ПРОГРАММИРОВАНИЕ, 2021, № 6, с. 3–15

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 519.718

КОНТРОЛЬ И ВОССТАНОВЛЕНИЕ ЦЕЛОСТНОСТИ МНОГОМЕРНЫХ МАССИВОВ ДАННЫХ ПОСРЕДСТВОМ КРИПТОКОДОВЫХ КОНСТРУКЦИЙ

© 2021 г. С. А. Диченко[a],*, О. А. Финько[a],**

[a] *Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С. М. Штеменко 350063 Краснодар, ул. Красина, д. 4, Россия*

**E-mail: dichenko.sa@yandex.ru*

***E-mail: ofinko@yandex.ru*

Поступила в редакцию 09.03.2021 г.
После доработки 31.05.2021 г.
Принята к публикации 23.06.2021 г.

Разработан метод контроля и восстановления целостности данных в многомерных системах хранения. Предложенные конструкции контроля и восстановления целостности многомерных массивов данных основаны на агрегировании методов криптографии и помехоустойчивого кодирования. На основе представленных криптокодовых конструкций показана особенность комплексирования существующих методов, заключающаяся в повышении вероятности обеспечения целостности информации в условиях разрушающих воздействий злоумышленника и среды для многомерных систем хранения данных. Получены расчетные данные вероятности обнаружения и исправления возникающих в многомерных массивах данных ошибок, приводящих к нарушению их целостности, посредством разработанного метода.

DOI: 10.31857/S0132347421060042

1. ВВЕДЕНИЕ

Эффективность любой информационной системы (ИС) независимо от условий ее функционирования, в первую очередь, характеризуется степенью достижения целей, поставленных при ее создании [1, 2]. Поэтому при разрушающих воздействиях злоумышленника и среды для ИС, обеспечивающих своевременность и правильность принимаемого пользователем системы решения, необходимым условием достижения требуемого качества их функционирования является обеспечение безопасности обрабатываемой информации.

Безопасность информации определяется состоянием защищенности ИС, а также систем хранения данных (СХД), применяемых в их интересах, от различных угроз и в итоге – способностью ИС обеспечить конкретному пользователю доступность, целостность и конфиденциальность требуемой информации в системе [3].

Для обеспечения целостности информации используются методы из области теоретических положений защиты информации и теории надежности, при которых для обнаружения и исправления ошибок, приводящих к искажению (или уничтожению) информации, требуется введение избыточности. При этом объем вводимой избыточности определяется, к примеру, необходимостью хранения эталонных хэш-кодов, используемых при контроле целостности данных посредством применения хэш-функции, и резервных копий данных для возможности восстановления их целостности на основе технологии резервного копирования.

Вместе с тем, в современных условиях постоянного роста объема и ценности непрерывно накапливаемой в СХД информации, введение требуемой избыточности и, как следствие, увеличение длительности загрузки новых данных в СХД, перерасход используемой памяти и снижение производительности системы в целом, приводит к необходимости разработки новых методов обеспечения целостности информации.

Формируемые в таких условиях требования к качеству функционирования ИС должны быть направлены на достижение цели применения системы за счет повышения уровня безопасности информации без увеличения при этом вводимой избыточности.

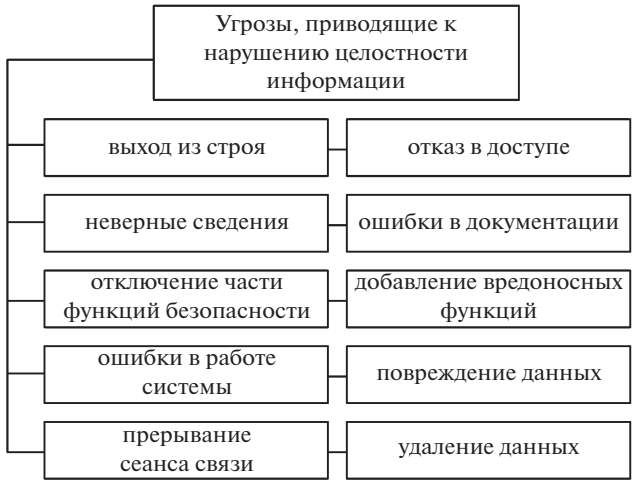

Рис. 1. Классификация угроз безопасности информации, приводящих к нарушению целостности информации.

2. УГРОЗЫ БЕЗОПАСНОСТИ ИНФОРМАЦИИ И ПОСЛЕДСТВИЯ ИХ РЕАЛИЗАЦИИ

Классификация угроз безопасности информации по признаку “результат реализации”, приводящие к нарушению ее целостности, представлена на рис. 1.

В то же время, целостность данных, хранящихся в СХД, чаще всего нарушается в результате случайных ошибок и преднамеренного несанкционированного изменения данных (например, посредством действия вредоносного кода) или выхода из строя части носителя (например, отдельных ячеек, секторов).

До настоящего времени задачу обеспечения целостности данных в СХД принято было решать путем повышения надежности средств хранения. Для этих целей широкое развитие получили системы резервирования и архивирования данных на дополнительные носители, построение каналов для их тиражирования и т.д.

3. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ ЗАЩИТЫ ЦЕЛОСТНОСТИ ДАННЫХ

Известны методы из области теоретических положений защиты информации, позволяющие осуществить контроль целостности данных за счет вычисления значений контрольных сумм (CRC-кодов) и их сравнения со значениями эталонных [4, 5] (рис. 2).

Достоинством данных методов является простота реализации, недостатком – отсутствие обеспечения криптографической стойкости выполняемых при контроле целостности данных преобразований.

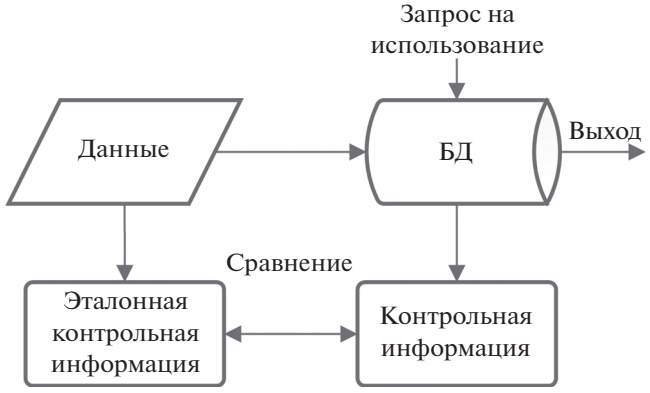

Рис. 2. Пояснение принципа контроля целостности данных на основе применения алгоритма вычисления контрольных сумм (CRC-кодов).

Известны решения, основанные на применении криптографических методов: ключевое и бесключевое хэширование, средства электронной подписи [6–9] (рис. 3).

Однако контроль целостности данных, разбитых на блоки небольшой размерности, где от каждого блока данных, подлежащего защите, вычисляется хэш-код, характеризуется вводом высокой избыточности контрольной информации. При попытке устранения данного недостатка за счет вычисления одного хэш-кода от нескольких блоков данных приводит к отсутствию возможности локализации блока данных с признаками нарушения целостности (рис. 4).

Наиболее популярные из существующих методов теории надежности, позволяющие осуществить контроль и восстановление целостности данных, основаны на применении различных видов резервирования (рис. 5), в том числе, с использованием программно-аппаратной или программной реализации технологии RAID (Redundant Array of Independent Disks) (RAID-массивы) [10] (рис. 6), методы дублирования и избыточного кодирования [11–15] (рис. 7).

Однако они предназначены для борьбы со случайными ошибками, поэтому в условиях разрушающих воздействий злоумышленника они не всегда являются эффективными. Данное утверждение складывается из того, что современная теория надежности, помимо множества других разделов математики, в основном, базируется на математическом аппарате теории вероятностей. При этом для повышения вероятности обнаружения и исправления возникающих ошибок, приводящих к нарушению целостности данных, требуется увеличение количества вводимой избыточности. К примеру, избыточность, вводимая для восстановления целостности данных посредством применения технологии резервного копи-

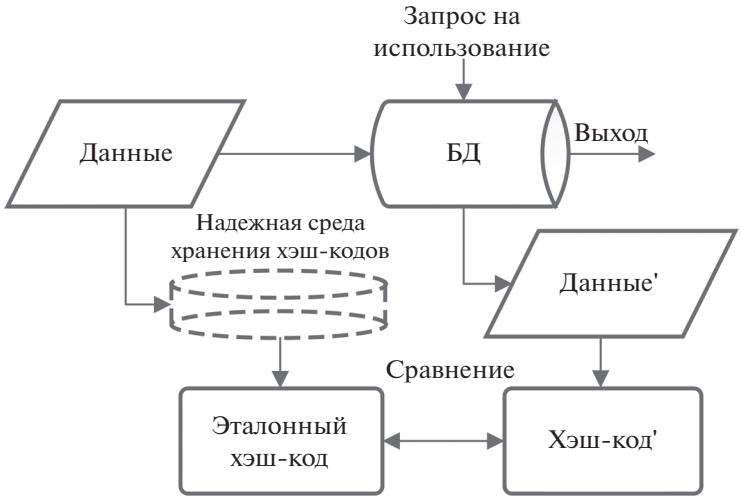

Рис. 3. Пояснение принципа контроля целостности данных на основе применения хэш-функции.

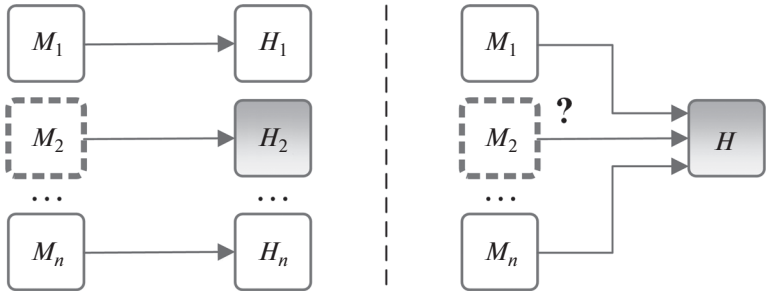

Рис. 4. Пояснение различий по выявлению блока данных M_2 при раздельном (схема слева) и общем (схема справа) применении криптографических методов (линией "---" обозначен блок данных с нарушением целостности).

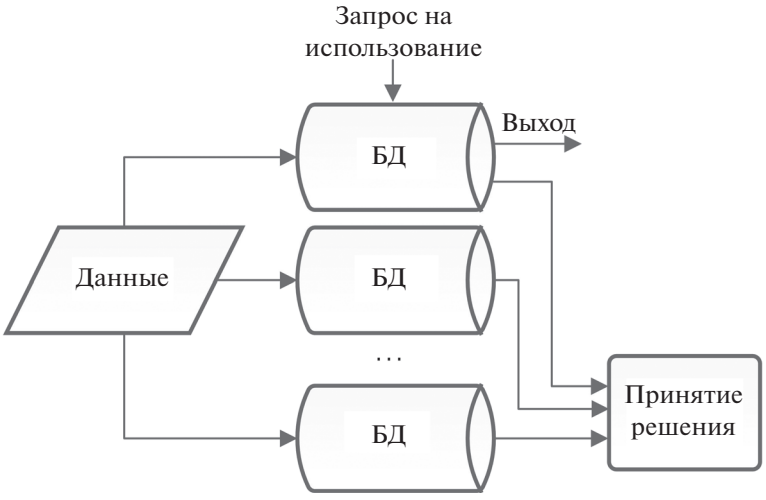

Рис. 5. Пояснение принципа резервирования БД.

Рис. 6. Пояснение принципа применения в БД технологии RAID (тип – 1).

рования, равна 100% от общего объема защищаемых данных (рис. 8).

Таким образом, ввод высокой избыточности в существующих методах объясняется потребностью в повышении обнаруживающей и исправляющей способностей при контроле и восстановлении целостности данных в условиях разрушающих воздействий.

4. МЕТОД КОНТРОЛЯ И ВОССТАНОВЛЕНИЯ ЦЕЛОСТНОСТИ МНОГОМЕРНЫХ МАССИВОВ ДАННЫХ

Предложенный метод предназначен для контроля и восстановления целостности данных, представленных в виде упорядоченных многомерных массивов. Он основан на совместном использовании методов криптографической защиты информации (функции хэширования) и теории кодов, контролирующих ошибки (избыточных модулярных полиномиальных кодов (МПК)), агрегирование которых позволяет повысить обнаруживающую и исправляющую способности при контроле и восстановлении целостности данных в условиях разрушающих воздействий без увеличения для этого вводимой избыточности.

Формализованное представление многомерных массивов данных. Блоки данных M, хранящиеся в СХД в виде упорядоченных многомерных массивов, по аналогии с теорией многомерных матриц [16] могут быть представлены p-мерным массивом данных, под которым будем понимать совокупность элементов $M_{\psi_1 \ldots \psi_p}$, где индексы $\psi_1 \ldots \psi_p$ принимают значения от 1 до k_a ($a = 1, \ldots, p$) соответственно.

При этом p-мерный массив данных содержит $k_1 \times k_2 \times \ldots \times k_p$ элементов и обозначается как

$$\mathbf{M}[k_1, k_2, \ldots, k_p] = \{M_{\psi_1 \ldots \psi_p}\}.$$

На примере 3-мерного куба данных (рис. 9), содержащего систему координат с осями: x, y, z, по которым откладываются блоки данных M_{ijr} ($i, j, r = 0, \ldots, k-1$), получим 3-мерный массив данных:

$$\mathbf{M}[k, k, k] = \{M_{ijr}\},$$

который может быть представлен в следующем виде:

$$\mathbf{M}[k,k,k] = \left\{ \begin{vmatrix} M_{000}, & \ldots, & M_{k-1,0,0} \\ \vdots & \ddots & \vdots \\ M_{0,k-1,0}, & \ldots, & M_{k-1,k-1,0} \end{vmatrix} \ldots \right.$$
$$\left. \ldots \begin{vmatrix} M_{0,0,k-1}, & \ldots, & M_{k-1,0,k-1} \\ \vdots & \ddots & \vdots \\ M_{0,k-1,k-1}, & \ldots, & M_{k-1,k-1,k-1} \end{vmatrix} \right\}^{\mapsto(z)} \begin{matrix} \to (x) \\ \downarrow \\ (y), \end{matrix}$$

где стрелка "$\mapsto$" с индексом (z) показывает порядок представления массива посредством сечений, ориентированных по оси z; стрелки "$\to$" и "$\downarrow$" с индексами (x) и (y) указывают направления, в которых возрастают соответствующие индексы у элементов массива по осям x и y.

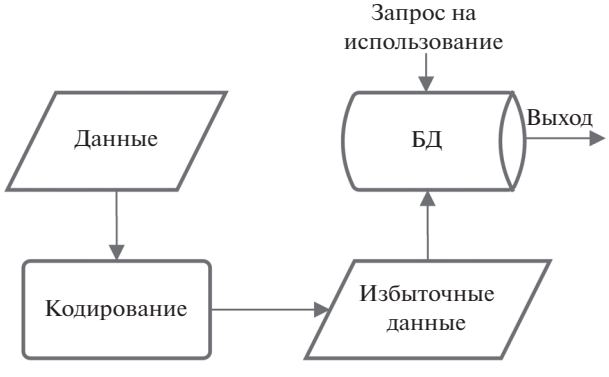

Рис. 7. Пояснение принципа применения метода избыточного кодирования в БД.

Построение модели контроля целостности многомерных массивов данных. Для осуществления контроля целостности полученного массива данных $\mathbf{M}[k, k, k]$ разместим его в массиве большего размера, который с помощью сечений ориентации (z) может быть представлен в следующем виде:

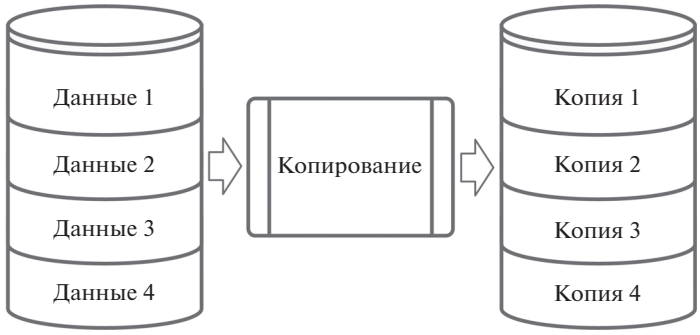

Рис. 8. Пояснение принципа применения технологии резервного копирования в БД.

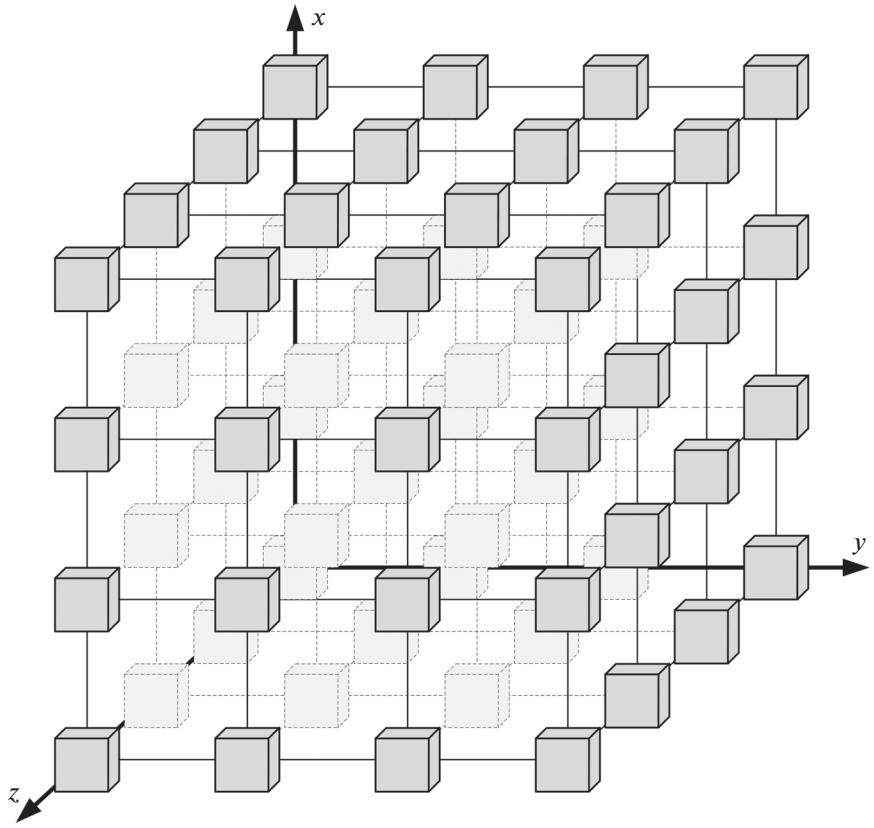

Рис. 9. Схема 3-мерного куба.

$$\mathbf{W}'[k+1,k+1,k+1] =$$
$$= \left\{ \begin{vmatrix} M_{000}, & \ldots, & M_{k-1,0,0}, & 0 \\ \vdots & \ddots & \vdots & \vdots \\ M_{0,k-1,0}, & \ldots, & M_{k-1,k-1,0}, & 0 \\ 0, & \ldots, & 0, & 0 \end{vmatrix} \ldots \right.$$
$$\left. \ldots \begin{vmatrix} M_{0,0,k-1}, & \ldots, & M_{k-1,0,k-1}, & 0 \\ \vdots & \ddots & \vdots & \vdots \\ M_{0,k-1,k-1}, & \ldots, & M_{k-1,k-1,k-1}, & 0 \\ 0, & \ldots, & 0, & 0 \end{vmatrix} \ldots \right\}^{\mapsto(z)}$$
$$\ldots \begin{vmatrix} 0, & 0, & \ldots, & 0 \\ 0, & 0, & \ldots, & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0, & 0, & 0, & 0 \end{vmatrix} \begin{matrix} \to (x) \\ \downarrow \\ (y), \end{matrix}$$

где “0” обозначает свободную для записи ячейку массива.

Контроль целостности 3-мерного массива данных $\mathbf{M}[k,k,k]$ осуществляется посредством применения к его элементам M_{ijr} хэш-функции [17].

Определение 4.1. Под хэш-функцией h понимается функция, отображающая строку бит $M \in \{0,1\}^*$ в строку бит $H \in \{0,1\}^q$:

$$h(M) = H,$$

где "$\{\cdot\}^*$" – произвольный размер строки бит, "$\{\cdot\}^q$" – фиксированный размер, $q \in N$.

Определение 4.2. Строка бит $M \in \{0,1\}^*$, которую хэш-функция h отображает в строку бит $H \in \{0,1\}^q$, называется блоком данных.

Определение 4.3. Под хэш-кодом понимается строка бит $H \in \{0,1\}^q$, являющаяся выходным результатом хэш-функции h.

На основе ранее разработанных схем хэширования [18–23] вычислим для блоков данных M_{ijr} $(i, j, r = 0, \ldots, k-1)$, подлежащих защите, хэш-коды H_{ijr} $(i, j, r = 0, \ldots, k)$.

В соответствии с правилами построения кубических кодов [15] разместим полученные хэш-коды в свободные для записи ячейки массива $\mathbf{W}'[k+1, k+1, k+1]$.

Получим массив, представленный с помощью сечений ориентации (z):

$$\mathbf{W}[k+1, k+1, k+1] =$$
$$= \begin{Vmatrix} M_{000}, & \ldots, & M_{k-1,0,0}, & H_{k00} \\ \vdots & \ddots & \vdots & \vdots \\ M_{0,k-1,0}, & \ldots, & M_{k-1,k-1,0}, & H_{k,k-1,0} \\ H_{0k0}, & \ldots, & H_{k-1,k,0}, & 0 \end{Vmatrix} \ldots$$
$$\ldots \begin{Vmatrix} M_{0,0,k-1}, & \ldots, & M_{k-1,0,k-1}, & H_{k,0,k-1} \\ \vdots & \ddots & \vdots & \vdots \\ M_{0,k-1,k-1}, & \ldots, & M_{k-1,k-1,k-1}, & H_{k,k-1,k-1} \\ H_{0,k,k-1}, & \ldots, & H_{k-1,k,k-1}, & 0 \end{Vmatrix} \ldots$$
$$\ldots \begin{Vmatrix} H_{00k}, & \ldots, & H_{k-1,0,k}, & 0 \\ \vdots & \ddots & \vdots & \vdots \\ H_{0,k-1,k}, & \ldots, & H_{k-1,k-1,k}, & 0 \\ 0, & \ldots, & 0, & 0 \end{Vmatrix}^{\mapsto(z)} \begin{matrix} \to (x) \\ \downarrow \\ (y). \end{matrix}$$

Хэш-коды H_{ijr}, вычисленные посредством применения к блокам данных M_{ijr} хэш-функции $h(M_{ijr})$ (или $h_k(M_{ijr})$, k – секретный ключ), будут являться эталонными.

Таким образом, полученный массив, содержащий $(k+1)^3$ ячеек, будет состоять из:

– k^3 блоков данных M_{ijr}, подлежащих защите целостности $(i, j, r = 0, \ldots, k-1)$;

– $3k^2$ блоков с вычисленными эталонными хэш-кодами H_{ijr} $(i, j, r = 0, \ldots, k)$;

– $3k+1$ оставшихся свободных ячеек.

В соответствии с полученной моделью к каждому блоку данных, подлежащему защите, по трем осям добавляются блоки с вычисленными от них эталонными хэш-кодами, используемыми для обнаружения данных с признаками нарушения целостности.

Определение 4.4. Под нарушением целостности одного блока данных будем понимать возникновение в нем ошибки, соответственно нарушение целостности q блоков данных определяется возникновением q-кратной ошибки.

Обнаружение блока данных с признаками нарушения целостности выполняется путем сравнения значений предварительно вычисленных от него эталонных хэш-кодов и хэш-кодов, вычисленных при запросе на его использование. В случае несоответствия сравниваемых значений хэш-кодов делается вывод о возникновении ошибки и определяется ее синдром.

Определение 4.5. Под синдромом ошибки будем понимать двоичное число, полученное при написании символа "0" для каждой выполненной проверки на соответствие значений вычисленного и эталонного хэш-кода и символа "1" при несоответствии сравниваемых значений.

Порядок контроля целостности многомерных массивов данных. В соответствии с разработанной моделью разместим в 3-мерном кубе (рис. 10) блоки данных, подлежащие защите, и соответствующие им хэш-коды.

При этом блоки данных $\mathbf{M}_{ijr}$, подлежащие защите, и вычисленные от них эталонные хэш-коды $\mathbf{H}_{ijr}$ интерпретируются как двоичные векторы:

$$\mathbf{M}_{ijr} = [\mu_1^{(ijr)}, \mu_2^{(ijr)}, \ldots, \mu_t^{(ijr)}];$$
$$\mathbf{H}_{ijr} = [\xi_1^{(ijr)}, \xi_2^{(ijr)}, \ldots, \xi_t^{(ijr)}],$$

где $\mu_g^{(ijr)}, \xi_g^{(ijr)} \in \{0,1\}$; $g = 1, 2, \ldots, t$.

Получим массив, представленный посредством сечений ориентации (x):

$$\mathbf{W}[3,3,3] = \begin{Vmatrix} \mathbf{M}_{000}, & \mathbf{M}_{010}, & \mathbf{H}_{020} \\ \mathbf{M}_{001}, & \mathbf{M}_{011}, & \mathbf{H}_{021} \\ \mathbf{H}_{002}, & \mathbf{H}_{012}, & 0 \end{Vmatrix}$$
$$\begin{Vmatrix} \mathbf{M}_{100}, & \mathbf{M}_{110}, & \mathbf{H}_{120} \\ \mathbf{M}_{101}, & \mathbf{M}_{111}, & \mathbf{H}_{121} \\ \mathbf{H}_{102}, & \mathbf{H}_{112}, & 0 \end{Vmatrix}$$
$$\begin{Vmatrix} \mathbf{H}_{200}, & \mathbf{H}_{210}, & 0 \\ \mathbf{H}_{101}, & \mathbf{H}_{111}, & 0 \\ 0, & 0, & 0, \end{Vmatrix}^{\mapsto(x)} \begin{matrix} \to (y) \\ \downarrow \\ (z), \end{matrix}$$

при этом каждый хэш-код вычисляется от двух блоков данных, расположенных с ним в одной строке или одном столбце массива.

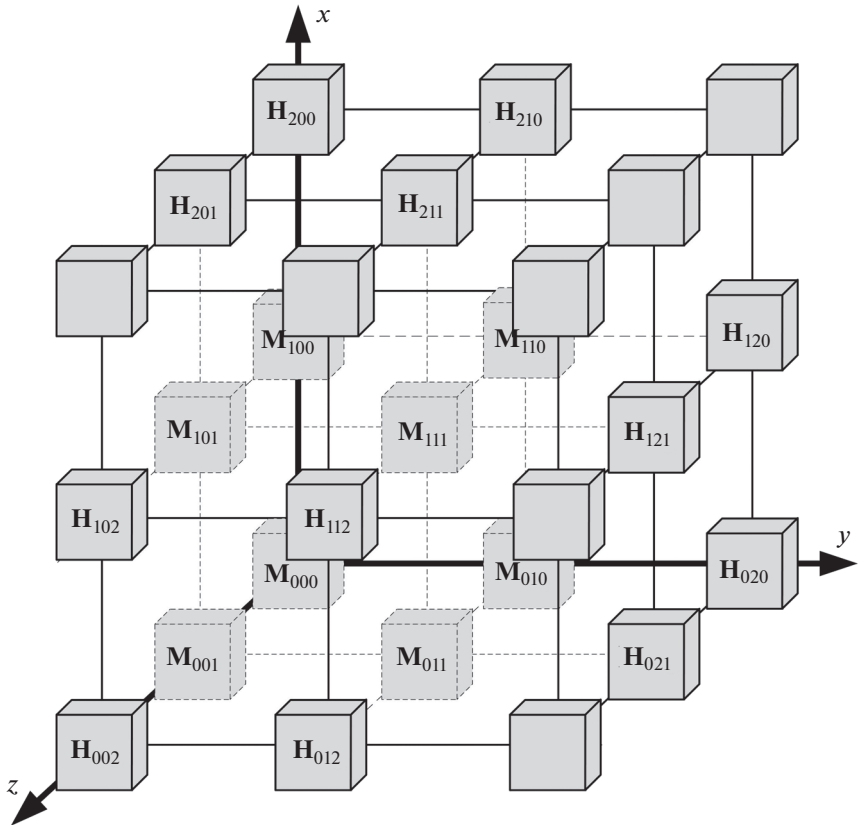

Рис. 10. Принцип организации 3-мерного куба (при $k = 2$).

Пример 4.1. Хэш-коды $\mathbf{H}_{002}$, $\mathbf{H}_{012}$, $\mathbf{H}_{020}$, $\mathbf{H}_{021}$ вычисляются следующим образом:

$$\mathbf{H}_{002} = h(\mathbf{M}_{000} \| \mathbf{M}_{001});$$
$$\mathbf{H}_{012} = h(\mathbf{M}_{010} \| \mathbf{M}_{011});$$
$$\mathbf{H}_{020} = h(\mathbf{M}_{000} \| \mathbf{M}_{010});$$
$$\mathbf{H}_{021} = h(\mathbf{M}_{001} \| \mathbf{M}_{011}),$$

где "$\|$" – операция конкатенации (объединения).

Построим сеть хэширования (рис. 11) для однократной ошибки, в которой каждому блоку данных $\mathbf{M}_{ijr}$ будет соответствовать неповторяющаяся совокупность из трех хэш-кодов $\mathbf{H}_{ijr}$.

Пример 4.2. Блокам данных $\mathbf{M}_{100}$, $\mathbf{M}_{011}$ соответствуют следующие неповторяющиеся совокупности из трех хэш-кодов:

– для $\mathbf{M}_{100}$: $\mathbf{H}_{120}, \mathbf{H}_{102}, \mathbf{H}_{200}$;

– для $\mathbf{M}_{011}$: $\mathbf{H}_{021}, \mathbf{H}_{012}, \mathbf{H}_{211}$.

Построим таблицу синдромов ошибок, приводящих к нарушению целостности блока данных $\mathbf{M}_{ijr}$, в которой место ошибки определяется наличием символова "1" в соответствующих столбцах и строках.

Таблица 1. Таблица с синдромами ошибок

	Синдромы ошибок			
Хэш-коды	–	$[\mathbf{M}_{100}]$	...	$[\mathbf{M}_{011}]$
$\mathbf{H}_{020}$	0	0	...	0
$\mathbf{H}_{021}$	0	0	...	1
$\mathbf{H}_{120}$	0	1	...	0
$\mathbf{H}_{121}$	0	0	...	0
$\mathbf{H}_{012}$	0	0	...	1
$\mathbf{H}_{002}$	0	0	...	0
$\mathbf{H}_{112}$	0	0	...	0
$\mathbf{H}_{102}$	0	1	...	0
$\mathbf{H}_{211}$	0	0	...	1
$\mathbf{H}_{201}$	0	0	...	0
$\mathbf{H}_{210}$	0	0	...	0
$\mathbf{H}_{200}$	0	1	...	0

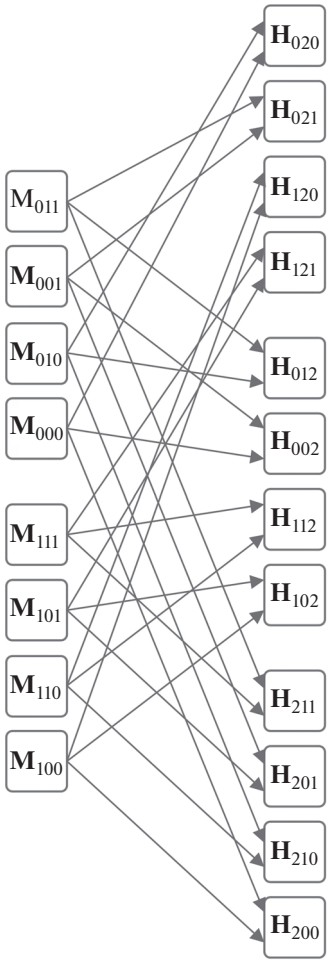

Рис. 11. Сеть хэширования.

Пример 4.3. Таблица синдромов ошибок, приводящих к нарушению целостности блоков данных $[\mathbf{M}_{100}]$ и $[\mathbf{M}_{011}]$, примет вид (табл. 1).

Сеть хэширования и соответствующая ей таблица синдромов ошибок для обнаружения и локализации q блоков данных с признаками нарушения целостности строятся по аналогичным правилам.

Построение модели восстановления целостности многомерных массивов данных. Блоки данных M_{ijr}, подлежащие защите, интерпретируются как элементы $\mathrm{GF}(2^t)$:

$$\Omega_{ijr}(\varpi) = \sum_{g=0}^{t-1} \mu_g^{(ijr)} \varpi^g = \mu_{t-1}^{(ijr)} \varpi^{t-1} + \\ + \mu_{t-2}^{(ijr)} \varpi^{t-2} + \ldots + \mu_0^{(ijr)}, \tag{4.1}$$

где ϖ – фиктивная переменная; $\mu_g^{(ijr)} \in \{0,1\}$; $g = t-1,\ t-2,\ldots,0$. Вычисленные от блоков данных M_{ijr} эталонные хэш-коды H_{ijr} также интерпретируются как элементы $\mathrm{GF}(2^t)$.

При этом представленные согласно выражению (4.1) блоки данных и эталонные хэш-коды являются наименьшими полиномиальными вычетами по основаниям $p_{ijr}(\varpi)$ таким, что

$$\gcd(p_{i_1jr}^{\mapsto(x)}(\varpi), p_{i_2jr}^{\mapsto(x)}(\varpi)) = 1, \tag{4.2}$$

где $p_{i_1jr}^{\mapsto(x)}(\varpi)$, $p_{i_2jr}^{\mapsto(x)}(\varpi)$ представлены при помощи сечений ориентации (x); $i_1, i_2 = 0,1,\ldots,k$; $i_1 \neq i_2$. При этом

$$\deg \Omega_{ijr}(\varpi) < \deg p_{ijr}(\varpi), \tag{4.3}$$

где $\deg \Omega_{ijr}(\varpi)$ – степень полинома $\Omega_{ijr}(\varpi)$.

Полученный массив, представленный посредством сечений ориентации (x):

$$\Omega[k+1,k+1,k+1] = \tag{4.4}$$

$$= \begin{Vmatrix} \Omega_{000}(\varpi), & \ldots, & \Omega_{0k0}(\varpi) \\ \vdots & \ddots & \vdots \\ \Omega_{0,0,k-1}(\varpi), & \ldots, & \Omega_{0,k,k-1}(\varpi) \\ \Omega_{00k}(\varpi), & \ldots, & 0 \end{Vmatrix} \ldots$$

$$\ldots \begin{Vmatrix} \Omega_{k-1,0,0}(\varpi), & \ldots, & \Omega_{k-1,k,0}(\varpi) \\ \vdots & \ddots & \vdots \\ \Omega_{k-1,0,k-1}(\varpi), & \ldots, & \Omega_{k-1,k,k-1}(\varpi) \\ \Omega_{k-1,0,k}(\varpi), & \ldots, & 0 \end{Vmatrix}$$

$$\begin{Vmatrix} \Omega_{k00}(\varpi), & \ldots, & 0 \\ \vdots & \ddots & \vdots \\ \Omega_{k,0,k-1}(\varpi), & \ldots, & 0 \\ 0, & \ldots, & 0 \end{Vmatrix}^{\mapsto(x)} \begin{matrix} \to (y) \\ \downarrow \\ (z), \end{matrix}$$

будем рассматривать как единый суперблок МПК по системе оснований:

$$\mathbf{P}[k+1,k+1,k+1] = \tag{4.5}$$

$$= \begin{Vmatrix} p_{000}(\varpi), & \ldots, & p_{0k0}(\varpi) \\ \vdots & \ddots & \vdots \\ p_{0,0,k-1}(\varpi), & \ldots, & p_{0,k,k-1}(\varpi) \\ p_{00k}(\varpi), & \ldots, & 0 \end{Vmatrix} \ldots$$

$$\ldots \begin{Vmatrix} p_{k-1,0,0}(\varpi), & \ldots, & p_{k-1,k,0}(\varpi) \\ \vdots & \ddots & \vdots \\ p_{k-1,0,k-1}(\varpi), & \ldots, & p_{k-1,k,k-1}(\varpi) \\ p_{k-1,0,k}(\varpi), & \ldots, & 0 \end{Vmatrix}$$

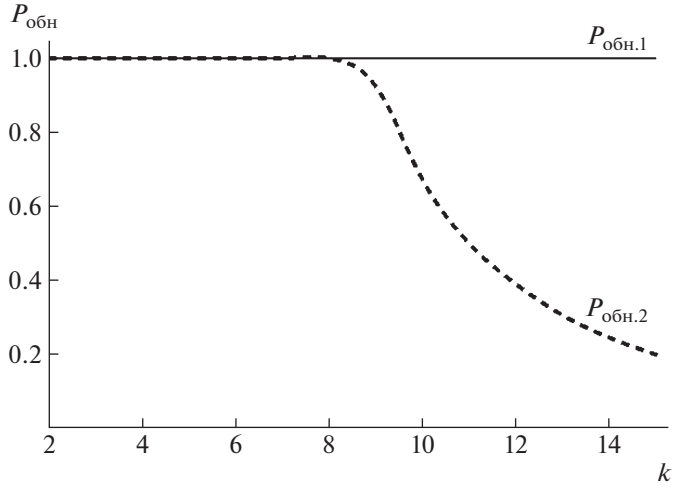

Рис. 12. Зависимости вероятностей $P_{\text{обн.1}}$, $P_{\text{обн.2}}$.

$$\left|\begin{matrix} p_{k00}(\varpi), & \ldots, & 0 \\ \vdots & \ddots & \vdots \\ p_{k,0,k-1}(\varpi), & \ldots, & 0 \\ 0, & \ldots, & 0 \end{matrix}\right|^{\mapsto(x)} \begin{matrix} \to (y) \\ \downarrow \\ (z). \end{matrix}$$

В соответствии с Китайской теоремой об остатках для многочленов, представленных в виде (4.5) и удовлетворяющих условию (4.2), и многочленов, представленных в виде (4.4) таких, что выполняется условие (4.3), система сравнений:

$$\begin{cases} \Xi(\varpi) \equiv \Omega_{000}(\varpi) \bmod p_{000}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{0k0}(\varpi) \bmod p_{0k0}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{00k}(\varpi) \bmod p_{00k}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{0,k-1,k}(\varpi) \bmod p_{0,k-1,k}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{k00}(\varpi) \bmod p_{k00}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{k,k-1,0}(\varpi) \bmod p_{k,k-1,0}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{k,0,k-1}(\varpi) \bmod p_{k,0,k-1}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Xi(\varpi) \equiv \Omega_{k,k-1,k-1}(\varpi) \bmod p_{k,k-1,k-1}(\varpi) \end{cases} \tag{4.6}$$

имеет единственное решение $\Xi(\varpi)$ [24].

Выполним операцию расширения МПК путем введения $n-k$ избыточных оснований: $p_{0,k+1,0}(\varpi)$, ..., $p_{0,k-1,n}(\varpi)$, ..., $p_{k-1,k+1,0}(\varpi)$, ..., $p_{k-1,k-1,n}(\varpi)$, $p_{k+1,0,0}(\varpi)$, ..., $p_{n,k-1,k-1}(\varpi)$ и получим соответствующие им избыточные вычеты:

$$\begin{cases} \Omega_{0,k+1,0}(\varpi) \equiv \Xi(\varpi) \bmod p_{0,k+1,0}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Omega_{0,k-1,n}(\varpi) \equiv \Xi(\varpi) \bmod p_{0,k-1,n}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Omega_{k-1,k+1,0}(\varpi) \equiv \Xi(\varpi) \bmod p_{k-1,k+1,0}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Omega_{k-1,k-1,n}(\varpi) \equiv \Xi(\varpi) \bmod p_{k-1,k-1,n}(\varpi), \\ \Omega_{k+1,0,0}(\varpi) \equiv \Xi(\varpi) \bmod p_{k+1,0,0}(\varpi), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Omega_{n,k-1,k-1}(\varpi) \equiv \Xi(\varpi) \bmod p_{n,k-1,k-1}(\varpi). \end{cases}$$

Причем выполняется условие (4.2), а условие (4.3) примет вид $\deg p_{000}(\varpi), \ldots, \deg p_{k,k-1,k-1}(\varpi) < \ldots < \deg p_{n,k-1,k-1}(\varpi)$.

Получим расширенный МПК в кольце многочленов:

$$\{\Omega_{000}(\varpi), \ldots, \Omega_{k,k-1,k-1}(\varpi), \Omega_{0,k+1,0}(\varpi), \ldots, \ldots, \Omega_{n,k-1,k-1}(\varpi)\}_{\text{МПК}}. \tag{4.7}$$

В соответствии с правилами построения кубических кодов [15] разместим избыточные блоки данных в свободных для записи ячейках массива.

Таким образом, полученный массив, содержащий n^3 ячеек, будет состоять из:

– k^3 блоков данных M_{ijr}, подлежащих защите целостности;

– $3k^2$ блоков с вычисленными эталонными хэш-кодами H_{ijr};

– $3k^2(n-k)$ избыточных блоков данных.

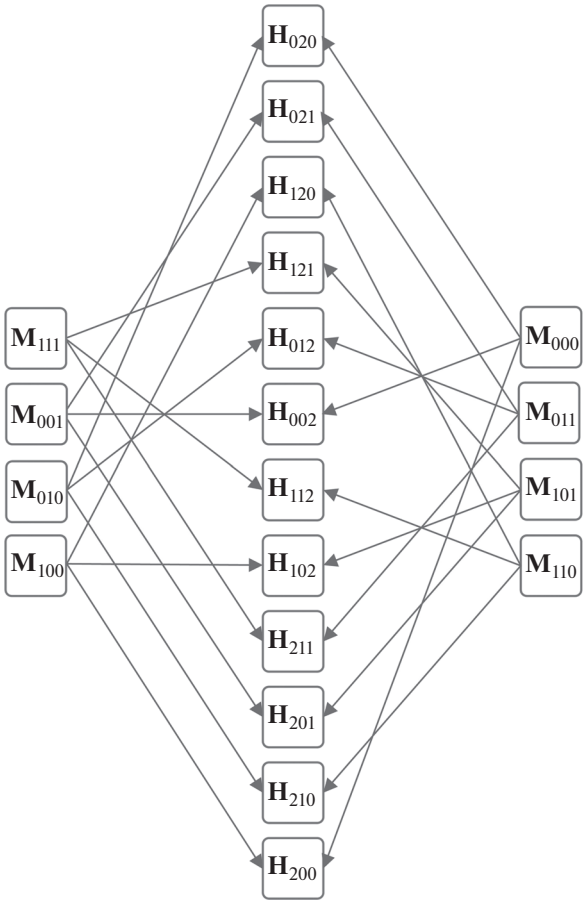

Рис. 13. Сеть хэширования для двух сочетаний блоков данных с совпадающими синдромами ошибок.

Данные, подлежащие защите, представленные в виде (4.7), отправляются на хранение.

Порядок восстановления целостности многомерных массивов данных. После осуществления контроля целостности данных в случае обнаружения ошибки выполняется процедура восстановления целостности данных.

В соответствии с правилами декодирования модулярных кодов [25–32] критерием отсутствия обнаруживаемых ошибок в расширенном МПК (4.7) является выполнение неравенства:

$$\deg \Xi'(\varpi) < \deg P_{nnn}(\varpi),$$

где $\Xi'(\varpi)$ является решением системы (4.6) для $\Omega'_{000}(\varpi),\ldots,\Omega'_{k,k-1,k-1}(\varpi)$, $\Omega'_{0,k+1,0}(\varpi),\ldots,\Omega'_{n,k-1,k-1}(\varpi)$; $P_{nnn}(\varpi) = \prod_{i,j,r=0}^{n} p_{ijr}(\varpi)$, символ "$(\cdot)'$" указывает на возможное нарушение целостности данных.

Критерий обнаруживаемой ошибки – выполнение неравенства:

$$\deg \Xi'(\varpi) \geq \deg P_{nnn}(\varpi).$$

Таблица 2. Таблица с совпадающими синдромами ошибок для двух сочетаний блоков данных

	Синдромы ошибок	
Хэш-коды	1 сочетание	2 сочетание
$\mathbf{H}_{020}$	1	1
$\mathbf{H}_{021}$	1	1
$\mathbf{H}_{120}$	1	1
$\mathbf{H}_{121}$	1	1
$\mathbf{H}_{012}$	1	1
$\mathbf{H}_{002}$	1	1
$\mathbf{H}_{112}$	1	1
$\mathbf{H}_{102}$	1	1
$\mathbf{H}_{211}$	1	1
$\mathbf{H}_{201}$	1	1
$\mathbf{H}_{210}$	1	1
$\mathbf{H}_{200}$	1	1

Восстановление целостности блока $[\Omega_{ijr}(\varpi)]$ выполняется путем вычисления наименьшего вычета:

$$\Omega_{ijr}(\varpi) \equiv \Xi(\varpi) \bmod p_{ijr}(\varpi),$$

где $\Xi(\varpi)$ повторно вычислено с учетом исключения искаженного блока данных $[\Omega_{ijr}(\varpi)]$.

Проверка достоверности и полноты данных после восстановления их целостности выполняется путем повторного сравнения значений эталонных хэш-кодов и вычисленных хэш-кодов от блоков данных.

5. ОЦЕНИВАНИЕ РЕЗУЛЬТАТОВ

Оценивание выполним в сравнении с существующим решением, в котором контроль целостности данных осуществляется посредством классического хэширования блоков данных, подлежащих защите, а восстановление целостности посредством кодов, корректирующих ошибки (МПК).

Оценивание результатов при контроле целостности данных. Разработанный метод в сравнении с существующим решением при фиксированном объеме вводимой избыточности позволяет гарантированно обнаружить бóльшее количество блоков данных с признаками нарушения целостности.

Так, например, СХД объемом 432 Кбайт позволяет хранить 675 хэш-кодов (при размере 1-го хэш-кода – 512 бит [33]).

Такое количество хэш-кодов при применении разработанного метода обеспечивает гарантированное обнаружение и локализацию (без учета коллизий) всех блоков данных с признаками на-

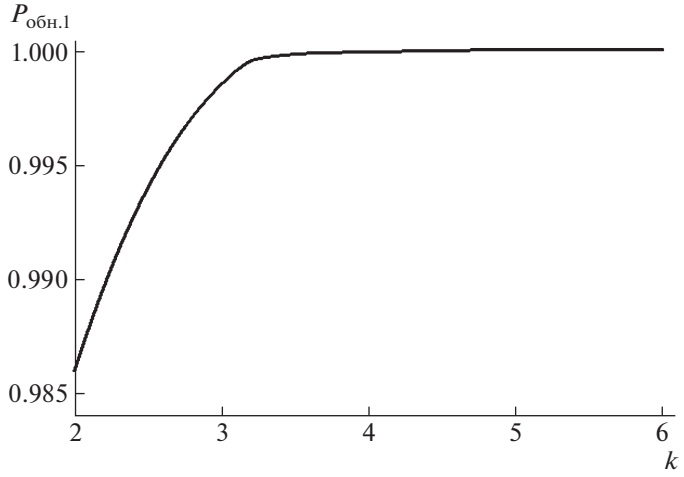

Рис. 14. Зависимость вероятности $P_{обн.1}$.

рушения целостности, размещенных в 3-мерном массиве с размером от $k = 2$ до $k = 15$.

Зависимости вероятностей $P_{обн}$ обнаружения и локализации блоков данных с признаками нарушения целостности представлены на рисунке 12, где $P_{обн.1}$ – вероятность, при применении разработанного метода, $P_{обн.2}$ – вероятность, при применении существующего решения.

Разработанный метод позволяет повысить вероятность обнаружения и локализации q-кратных ошибок. Его применение (без учета коллизий) позволяет обнаруживать и локализовывать все 1-, 2-, 3-кратные ошибки, а также до ≈98.6% всех 4-кратных ошибок без увеличения для этого количества вычисляемых хэш-кодов.

В частности, при $k = 2$ в 3-мерном массиве из всех 70 возможных различных сочетаний по четыре блока данных обнаруживаются и локализуются (без учета коллизий) до ≈98.6% всех ошибок, за исключением одной (≈1.4%), возникающей в одном из следующих сочетаний блоков данных:

- первое сочетание: $\mathbf{M}_{001}$, $\mathbf{M}_{010}$, $\mathbf{M}_{100}$, $\mathbf{M}_{111}$;
- второе сочетание: $\mathbf{M}_{000}$, $\mathbf{M}_{011}$, $\mathbf{M}_{101}$, $\mathbf{M}_{110}$.

Для рассматриваемых сочетаний построим сеть хэширования (рис. 13), на основе которой составим таблицу синдромов ошибок (табл. 2), из которой видно, что для представленных сочетаний блоков данных, подлежащих защите, синдромы ошибок совпадают.

При $k = 3$ в 3-мерном массиве существует 17550 различных сочетаний по четыре блока данных. В этом случае обнаруживаются и локализуются (без учета коллизий) до ≈99.85% всех ошибок, за исключением 27 (≈0.15%), которые не обнаружи-

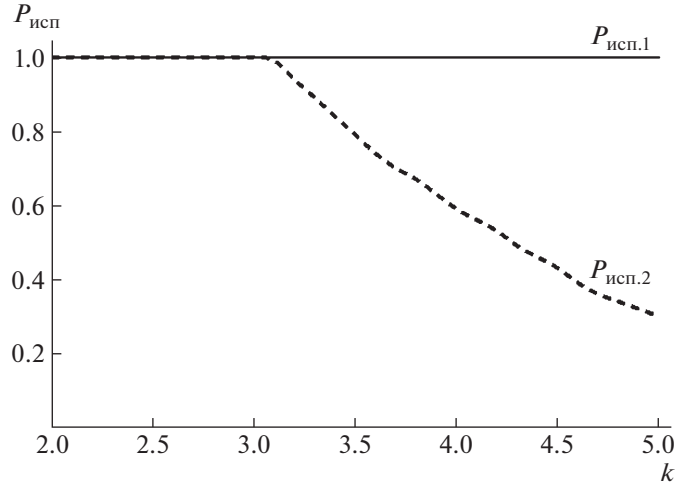

Рис. 15. Зависимости вероятностей $P_{исп.1}$, $P_{исп.2}$.

ваются в результате совпадения синдромов ошибок у 27 различных пар сочетаний блоков данных.

При последующем возрастании размерности k количество всех обнаруживаемых и локализуемых 4-кратных ошибок при применении разработанного метода стремится к 100% (без учета коллизий).

Зависимость вероятности $P_{\text{обн.1}}$ обнаружения и локализации 4-кратных ошибок при возрастании размерности k представлена на рис. 14.

Оценивание результатов при восстановлении целостности данных. В разработанном методе, учитывающем структуру многомерного представления данных, в соответствии с полученной моделью избыточные блоки данных располагаются по осям гиперкуба, что в сравнении с классическим применением избыточного МПК позволяет исправлять бóльшее количество возникающих ошибок.

В зависимости от размерности k максимальное количество возможных q блоков данных с нарушением целостности в кубе определяется в соответствии с: $q = k^3$. При этом для восстановления целостности всех искаженных блоков данных в разработанном методе потребуется $3k^2$ избыточных блоков данных. В то же время, классическое применение избыточного МПК, при котором, как известно [25, 26] исправляется q или менее ошибок, если $n - k \geq 2q$, требуется $2k^3$ избыточных блоков данных.

Зависимости вероятностей $P_{\text{исп}}$ исправления возникающих ошибок представлены на рисунке 15, где $P_{\text{исп.1}}$ – вероятность, при применении разработанного метода, $P_{\text{исп.2}}$ – вероятность, при применении существующего решения.

Повышение вероятности исправления возникающих ошибок за счет использования разработанного метода объясняется тем, что $n - k$ избыточных блоков данных используется для восстановления целостности не только $\frac{n-k}{2}$ искаженных блоков данных, как при классическом применении избыточного МПК, но и других блоков данных, подлежащих защите, в соответствии с введенными связями в 3-мерном кубе при построении модели восстановления целостности многомерных массивов данных.

6. ЗАКЛЮЧЕНИЕ

Представленное решение основано на агрегировании методов криптографической защиты информации (область теоретических положений защиты информации) и теории кодов контролирующих ошибки (область теории надёжности).

Объединение существующих методов контроля и восстановления целостности данных с учетом особенностей многомерной модели представления данных в современных СХД открывает новые возможности для повышения вероятности обнаружения и исправления возникающих ошибок без требуемого для этого увеличения объема вводимой избыточности, что, как следствие, приводит к сокращению времени загрузки новых данных в СХД и повышению эффективности ИС в целом.

СПИСОК ЛИТЕРАТУРЫ

1. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. ГОСТ 34.003-90. М.: Стандартинформ, 2009. 48 с.
2. *Петухов Г.Б., Якунин В.И.* Методологические основы внешнего проектирования целенаправленных процессов и целеустремленных систем. М.: АСТ, 2006. 504 с.
3. Федеральный закон “Об информации, информационных технологиях и о защите информации” от 27.07.2006 № 149-ФЗ. 99 с.
4. *Schneier B.* Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons, Inc. 2016. 653 p.
5. *Menezes A.J., Oorschot P., Vanstone S.* Handbook of Applied Cryptography. CRC Press, Inc. 2015. 770 p.
6. *Knuth D.E.* The Art of Computer Programming: Volume 3: Sorting and Searching. 2nd edn. 2018. 803 p.
7. *Biham E., Dunkelman O.* A framework for iterative hash functions. – HAIFA. ePrint Archive, Report 2007/278. 2007. P. 1–20.
8. *Wang X., Yu H.* How to break MD5 and Other Hash Function. EUROCRYPT. LNCS 3494. Springer-Verlag. 2005. P. 19–35.
9. *Bellare M.* New Proofs for NMAC and HMAC: Security without Collision-Resistance.CRYPTO. ePrint Archive, Report 2006/043. 2006. P. 1–16.
10. *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. P. 14–35.
11. *Сапожников В.В.* Основы теории надежности и технической диагностики. СПб.: Лань, 2019. 588 с.
12. *Хетагуров Я.А.* Повышение надежности цифровых устройств методами избыточного кодирования. М.: Энергия, 2015. 376 с.
13. *Зубарев Ю.М.* Основы надежности машин и сложных систем. СПб.: Лань, 2017. 180 с.
14. *Morelos-Zaragoza R.H.* The Art of Error Correcting Coding. 2nd edn John Wiley & Sons, Inc. 2006. 263 p.
15. *Hamming R.* Coding and Information Theory. Prentice-Hall, 2008. 259 p.
16. *Соколов Н.П.* Введение в теорию многомерных матриц. М.: Просвещение, 2012. 175 с.
17. *Tilborg H.* Fundamentals of cryptology. A professional reference and interactive tutorial. Kluwer Academic Publishers, 2014. 414 p.
18. *Dichenko S.A., Finko O.A.* Two-dimensional control and assurance of data integrity in information systems based on residue number system codes and cryptographic hash functions // Integrating Research Agen-

das and Devising Joint Challenges International Multidisciplinary Symposium ICT Research in Russian Federation and Europe. 2018. P. 139–146.

19. *Диченко С.А., Финько О.А.* Гибридный крипто-кодовый метод контроля и восстановления целостности данных для защищенных информационно-аналитических систем // Вопросы кибербезопасности. 2019. № 6 (34). С. 17–36.
20. *Диченко С.А.* Контроль и обеспечение целостности информации в системах хранения данных // Наукоемкие технологии в космических исследованиях Земли. 2019. Т. 11. № 1. С. 49–57.
21. *Диченко С.А., Финько О.А.* Обобщенный способ применения хэш-функции для контроля целостности данных // Наукоемкие технологии в космических исследованиях Земли. 2020. Т. 12. № 6. С. 48–59.
22. Способ многоуровневого контроля и обеспечения целостности данных. Пат. на изобретение RU 2707940. Заявка № 2019103723 от 11.02.2019. Опуб. 02.12.2019. Бюл. № 34. 17 с.
23. Способ двумерного контроля и обеспечения целостности данных. Пат. на изобретение RU 2696425. Заявка № 2018118919 от 22.05.2018. Опуб. 02.08.2019. Бюл. № 22. 26 с.
24. *Ananda Mohan P.V.* Residue Number Systems. Springer International Publishing, 2016. 351 p.
25. *Акушский И.Я., Юдицкий Д.М.* Машинная арифметика в остаточных классах. М.: Советское радио, 1968. 604 с.
26. *Бояринов И.М.* Помехоустойчивое кодирование числовой информации. М.: Наука, 1983. 386 с.
27. *Szabo N.S., Tanaka R.I.* Residue Arithmetic and its Applications Computer Technology. New York: McGraw-Hill, 1967. 476 p.
28. *Амербаев В.М.* Теоретические основы машинной арифметики. Алма-Ата: Наука, 1976. 498 с.
29. *Торгашов В.А.* Система остаточных классов и надежность ЦВМ. М.: Советское радио, 1973. 329 с.
30. *Mandelbaum D.M.* Error correction in residue arithmetic // IEEE Trans. Comput. 1972. № 21. P. 538–545.
31. *Omondi A., Premkumar B.* Residue Number Systems: Theory and Implementation. Imperial College Press. London, 2007. 296 p.
32. *Mandelbaum D.M.* A method of coding for multiple errors // IEEE Trans. OnInformation Theory. 1968. № 14. P. 518–521.
33. Информационная технология. Криптографическая защита информации. Функция хэширования. ГОСТ Р 34.11-2012. М.: Стандартинформ, 2013. 30 с.