5} & m_{7,6} & m_{7,7} & m_{7,8} \\ m_{8,1} & m_{8,2} & m_{8,3} & m_{8,4} & m_{8,5} & m_{8,6} & m_{8,7} & m_{8,8} \end{array} \right] \left. \vphantom{\begin{array}{ccccccccc} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} & m_{1,7} & m_{1,8} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} & m_{2,7} & m_{2,8} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} & m_{3,6} & m_{3,7} & m_{3,8} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} & m_{4,6} & m_{4,7} & m_{4,8} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} & m_{5,6} & m_{5,7} & m_{5,8} \\ m_{6,1} & m_{6,2} & m_{6,3} & m_{6,4} & m_{6,5} & m_{6,6} & m_{6,7} & m_{6,8} \\ m_{7,1} & m_{7,2} & m_{7,3} & m_{7,4} & m_{7,5} & m_{7,6} & m_{7,7} & m_{7,8} \\ m_{8,1} & m_{8,2} & m_{8,3} & m_{8,4} & m_{8,5} & m_{8,6} & m_{8,7} & m_{8,8} \end{array}} \right\} \begin{array}{l} 512 \text{ бит} \\ 512 \text{ бит} \end{array}.$$

Для осуществления контроля целостности к блокам данных M_i применяется хэш-функция [27].

Определение 1. Под хэш-функцией $h(M)$ понимается функция, отображающая блоки бит $M \in (0,1)^*$ в строки бит фиксированной длины $S \in (0,1)^q$: $h(M) = S$, где « $(\bullet)^*$ » – произвольный размер блока бит, « $(\bullet)^q$ » – фиксированный размер блока бит, $q \in N$,

и удовлетворяющая следующим свойствам:

- по данному значению хэш-функции $S \in (0,1)^q$ сложно вычислить исходные данные $M \in (0,1)^*$, отображаемые в это значение;
- для заданных исходных данных $M_1 \in (0,1)^*$ сложно вычислить другие исходные данные $M_2 \in (0,1)^*$, отображаемые в то же значение

хэш-функции, то есть $h(M_1) = h(M_2)$, где

- $M_1 \neq M_2$;
- сложно вычислить какую-либо пару исходных данных (M_1, M_2) , где $M_1 \neq M_2$, $M_t \in (0,1)^*$, $t = 1, 2$, отображаемых в одно и то же значение хэш-функции, то есть $h(M_1) = h(M_2)$.

Определение 2. Под хэш-кодом понимается строка бит $S \in (0,1)^q$, являющаяся выходным результатом

хэш-функции $h(M)$.

Определение 3. Строки бит $M \in (0,1)^*$, которые хэш-функция h отображает в хэш-код $S \in (0,1)^q$, называется блоком данных.

Полученные хэш-коды S_i хэш-функции $h(M_i)$ (или $h_k(M_i)$, k – секретный ключ) от блоков данных M_i являются эталонными:

$$\Psi = \left[\begin{array}{cccc} \overbrace{m_{1,1} \ m_{1,2} \ \cdots \ m_{1,n}}^{512 \text{ бит}} & \rightarrow & \overbrace{s_{1,1} \ s_{1,2} \ \cdots \ s_{1,n}}^{512 \text{ бит}} \\ m_{2,1} \ m_{2,2} \ \cdots \ m_{2,n} & \rightarrow & s_{2,1} \ s_{2,2} \ \cdots \ s_{2,n} \\ \vdots & & \vdots \\ m_{n,1} \ m_{n,2} \ \cdots \ m_{n,n} & \rightarrow & s_{n,1} \ s_{n,2} \ \cdots \ s_{n,n} \end{array} \right] \left. \vphantom{\begin{array}{c} m_{1,1} \\ m_{2,1} \\ \vdots \\ m_{n,1} \end{array}} \right\} 512 \text{ бит},$$

где $S_i = \{s_{i,1} \| s_{i,2} \| \cdots \| s_{i,n}\}; s_{i,j} \in \{0,1\} \ (i, j = 1, \dots, n)$.

Пример 2.

В таблице 1 представлены вычисленные в соответствии с действующим государственным стандартом

(ГОСТ Р 34.11-2012) хэш-коды S_i хэш-функции $h(M_i)$

от произвольных блоков данных M_i (строки бит представлены в 16-ричной системе счисления).

Таблица 1

Таблица значений вычисленных хэш-кодов хэш-функции от блоков данных

i	Блоки данных M_i	Хэш-коды S_i хэш-функции $h(M_i)$
1	HJSA-DJFT-BAWL-344H-B235-7634-7856-2309-4GH0-234G-H234-9087-GT30-14GY-1390-7841-GY03-74HG-1304-8BV1-0348-FH10-87TW-S0A8-7F01-2370-47A0-SE4A-OQW3-7401-OQ23-74TG	298F-09B4-2614-3A65-1E4E-B5E3-E11B-2DCE-A562-6AB1-1C27-E0F4-2C75-8DEF-B780-C4A3-82C5-03D2-FBCD-A6C7-7CD8-D357-7FB5-3F4C-9B8B-B767-87E3-1564-8F39-084C-41D4-FD18
2	SKDJ-FHAL-SKJ3-4H5L-K34J-H26P-4049-8TYE-RJLG-HSLD-RNMT-L23K-J45H-2349-8671-2345-9082-634T-GKJA-DFNB-LZCF-JKGH-W034-5782-0398-4563-405H-234L-GKBJ-LESL-4857-2034	2846-BD24-7955-88E2-9FE2-32D5-8580-0D01-79BF-55B4-DA6A-0F89-BD75-6B72-457C-8306-C2D1-011B-EODA-21BE-FD94-AC57-1B0E-2FEB-AD2B-A792-292C-CDBE-A804-E8CB-8D75-FCAB
3	AGHR-KLJH-TLWE-KRJT-HWLE-KSLD-KFJG-H3EQ-WOPU-IRWT-HELK-RZV2-3573-094G-HQW3-0I45-BH23-K45H-G234-0957-8126-Y398-4712-T3HF-BF01-2370-47A0-SEAD-JKLF-GBEL-RKTJ-HELK	C70E-1BB9-2FBD-0A40-B631-08F5-5798-F236-1F7B-A0D1-24D3-61D7-C3F0-E7F7-156E-714C-EDA9-E913-7DFC-3EED-8295-4E18-D50F-97BD-ABFD-FODA-8138-5A83-14A9-75DA-DD2B-B5E8
4	QWEU-ITHL-SDKJ-BVAL-SERU-KOQW-EIRB-GFAS-LDFK-JBGA-LDFK-BGAL-ERUI-TYOE-RIUV-HAOP-WI34-U5GO-Q234-UIYU-3478-95YQ-34IL-JBGF-KLSD-BGAL-SDKT-HOER-IUTY-WEOR-ITUH-ERLJ	54A8-9890-6DDA-2F2A-2846-13BE-B475-F307-D649-E9F9-F3B4-2D29-B3DD-BA1A-F31B-3AAB-E116-E505-4EBD-B510-4063-187E-FFDA-0431-B6C5-37D6-8A45-F9BF-3F64-1E13-42D8-F520
5	HAEL-OUTY-AOWE-BGJA-KLSD-NFBG-AERT-IOUW-ERHG-LKSD-BFVG-ASDR-IOUT-YWEO-PIRZ-SGLD-FKJS-GHDL-FTUY-EOEB-TKER-HJGT-3457-6239-487T-GFKB-VKJH-GFRU-TYWK-LERH-JGT3-45GO	9ACB-B37E-3417-0B9A-1E78-9700-9C79-18A4-6B8E-F85D-92F9-F5F3-412C-6177-EE2E-9971-DD1D-E116-7AE8-3161-AD8B-566F-6C9D-E1C0-DD7C-9523-BA4B-EC93-E309-C5E5-D9AD-EE9C
6	DEFB-GAEK-JGRT-6W4K-J5HG-TKEG-FRAD-KFBG-AHRG-FTEK-RJHG-TKEJ-RBVG-FKEJ-RHGT-WERU-IGFA-SDKF-HBGK-AJVR-HTKA-HGFI-ASDU-FYVA-KERH-BTKE-RHJT-GWEI-ORUT-YGDF-OVIH-BASD	2970-9095-E758-2AAE-4458-8B6F-356D-FA47-C09A-A06C-72BF-1B63-C716-18BF-4C54-AC5D-5270-5A0B-2DD2-8FAC-2B3D-211C-06BD-8717-483D-8782-9D91-90E7-02BE-D78D-D432-ADCO
7	FEHR-GTKW-EJBG-RKER-JWGO-KRIB-TWEI-OUGH-RILB-GDIF-UGYO-IE45-U395-7861-3948-7523-49OY-GTK3-Q2J4-HVTK-QJHG-UIYU-IERG-JKSE-RHKG-SDJF-BGSK-DJFG-HKWE-GRIQ-WEUG-TKEH	25D6-F831-C269-E656-D4FA-E5DA-8B29-2766-68E9-D90E-69E2-4BE2-6BFE-DA59-E2F3-9075-6389-C2D9-583E-7F67-EDAD-73F4-59C6-FC31-3D38-8E06-EEF6-91FA-8C65-5DD9-1416-0001
8	AFSJ-KBVT-RWEK-JTHG-WKEJ-R24U-ITY3-4730-14GY-4BLZ-ASDF-NAEO-RIUW-THKL-E45J-634K-5IUG-TKSD-GT3K-WUIT-Y347-TY34-UIYU-3BLZ-CFJK-GHW4-74UA-SEGT-RKWE-RJTH-BVGD-SKRI	075E-9B94-5909-C8A1-2624-AFE6-6E21-0F09-C425-0BAB-903D-897B-F92B-B9A8-2E41-EA3C-DE44-CEE9-DC6E-1A97-40E4-5FCF-85E8-A873-8B6D-33B8-91C9-A28C-E0BE-3530-C4C7-E986

Далее рассматриваются блоки данных M_j ($j = 1, 2, \dots, n$), представленные подблоками:

$$m_{1,1}, m_{2,1}, \dots, m_{n,1};$$

$$m_{1,2}, m_{2,2}, \dots, m_{n,2};$$

$$\dots;$$

$$m_{1,n}, m_{2,n}, \dots, m_{n,n}.$$

Подблоки $m_{i,j}$ рассматриваемых блоков данных M_j интерпретируются как наименьшие неотрицательные вычеты по сгенерированным, упорядоченным по

величине, попарно простым модулям $P_{i,j}$, подбираемым по правилам, представленным в [28], и образуют информационный суперблок МК.

После операции расширения формируются избыточные подблоки:

$$m_{n+1,1}, m_{n+2,1}, \dots, m_{k,1};$$

$$m_{n+1,2}, m_{n+2,2}, \dots, m_{k,2};$$

$$\dots;$$

$$m_{n+1,n}, m_{n+2,n}, \dots, m_{k,n},$$

совокупность которых, а также подблоки, образующие информационный суперблок МК, образуют кодовый вектор МК.

Получим матрицу с избыточными подблоками кодового вектора МК:

$$Y = \left[\begin{array}{cccc|cccc} \overbrace{m_{1,1} \quad m_{1,2} \quad \dots \quad m_{1,n}}^{512 \text{ бит}} & \overbrace{s_{1,1} \quad s_{1,2} \quad \dots \quad s_{1,n}}^{512 \text{ бит}} \\ \overbrace{m_{2,1} \quad m_{2,2} \quad \dots \quad m_{2,n}} & \overbrace{s_{2,1} \quad s_{2,2} \quad \dots \quad s_{2,n}} \\ \vdots & \vdots \\ \overbrace{m_{n,1} \quad m_{n,2} \quad \dots \quad m_{n,n}} & \overbrace{s_{n,1} \quad s_{n,2} \quad \dots \quad s_{n,n}} \\ \downarrow & \downarrow \\ \overbrace{m_{n+1,1} \quad m_{n+1,2} \quad \dots \quad m_{n+1,n}} & \\ \overbrace{m_{n+2,1} \quad m_{n+2,2} \quad \dots \quad m_{n+2,n}} & \\ \vdots & \\ \overbrace{m_{k,1} \quad m_{k,2} \quad \dots \quad m_{k,n}} & \end{array} \right] \left. \vphantom{\begin{array}{c} m_{1,1} \\ m_{2,1} \\ \vdots \\ m_{n,1} \\ m_{n+1,1} \\ m_{n+2,1} \\ \vdots \\ m_{k,1} \end{array}} \right\} 512 \text{ бит.}$$

Пусть:

$$s_{i,j} = \sum_{t=0}^{-1+\log p_{i,j}} \zeta_{i,j}^{(t)} 2^t;$$

$$m_{i,j} = \sum_{t=0}^{-1+\log p_{i,j}} \gamma_{i,j}^{(t)} 2^t;$$

$$g_{i,j} = \sum_{t=0}^{-1+\log p_{i,j}} \zeta_{i,j}^{(t)} 2^t;$$

$$\gamma_{i,j}, \zeta_{i,j}, \zeta_{i,j} \in \{0,1\}.$$

Тогда числа $s_{i,j}, m_{i,j}, g_{i,j}$ могут быть сопоставлены с двоичными векторами:

$$s_{i,j} = [\zeta_{i,j}^{(0)} \quad \zeta_{i,j}^{(1)} \quad \dots \quad \zeta_{i,j}^{(-1+\log p_{i,j})}];$$

$$m_{i,j} = [\mu_{i,j}^{(0)} \quad \mu_{i,j}^{(1)} \quad \dots \quad \mu_{i,j}^{(-1+\log p_{i,j})}];$$

$$g_{i,j} = [\zeta_{i,j}^{(0)} \quad \zeta_{i,j}^{(1)} \quad \dots \quad \zeta_{i,j}^{(-1+\log p_{i,j})}].$$

Выполним сложение i -х подблоков хэш-кодов S_i с j -ми избыточными подблоками блоков данных M_j^* кодового вектора МК:

$$G_i = S_i \oplus M_j^* = [s_{i,1} \oplus m_{n+1,j} \quad s_{i,2} \oplus m_{n+2,j} \quad \dots \quad s_{i,n} \oplus m_{k,j}],$$

где « $\oplus$ » – символ сложения в GF(2),

$$G_i = [g_{i,1} \quad g_{i,2} \quad \dots \quad g_{i,n}];$$

$$S_i = [s_{i,1} \quad s_{i,2} \quad \dots \quad s_{i,n}];$$

$$M_j^* = [m_{n+1,j} \quad m_{n+2,j} \quad \dots \quad m_{k,j}].$$

Получим матрицу:

$$\Omega = \left[\begin{array}{cccc|cccc} \text{512 бит} & & & & \text{512 бит} & & & \\ m_{1,1} & m_{1,2} & \cdots & m_{1,n} & \rightarrow & g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} & \rightarrow & g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} & \rightarrow & g_{n,1} & g_{n,2} & \cdots & g_{n,n} \end{array} \right] \left. \vphantom{\begin{array}{cccc|cccc} \end{array}} \right\} \text{512 бит.} \quad (1)$$

Результатом подготовительного этапа (рис. 7) разработанного метода является построение системы параллельного контроля и обеспечения целостности дан-

ных, подлежащих хранению, которые представляются в виде (1), что позволит в последующем осуществить контроль и обеспечить их целостность.

Рис. 7. Схема, поясняющая подготовительный этап разработанного метода

Контроль целостности данных

При запросе на использование данных (основной этап разработанного метода), выполняется контроль их целостности, возможность осуществления которого обеспечивается за счет выполнения операции расширения информационного суперблока МК [29], при этом формируются избыточные подблоки:

$$m'_{n+1,1}, m'_{n+2,1}, \dots, m'_{k,1};$$

$$m'_{n+1,2}, m'_{n+2,2}, \dots, m'_{k,2}; \dots;$$

$$m'_{n+1,n}, m'_{n+2,n}, \dots, m'_{k,n}$$

блоков данных M_j^* кодового вектора МК, где « $\bullet'$ » обозначает, что в подблоках:

$$m'_{1,1}, m'_{2,1}, \dots, m'_{n,1};$$

$$m'_{1,2}, m'_{2,2}, \dots, m'_{n,2}; \dots;$$

$$m'_{1,n}, m'_{2,n}, \dots, m'_{n,n}$$

блоков данных M_j' могли произойти изменения.

Матрица Ω с избыточными подблоками кодового вектора МК примет вид:

$$\Omega' = \left[\begin{array}{cccc|cccc} \overbrace{m'_{1,1} \quad m'_{1,2} \quad \cdots \quad m'_{1,n}}^{512 \text{ бит}} & \overbrace{g'_{1,1} \quad g'_{1,2} \quad \cdots \quad g'_{1,n}}^{512 \text{ бит}} & & & & & & \\ \overbrace{m'_{2,1} \quad m'_{2,2} \quad \cdots \quad m'_{2,n}}^{512 \text{ бит}} & \overbrace{g'_{2,1} \quad g'_{2,2} \quad \cdots \quad g'_{2,n}}^{512 \text{ бит}} & & & & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \overbrace{m'_{n,1} \quad m'_{n,2} \quad \cdots \quad m'_{n,n}}^{512 \text{ бит}} & \overbrace{g'_{n,1} \quad g'_{n,2} \quad \cdots \quad g'_{n,n}}^{512 \text{ бит}} & & & & & & \\ \downarrow & \downarrow & \cdots & \downarrow & & & & \\ \overbrace{m'_{n+1,1} \quad m'_{n+1,2} \quad \cdots \quad m'_{n+1,n}}^{512 \text{ бит}} & & & & & & & \\ \overbrace{m'_{n+2,1} \quad m'_{n+2,2} \quad \cdots \quad m'_{n+2,n}}^{512 \text{ бит}} & & & & & & & \\ \vdots & & & & & & & \\ \overbrace{m'_{k,1} \quad m'_{k,2} \quad \cdots \quad m'_{k,n}}^{512 \text{ бит}} & & & & & & & \end{array} \right] \left. \vphantom{\begin{array}{c} m'_{1,1} \\ m'_{2,1} \\ \vdots \\ m'_{n,1} \\ m'_{n+1,1} \\ m'_{n+2,1} \\ \vdots \\ m'_{k,1} \end{array}} \right\} 512 \text{ бит.}$$

Выполним обратное преобразование (здесь двоичные векторы $\mathbf{g}'_{i,j}$, $\mathbf{m}'_{i,j}$ эквивалентны числам $g'_{i,j}$, $m'_{i,j}$):

$$\begin{aligned} \mathbf{S}'_i &= \mathbf{G}'_i \oplus \mathbf{M}'_j^* = \\ &= [\mathbf{g}'_{i,1} \oplus \mathbf{m}'_{n+1,j} \quad \mathbf{g}'_{i,2} \oplus \mathbf{m}'_{n+2,j} \quad \cdots \quad \mathbf{g}'_{i,n} \oplus \mathbf{m}'_{k,j}], \end{aligned}$$

где

$$\mathbf{S}'_i = [s'_{i,1} \quad s'_{i,2} \quad \cdots \quad s'_{i,n}];$$

$$\mathbf{M}'_j^* = [\mathbf{m}'_{n+1,j} \quad \mathbf{m}'_{n+2,j} \quad \cdots \quad \mathbf{m}'_{k,j}].$$

Сравним значения полученных хэш-кодов $\mathbf{S}'_i$ со значениями ранее вычисленных эталонных хэш-кодов $\mathbf{S}_i$ и хэш-кодов $\mathbf{S}''_i$ хэш-функции $h(\mathbf{M}'_i)$.

По результатам сравнения сделаем вывод:

- об отсутствии нарушения целостности данных, при $\mathbf{S}'_i = \mathbf{S}_i$ и $\mathbf{S}'_i = \mathbf{S}''_i$;
- о нарушении целостности данных, при $\mathbf{S}'_i \neq \mathbf{S}_i$ и $\mathbf{S}'_i \neq \mathbf{S}''_i$.

Обеспечение целостности данных

При несоответствии значений сравниваемых между собой хэш-кодов, что характеризуется возникновением ошибки (нарушением целостности) в хранящихся данных, выполним ее локализацию.

Локализация обнаруженной ошибки (подблока $\tilde{m}'_{i,j}$ с нарушением целостности) выполняется первоначально по строкам матрицы Ω' (определяется i -й блок данных с нарушением целостности, в который входит подблок $\tilde{m}'_{i,j}$), а затем по столбцам (определяется j -й блок данных с нарушением целостности, в который входит подблок $\tilde{m}'_{i,j}$).

Блок данных $\tilde{M}'_i$ с нарушением целостности, подблоки которого располагаются по строке матрицы Ω' , определяется по результатам сравнения значений вычисленных и эталонных хэш-кодов хэш-функции. Блок данных $\tilde{M}'_j$ с нарушением целостности, подблоки которого располагаются по столбцу матрицы Ω' , опреде-

ляется посредством математического аппарата избыточных МК, основанного на фундаментальных положениях теории сравнения [30].

В соответствии с математическим аппаратом модулярной арифметики (МА) [28] проверяемый блок данных $\tilde{M}'_j$ интерпретируется как целое неотрицательное число A_j , однозначно представленное набором вычетов по основаниям

$$p_{1,j}, p_{2,j}, \dots, p_{n,j} < p_{n+1,j} < \dots < p_{k,j} :$$

$$A_j = (\alpha_{1,j}, \alpha_{2,j}, \dots, \alpha_{n,j}, \alpha_{n+1,j}, \dots, \alpha_{k,j}),$$

где

$$P_{n,j} = p_{1,j} p_{2,j} \dots p_{n,j} > A_j;$$

$$\alpha_{i,j} = |A|_{p_{i,j}};$$

$| \bullet |_p$ – наименьший неотрицательный вычет числа « $\bullet$ » по модулю p ;

$$j = 1, 2, \dots, n, n+1, \dots, k;$$

$$i = 1, 2, \dots, n;$$

$p_{1,j}, p_{2,j}, \dots, p_{n,j} < p_{n+1,j} < \dots < p_{k,j}$ – попарно простые.

Полученные вычеты $\alpha_{i,j}$ интерпретируются как подблоки $\tilde{m}'_{i,j}$ блока данных $\tilde{M}'_j$, то есть вычеты МА $\alpha_{1,j}, \alpha_{2,j}, \dots, \alpha_{n,j}$ интерпретируются как подблоки $m'_{1,j}, m'_{2,j}, \dots, m'_{n,j}$ и считаются информационными (информационной группой n подблоков), а $\alpha_{n+1,j}, \dots, \alpha_{k,j}$ – интерпретироваться как подблоки $m'_{n+1,j}, \dots, m'_{k,j}$ и считаются контрольными (избыточными) (контрольной (избыточной) группой $(k-n)$ подблоков). МА является в этом случае расширенной, где $P_{k,j} = p_{n,j} p_{n+1,j} \dots p_{k,j}$, и охватывает полное мно-

жество состояний, представляемых всеми k вычетами. Эта область является полным диапазоном МА $[0, P_{k,j})$ и состоит из рабочего диапазона $[0, P_{n,j})$, где $P_{n,j} = p_{1,j} p_{2,j} \dots p_{n,j}$, определяемого избыточными основаниями МА (подблоками $m'_{1,j}, m'_{2,j}, \dots, m'_{n,j}$), и диапазона $[P_{n,j}, P_{k,j})$ определяемого избыточными основаниями МА (подблоками $m'_{n+1,j}, \dots, m'_{k,j}$) и представляющего недопустимую область. Это означает, что операции над числом A_j выполняются в диапазоне $[0, P_{k,j})$, и если результат операции МА выходит за пределы $P_{n,j}$, то следует вывод об ошибке вычислений. Проверка этого правила позволяет локализовать ошибку в блоке данных M'_j матрицы Ω' .

Пример 3

Выберем систему оснований $p_1 = 2, p_2 = 3, p_3 = 5, p_4 = 7$ для которой рабочий диапазон равен $P_4 = p_1 p_2 p_3 p_4 = 2 \cdot 3 \cdot 5 \cdot 7 = 210$. Введем контрольные основания $p_5 = 11, p_6 = 13$, тогда полный диапазон определяется как $P_6 = P_4 p_5 p_6 = 210 \cdot 11 \cdot 13 = 30030$.

Вычислим ортогональные базисы системы:

$$B_1 = (1, 0, 0, 0, 0, 0) = 15015;$$

$$B_2 = (0, 1, 0, 0, 0, 0) = 20020;$$

$$B_3 = (0, 0, 1, 0, 0, 0) = 6006;$$

$$B_4 = (0, 0, 0, 1, 0, 0) = 25740;$$

$$B_5 = (0, 0, 0, 0, 1, 0) = 16380;$$

$$B_6 = (0, 0, 0, 0, 0, 1) = 6930.$$

Дано число

$$A = (1, 2, 2, 3, 6, 4) = 17.$$

При считывании данных получили

$$\tilde{A} = (1, 2, 2, 3, 1, 4).$$

Для локализации ошибки вычислим величину числа:

$$\begin{aligned} \tilde{A} &= 1 \cdot 15015 + 2 \cdot 20020 + 2 \cdot 6006 + \\ &+ 3 \cdot 25740 + 1 \cdot 16380 + 4 \cdot 6930 - R \cdot 30030 \\ &= 8207 > 210. \end{aligned}$$

Полученное число является неправильным ($\tilde{A} > 210$), что свидетельствует о нарушении целостности данных. В результате локализации, проводимой по правилам, представленным в [28], определили, что нарушена целостность цифры $\tilde{a}_5$ по основанию $p_5 = 11$.

После определения блоков данных M'_i и M'_j с нарушенной целостностью принимается решение о том, что в подблоке $\tilde{m}'_{i,j}$, находящемся на пересечении локализованных строки и столбца матрицы Ω'

произошло нарушение целостности данных. После локализации подблока $\tilde{m}'_{i,j}$ с нарушением целостности, выполним операцию реконфигурации, возможность выполнения которой обеспечивается посредством избыточных МК [28, 29].

Операция реконфигурации выполняется вычислением A^* из системы уравнений:

$$\left| A^* \right|_{p_1} = \alpha_1, \dots, \left| A^* \right|_{p_n} = \alpha_n, \dots, \left| A^* \right|_{p_k} = \alpha_k$$

по «правильным» основаниям МА:

$$A^* = \left| \tilde{\alpha}_1 B_{1,r} + \dots + \tilde{\alpha}_n B_{n,r} + \dots + \tilde{\alpha}_k B_{k,r} \right|_{P_r}, \quad (2)$$

где

$\tilde{\alpha}_i$ – ошибочный вычет;

$B_{i,r}$ – ортогональные базисы;

$$i, r = 1, \dots, n, \dots, k; i \neq r;$$

$$B_{i,r} = \frac{P_r \mu_{i,r}}{p_i}; P_r = \frac{P_k}{m_r};$$

$\mu_{i,r} \in N$, которое находится из сравнения:

$$\frac{P_r \mu_{i,r}}{p_i} \equiv 1 \pmod{p_i}.$$

Составим таблицу 2, содержащую значения пересчитанных ортогональных базисов и модулей системы при условии возникновения однократной ошибки по каждому основанию МА соответственно.

После вычисления A^* по правильным основаниям системы вычислим α_i взамен ранее исключенного из вычисления ошибочного вычета $\tilde{\alpha}_i$:

$$\alpha_i = \left| A^* \right|_{p_i} \quad (3)$$

Пример 4

В соответствии с (2) вычислим A^* (исходные данные из Примера 3), используя таблицу 2, получим:

$$\begin{aligned} A^* &= \left| \alpha_1 B_{1,r} + \dots + \tilde{\alpha}_5 B_{5,r} + \dots + \alpha_6 B_{6,r} \right|_{P_5} = \\ &= \left| 1 \cdot B_{1,r} + \dots + 0 \cdot B_{5,r} + \dots + 4 \cdot B_{6,r} \right|_{P_5} = 17. \end{aligned}$$

Таблица 2

Таблица значений ортогональных базисов и модулей системы

i	$B_{1,r}$	...	$B_{n,r}$	...	$B_{k,r}$	P_r
1	0	...	$\frac{P_1 \mu_{n,1}}{p_n}$	...	$\frac{P_1 \mu_{k,1}}{p_k}$	$p_2 \cdots p_n \cdots p_k$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n	$\frac{P_n \mu_{1,n}}{p_1}$	...	0	...	$\frac{P_n \mu_{k,n}}{p_k}$	$p_1 \cdots p_{n-1} p_{n+1} \cdots p_k$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
k	$\frac{P_k \mu_{1,k}}{p_1}$	...	$\frac{P_k \mu_{n,k}}{p_n}$	...	0	$p_1 \cdots p_n \cdots p_{k-1}$

В соответствии с (3) вычислим α_i :

$$\alpha_i = \left| A^* \right|_{p_i} = |17|_{11} = 6.$$

При выполнении реконфигурации (с учетом обнаружения и локализации двукратной ошибки по двум основаниям системы) A^* вычисляется следующим образом:

$$A^* = \left| \alpha_1 B_{1;r^{(1)},r^{(2)}} + \dots + \alpha_n B_{n;r^{(1)},r^{(2)}} + \dots + \alpha_n B_{n;r^{(1)},r^{(2)}} + \dots + \alpha_k B_{k;r^{(1)},r^{(2)}} \right|_{p_{r^{(1)},r^{(2)}}} \quad (4)$$

где

$B_{i;r^{(1)},r^{(2)}}$ – ортогональный базис;

$i, r^{(1)}, r^{(2)} = 1, 2, \dots, n, \dots, k ; i \neq r^{(1)} \neq r^{(2)} ;$

$$B_{i;r^{(1)},r^{(2)}} = \frac{P_{r^{(1)},r^{(2)}} \mu_{i;r^{(1)},r^{(2)}}}{p_i} ;$$

$$P_{r^{(1)},r^{(2)}} = \frac{P_k}{p_{r^{(1)}} p_{r^{(2)}}} ;$$

$\mu_{i;r^{(1)},r^{(2)}} \in N$ (вес ортогонального базиса), который находится из сравнения:

$$\frac{P_{r^{(1)},r^{(2)}} \mu_{i;r^{(1)},r^{(2)}}}{p_i} \equiv 1 \pmod{p_i}$$

Для реализации (4) предварительно составим таблицу 3, содержащую значения пересчитанных ортогональных базисов и модулей системы (включающей пять оснований) для условия возникновения двукратной ошибки по двум различным основаниям системы.

Так как кратность ошибки равна двум, а общее количество возможных ошибок равно количеству модулей системы (по условию равно пяти), то для вычисления количества возможных сочетаний ошибок необходимо использовать биномиальные коэффициенты:

$$C_\eta^\kappa = \frac{\eta!}{\kappa!(\eta-\kappa)!}, \text{ где } \eta - \text{общее количество воз-}$$

можных ошибок, κ – кратность ошибки. Для рассматриваемого случая получим $C_5^2 = 10$.

В разработанном методе подблоки $m'_{1,j}, m'_{2,j}, \dots, m'_{n,j}, m'_{n+1,j}, \dots, m'_{k,j}$ интерпретируются как избыточный МК, который позволяет обнаруживать возникающую ошибку на любой стадии их обработки (при условии, что кратность гарантированно обнаруживаемой ошибки $t_{\text{обн}} = d_{\min} - 1$, где $d_{\min}$ – минимальное кодовое расстояние).

Восстановление блоков данных M'_j в случае нарушения их целостности возможно путем исключения из процесса восстановления любых r подблоков без ущерба для однозначности их представления (где $r = k - n$ – количество дополнительных подблоков), вследствие чего система подблоков блоков данных M'_j интерпретируется как несистематический или неразделимый код (где каждая разрядная цифра несет часть информации о числе, включая и избыточные символы, а также любые из r разрядных цифр можно считать избыточными символами), после чего вычисляется подблок $m'_{i,j}$ взамен ранее исключенного подблока $\tilde{m}'_{i,j}$ с обнаруженной ошибкой.

Таблица 3

Таблица значений ортогональных базисов и модулей системы

i		B_1				$P_{r^{(1)}, r^{(2)}}$
$r^{(1)}$	$r^{(2)}$	B_2	B_3	B_4	B_5	
1	2	0	$\frac{P_{1,2}\mu_{3;1,2}}{p_3}$	$\frac{P_{1,2}\mu_{4;1,2}}{p_4}$	$\frac{P_{1,2}\mu_{5;1,2}}{p_5}$	$p_3 p_4 p_5$
	3	$\frac{P_{1,3}\mu_{2;1,3}}{p_2}$	0	$\frac{P_{1,3}\mu_{4;1,3}}{p_4}$	$\frac{P_{1,3}\mu_{5;1,3}}{p_5}$	$p_2 p_4 p_5$
	4	$\frac{P_{1,4}\mu_{2;1,4}}{p_2}$	$\frac{P_{1,4}\mu_{3;1,4}}{p_3}$	0	$\frac{P_{1,4}\mu_{5;1,4}}{p_5}$	$p_3 p_3 p_5$
	5	$\frac{P_{1,5}\mu_{2;1,5}}{p_2}$	$\frac{P_{1,5}\mu_{3;1,5}}{p_3}$	$\frac{P_{1,5}\mu_{4;1,5}}{p_4}$	0	$p_2 p_3 p_4$
i		B_2				$P_{r^{(1)}, r^{(2)}}$
$r^{(1)}$	$r^{(2)}$	B_1	B_3	B_4	B_5	
2	3	$\frac{P_{2,3}\mu_{1;2,3}}{p_1}$	0	$\frac{P_{2,3}\mu_{4;2,3}}{p_4}$	$\frac{P_{2,3}\mu_{5;2,3}}{p_5}$	$p_1 p_4 p_5$
	4	$\frac{P_{2,4}\mu_{1;2,4}}{p_1}$	$\frac{P_{2,4}\mu_{3;2,4}}{p_3}$	0	$\frac{P_{2,4}\mu_{5;2,4}}{p_5}$	$p_1 p_3 p_5$
	5	$\frac{P_{2,5}\mu_{1;2,5}}{p_1}$	$\frac{P_{2,5}\mu_{3;2,5}}{p_3}$	$\frac{P_{2,5}\mu_{4;2,5}}{p_4}$	0	$p_1 p_3 p_4$
i		B_3				$P_{r^{(1)}, r^{(2)}}$
$r^{(1)}$	$r^{(2)}$	B_1	B_2	B_4	B_5	
3	4	$\frac{P_{3,4}\mu_{1;3,4}}{p_1}$	$\frac{P_{3,4}\mu_{2;3,4}}{p_2}$	0	$\frac{P_{3,4}\mu_{5;3,4}}{p_5}$	$p_1 p_2 p_5$
	5	$\frac{P_{3,5}\mu_{1;3,5}}{p_1}$	$\frac{P_{3,5}\mu_{2;3,5}}{p_2}$	$\frac{P_{3,5}\mu_{4;3,5}}{p_4}$	0	$p_1 p_2 p_4$
i		B_4				$P_{r^{(1)}, r^{(2)}}$
$r^{(1)}$	$r^{(1)}$	B_1	B_2	B_3	B_5	
4	5	$\frac{P_{4,5}\mu_{1;4,5}}{p_1}$	$\frac{P_{4,5}\mu_{2;4,5}}{p_2}$	$\frac{P_{4,5}\mu_{3;4,5}}{p_3}$	0	$p_1 p_2 p_3$

Таким образом, целостность блока данных M_i была обеспечена путем осуществления контроля и восстановления подблока $\tilde{m}'_{i,j}$ блока данных с нарушенной целостностью. Выполнение проверки достоверности восстановленных данных в случае нарушения их целостности осуществляется путем сравнения значения предварительно вычисленного эталонного хэш-кода S_i (или S'_i) со значением вычисленного хэш-кода S'_i уже от восстановленного блока данных M'_i .

Схема, поясняющая порядок осуществления контроля и обеспечения целостности информации (основной этап разработанного метода), представлена на рисунке 8.

Общая схема разработанного метода построения современных отказоустойчивых систем хранения данных на основе методов из области безопасности информации и методов теории надежности представлена на рисунке 9.

Рис. 8. Схема, поясняющая основной этап разработанного метода

Рис. 9. Общая схема реализации разработанного метода

Оценка разработанного метода

Оценка разработанного метода выполняется в сравнении с наиболее популярными из существующих решений комплексного обеспечения целостности данных, в котором последовательно применяются сначала криптографические преобразования к данным для контроля их целостности, а затем – технология резервного копирования для восстановления данных в случае нарушения их целостности.

При проектировании и разработке новых СХД ИАС возникает задача нахождения оптимального решения при контроле и обеспечении целостности хранящейся в них информации. К показателям качества оптимальности СХД могут быть отнесены: безопасность (надежность) хранения, объем вводимой избыточности и прочее. Разработанный метод при рассмотрении вопросов обеспечения целостности (восстановления) информации в случае ее нарушения является, по сути, инструментом решения задачи обеспечения безопасности (надежности) СХД ИАС. При этом результативность рассматриваемого решения может быть охарактеризована свойством отказоустойчивости. Тогда в качестве показателя отказоустойчивости определим вероятность безотказной работы в течение времени t (показатель $P(t)$) [31]. Критерий качества: $P(t) \rightarrow \max$.

Вероятность безотказной работы $P_1(t)$ СХД ИАС, построенных на основе разработанного метода, с функцией восстановления информации по принципу функционирования – скользящее резервирование, соответствует выражению:

$$P_1(t) = \sum_{i=0}^q \frac{(k-q) \left(\frac{\lambda}{k}\right)^i}{i!} \cdot \exp\left(-\left(k-q\right)\frac{\lambda}{k}t\right), \quad (5)$$

где k – общее число элементов схемы, q – число резервных элементов схемы, λ – интенсивность отказов.

В свою очередь, вероятность безотказной работы $P_2(t)$ СХД ИАС, построенной на основе существующего решения (технологии резервного копирования), соответствует выражению:

$$P_2(t) = \sum_{i=0}^d \frac{(\lambda t)^i}{i!} \cdot \exp(-\lambda t), \quad (6)$$

где d – кратность резервирования.

Расчет производится при продолжительности эксплуатации 87600 часов, интенсивность отказов: $\lambda = 0,00001 \text{ час}^{-1}$. В качестве исходных данных для

расчета вероятности безотказной работы рассмотрим избыточные МК, следующих структур:

- при изменении количества информационных оснований избыточного МК: $k = \{5, 7, 11\}$, при $q = 2 - \text{const}$;
- при изменении количества контрольных оснований избыточного МК: $k = 7 - \text{const}$, при $q = \{3, 4, 5\}$.

На основе (5) и (6) получим результаты, приведенные в таблице 4.

Таблица 4

Расчетные данные вероятности безотказной работы СХД ИАС, построенных на основе разработанного метода и существующего решения

t, час	$P_1(t)$, при $q = 2 - \text{const}$			$P_1(t)$, при $k = 7 - \text{const}$			$P_2(t)$
	$k = 5$	$k = 7$	$k = 11$	$q = 3$	$q = 4$	$q = 5$	$d = 2$
8400	0,999979	0,999966	0,999949	0,999982	0,999992	0,999998	0,996664
15600	0,999873	0,999788	0,999685	0,99989	0,999953	0,999986	0,989027
22800	0,999615	0,999363	0,999059	0,999666	0,999856	0,999956	0,977641
30000	0,99915	0,998602	0,997948	0,999261	0,999678	0,999902	0,963064
37200	0,998431	0,997435	0,996253	0,998634	0,999401	0,999815	0,945794
44400	0,997416	0,995801	0,9939	0,997747	0,999004	0,999651	0,926276
51600	0,996072	0,993654	0,990831	0,99657	0,998472	0,999522	0,904906
58800	0,99437	0,990958	0,987006	0,995076	0,99779	0,999303	0,882034
63600	0,993025	0,988842	0,984022	0,993893	0,997245	0,999127	0,866108
73200	0,989805	0,98382	0,979349	0,991057	0,995926	0,998696	0,832999
80400	0,986912	0,979342	0,976993	0,988502	0,994723	0,998298	0,807354
87400	0,983597	0,994269	0,963792	0,985569	0,993327	0,997832	0,781252

Оценка полученных результатов для расчета вероятности безотказной работы СХД ИАС, построенных на основе разработанного метода, представлена на графиках (рис. 10).

Рис. 10. Зависимость вероятности безотказной работы от структуры применяемого избыточного МК:
а) при $q = 2$ – const б) при $k = 7$ – const

Из графического представления полученных результатов видно, что наиболее выигрышным является применение избыточного МК со структурой: $k = 7$, $q = 5$, а менее выигрышным – при $k = 11$, $q = 2$.

Сравнительная оценка вероятностей безотказной работы СХД ИАС, построенных на основе разработанного метода (при $k = 11$, $q = 2$) и существующего решения (при $d = 2$) представлена на графике (рис. 11).

Рис. 11. Зависимость вероятности безотказной работы СХД ИАС, построенных на основе разработанного метода ($P_1(t)$) и существующего решения ($P_2(t)$)

Из рис. 11 видно, что вероятность безотказной работы СХД ИАС, построенной на основе разработанного метода (даже при использовании избыточного МК с наименее выигрышной структурой: $k = 11$, $q = 2$), выше по отношению к СХД ИАС, основанных на существующем решении, в частности, на следующих точках:

- при суммарной продолжительности эксплуатации равной 58800 часов выигрыш составляет $\approx 11,9\%$;
- при суммарной продолжительности эксплуатации равной 80400 часов выигрыш составляет $\approx 21\%$;
- при суммарной продолжительности эксплуатации равной 87600 часов выигрыш составляет $\approx 23,4\%$.

К тому же, использование избыточного МК с двумя контрольными основаниями ($q = 2$) позволит восстановить данные в случае нарушения их целостности не менее чем в 2-х подблоках блока данных. При этом избыточность сократится со 100% (при технологии резервного копирования) до 30-40% (при использовании разработанного метода).

Разработанный метод, как и существующие решения, основан на применении криптографических методов. Однако в отличие от известных, его использование в СХД ИАС с многомерной моделью данных, позволяет осуществить не только контроль, но и восстановление данных с криптографической проверкой их достоверности.

Выводы

Представленный гибридный метод контроля и восстановления целостности данных для защищённых ИАС, учитывающий структуру многомерного представления данных в современных СХД, основан на совместном применении хэш-функции и избыточных кодов.

Агрегирование известных классических решений для обеспечения целостности данных позволило расширить функциональные возможности СХД ИАС по обеспечению проверки достоверности данных после их восстановления.

Полученные результаты дают научный и инженерный инструментарий для контроля и обеспечения целостности данных с сокращением объема вводимой избыточности при повышении вероятности безотказной работы СХД ИАС.

Литература

1. Тарасов С.В. СУБД для программиста. Базы данных изнутри. М.: СОЛОН-Пресс, 2015. 320 с.
2. Murray D.R., Newman M.B. Probability analyses of combining background concentrations with model-predicted concentrations. // J. Air Waste Manag. Assoc. 2016, vol. 64, no. 3, pp. 248-254.
3. Соколов Н.П. Введение в теорию многомерных матриц. Издательство: «Просвещение», 2012. 175 с.
4. Диченко С.А. Концептуальная модель обеспечения целостности информации в современных системах хранения данных // Информатика: проблемы, методология, технологии. Сборник материалов XIX международной научно-методической конференции. Под ред. Д.Н. Борисова. Воронеж, 2019. С. 697-701.
5. Самойленко Д.В., Финько О.А. Помехоустойчивая передача данных в радиоканалах робототехнических комплексов на основе полиномиальных классов вычетов // Научные технологии в космических исследованиях Земли. 2016. Т. 8. № 3. С. 49-55.
6. Schneier B. Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons, Inc. 2016. 653 p.
7. Menezes A.J., Oorschot P., Vanstone S. Handbook of Applied Cryptography. CRC Press, Inc. 2015. 770 p.
8. Knuth D.E. The Art of Computer Programming: Volume 3: Sorting and Searching. 2nd edn. 2018. 803 p.
9. Biham E., Dunkelman O. A framework for iterative hash functions. – HAIFA. ePrint Archive, Report 2007/278. 2007, pp.1-20.
10. Wang X., Yu H. How to break MD5 and Other Hash Function. EUROCRYPT. LNCS 3494. Springer-Verlag. 2005, pp. 19-35.
11. Bellare M. New Proofs for NMAC and HMAC: Security without Collision-Resistance. CRYPTO. ePrint Archive, Report 2006/043. 2006, pp.1-16.
12. Самойленко Д.В. Повышение информационной живучести группировки робототехнических комплексов в условиях деструктивных воздействий злоумышленника // Автоматизация процессов управления. 2018. № 2(52). С. 4-13.
13. Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, David A. Patterson. RAID: High-Performance, Reliable Secondary Storage. ACM Computing Surveys, 26, 2014. pp. 14-35.
14. Саложников В.В. Основы теории надежности и технической диагностики. СПб.: Лань, 2019. 588 с.
15. Хетагуров Я.А. Повышение надежности цифровых устройств методами избыточного кодирования. 3-е изд. М.: Энергия, 2015. 376 с.
16. Зубарев Ю.М. Основы надежности машин и сложных систем. СПб.: Лань, 2017. 180 с.
17. Morelos-Zaragoza R.H. The Art of Error Correcting Coding. 2nd edn John Wiley & Sons, Inc. 2006. 263 p.
18. Hamming R. Coding and Information Theory. Prentice-Hall. 3-е изд. 2008. 259 p.
19. Sieck J., Yatskiv V., Sachenko A., Tsavolyk T. Two-dimensional Error Control Based on Modular Correcting Codes // International Journal of Computing, 14(4), 2015. pp. 208-215.
20. Finko O.A., Dichenko S.A. Two-dimensional control and assurance of data integrity in information systems based on residue number system codes and cryptographic hash functions // Proceedings of the 2018 Multidisciplinary Symposium on Computer Science and ICT (Stavropol, Russia, October 15, 2018), 2254, CEUR Workshop Proceedings, 2018. 8 p.
21. Устройство имитостойкого кодирования и декодирования информации избыточными систематическими кодами: пат. 2634201 Рос. Федерация. № 2016130881/08 / Самойленко Д.В., Финько О.А., Диченко С.А. и др.; заявл. 26.07.16; опубл. 24.10.2017, Бюл. № 30. 9 с.

22. Способ и система распределенного хранения восстанавливаемых данных с обеспечением целостности и конфиденциальности информации: положительное решение о выдаче пат. Рос. Федерация. № 2017115539 / Самойленко Д.В., Финько О.А., Еремеев М.А., Диченко С.А.; заявл. 02.05.17. 27 с.
23. Способ обеспечения целостности данных: положительное решение о выдаче пат. Рос. Федерация. № 2017117714 / Диченко С.А., Финько О.А., Самойленко Д.В.; заявл. 22.05.17. 11 с.
24. Способ контроля и обеспечения целостности данных: положительное решение о выдаче пат. Рос. Федерация. № 2017141538 / Диченко С.А., Финько О.А., Самойленко Д.В.; заявл. 28.11.17. 14 с.
25. Диченко С.А. Контроль и обеспечение целостности информации в системах хранения данных // Научные технологии в космических исследованиях Земли. 2019. Т. 11. № 1. С. 49-57.
26. Tchernykh A., Babenko M., Chervyakov N. and etc. AC-RRNS: Anti-Collusion Secured Data Sharing Scheme for Cloud Storage // International Journal of Approximate Reasoning. Special Issue on Uncertainty in Cloud Computing: Concepts, Challenges and Current Solutions. Vol. 102, pp. 60-73.
27. Tilborg H. Fundamentals of cryptology. A professional reference and interactive tutorial. Kluwer Academic Publishers 2014. 414 p.
28. Yang L.-L., Hanzo L. Coding Theory and Performance Of Redundant Residue Number System Codes, Vehicular Technology Conference Fall. IEEE VTS 54th. 2001. pp. 1472-1476.
29. Samoylenko D.V., Eremeev M.A., Finko O.A. A method of providing the integrity of information in the group of robotic engineering complexes based on crypt-code constructions // Automatic Control and Computer Sciences, 51:8. 2017, pp. 965-971.
30. Бухштаб А.А. Теория чисел. М.: Просвещение, 2016. 384 с.
31. Диченко С.А. Разработка алгоритма контроля и обеспечения целостности данных при их хранении в центрах обработки данных // Информационный бюллетень Омского научно-образовательного центра ОмГТУ и ИМ СО РАН в области математики и информатики Материалы VIII Международной молодежной научно-практической конференции с элементами научной школы. 2018. С. 110-113.

HYBRID CRYPTO-CODE METHOD FOR MONITORING AND RECOVERY OF DATA INTEGRITY FOR PROTECTED INFORMATION AND ANALYTICAL SYSTEMS

Dichenko S.A.⁷, Finko O.A.⁸

The purpose of the article is to improve fault tolerance of data storage systems of protected information and analytical systems and simultaneously reduce introduced redundancy. The developed method is original due to consideration of peculiarities of protected information and analytical systems operation under conditions of destructive effects caused by an intruder's actions and the environment. Hashing is used to monitor the data integrity while data recovery is performed based on application of redundant modular codes. Simultaneous use of known solutions takes into account the structure of multidimensional data presentation in modern protected information and analytical systems and allows checking the validity of recovered data. The article describes the results of comparative evaluation of the used and suggested methods for data integrity monitoring and recovery. The article is useful because it describes original schemes that take into account the structure of multidimensional data presentation in modern protected information and analytical systems.

Keywords: *information security, protected information-analytical systems, data storage systems, data integrity control, data recovery, cryptographic primitives, hash function, hash-code, redundant residue number system codes.*

References

1. Tarasov S.V. DBMS for the programmer. Databases inside. M.: SOLON-Press, 2015. 320 p.
2. Murray D.R., Newman M.B. Probability analyses of combining background concentrations with model-predicted concentrations. J. Air Waste Manag. Assoc. 2016, vol. 64, no. 3, pp. 248-254.
3. Sokolov N.P. Introduction to the theory of multidimensional matrices. Publisher: "Prosveshcheniye", 2012. 175 p.

7 Sergey Dichenko, Ph.D., University Teachers, Krasnodar Higher Military School, Krasnodar, Russia. E-mail: dichenko.sa@yandex.ru

8 Oleg Finko, Dr.Sc., Professor, Academic Advisor of the Russian Academy of Rocket and Artillery Sciences (RARAN), Krasnodar Higher Military School, Krasnodar, Russia. E-mail: ofinko@yandex.ru. Web: <http://www.mathnet.ru/person40004>. ORCID 0000-0002-7376-2714.

4. Dichenko S.A. A conceptual model for ensuring the integrity of information in modern data storage systems // Informatics: problems, methodology, technologies. The collection of materials of the XIX international scientific and methodological conference. Ed. D.N. Borisov. Voronezh, 2019. pp. 697-701.
5. Samoilenko D.V., Finko O.A. Noise-resistant data transmission in the radio channels of robotic systems based on polynomial residue classes // High-tech in space exploration of the Earth. 2016. V.8. N3. pp 49-55.
6. Schneier B. Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons, Inc. 2016. 653 p.
7. Menezes A.J., Oorschot P., Vanstone S. Handbook of Applied Cryptography. CRC Press, Inc. 2015. 770 p.
8. Knuth D.E. The Art of Computer Programming: Vol. 3: Sorting and Searching. 2nd edn. 2018. 803 p.
9. Biham E., Dunkelman O. A framework for iterative hash functions. – HAIFA. ePrint Archive, Report 2007/278. 2007, pp.1-20.
10. Wang X., Yu H. How to break MD5 and Other Hash Function. EUROCRYPT. LNCS 3494. Springer-Verlag. 2005, pp. 19-35.
11. Bellare M. New Proofs for NMAC and HMAC: Security without Collision-Resistance. CRYPTO. ePrint Archive, Report 2006/043. 2006, pp.1-16.
12. Samoilenko D.V. Improving the information survivability of a grouping of robotic complexes under the conditions of the destructive effects of an attacker // Automation of control processes. 2018. N2 (52). pp.4-13.
13. Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, David A. Patterson. RAID: High-Performance, Reliable Secondary Storage. ACM Computing Surveys, 26, 2014. pp. 14-35.
14. Sapozhnikov V.V. Fundamentals of the theory of reliability and technical diagnostics. SPb.: Doe, 2019. 588 p.
15. Khetagurov Ya.A. Improving the reliability of digital devices using redundant coding methods. - 3rd ed. M.: Energy, 2015. 376 p.
16. Zubarev Yu.M. The basics of machine reliability and complex systems. SPb.: Doe, 2017. 180 p.
17. Morelos-Zaragoza R.H. The Art of Error Correcting Coding. 2nd edn John Wiley & Sons, Inc. 2006. 263 p.
18. Hamming R. Coding and Information Theory. Prentice-Hall. - 3rd ed. 2008. 259 p.
19. Sieck J., Yatskiv V., Sachenko A., Tsavolyk T. Two-dimensional Error Control Based on Modular Correcting Codes // International Journal of Computing, 14(4), 2015. pp. 208-215.
20. Finko O.A., Dichenko S.A. Two-dimensional control and assurance of data integrity in information systems based on residue number system codes and cryptographic hash functions // Proceedings of the 2018 Multidisciplinary Symposium on Computer Science and ICT (Stavropol, Russia, October 15, 2018), 2254, CEUR Workshop Proceedings, 2018. 8 p.
21. The device is imitative of coding and decoding information redundant systematic codes: pat. 2634201 Ros. Federation. N2016130881/08 / Samoylenko D.V., Finko O.A., Dichenko S.A. and etc.; declare 26.07.16; publ. 24.10.2017, Bull. N30. 9 p.
22. A way and system for distributed storage of recoverable data while ensuring integrity and confidentiality of information: positive decision on the issuance of the patent. Ros. Federation. N2017115539 / Samoylenko D.V., Finko O.A., Ereemeev M.A., Dichenko S.A.; declare 02.05.17. 27 p.
23. A way to ensure data integrity: positive decision on the issuance of the patent. Ros. Federation. N2017117714 / Dichenko S.A., Finko O.A., Samoylenko D.V.; declare 22.05.17. 11 p.
24. The way to control and ensure data integrity: positive decision on the issuance of the patent. Ros. Federation. N2017141538 / Dichenko S.A., Finko O.A., Samoylenko D.V.; declare 28.11.17. 14 p.
25. Dichenko S.A. Control and security of information integrity in data storage systems // High technology in space exploration of the Earth. 2019. Vol. 11. N 1. pp. 49-57.
26. Tchernykh A., Babenko M., Chervyakov N. and etc. AC-RRNS: Anti-Collusion Secured Data Sharing Scheme for Cloud Storage // International Journal of Approximate Reasoning. Special Issue on Uncertainty in Cloud Computing: Concepts, Challenges and Current Solutions. Vol. 102, pp. 60-73.
27. Tilborg H. Fundamentals of cryptology. A professional reference and interactive tutorial. Kluwer Academic Publishers 2014. 414 p.
28. Yang L.-L., Hanzo L. Coding Theory and Performance Of Redundant Residue Number System Codes, Vehicular Technology Conference Fall. IEEE VTS 54th. 2001. pp. 1472-1476.
29. Samoylenko D.V., Ereemeev M.A., Finko O.A. A method of providing the integrity of information in the group of robotic engineering complexes based on crypt-code constructions // Automatic Control and Computer Sciences, 51:8. 2017, pp. 965-971.
30. Buchstab A.A. Theory of numbers. M.: Education, 2016. 384 p.
31. Dichenko S.A. Development of an algorithm for monitoring and ensuring the integrity of data during their storage in data processing centers // Newsletter of the Omsk Scientific and Educational Center of OmSTU and IM SB RAS in the field of mathematics and computer science Materials of the VIII International Youth Scientific and Practical Conference with elements of a scientific school. 2018. pp. 110-113.