Two-dimensional control and assurance of data integrity in information systems based on residue number system codes and cryptographic hash functions

Sergey Dichenko
University Teachers
Institute of Computer Systems and Information Security of Kuban State Technological University
Krasnodar, Russia
dichenko.sa@yandex.ru

Oleg Finko
Professor
Institute of Computer Systems and Information Security of Kuban State Technological University
Krasnodar, Russia
ofinko@yandex.ru

Abstract

The method of two-dimensional control and assurance of data integrity with the possibility of their recovery for information systems operating under conditions of random errors as well as errors generated through deliberate actions of the attacker is proposed. The data recovery procedure is based on the application of the mathematical apparatus of redundant residue number system codes, and the control (verification of the recovered data validity (reliability, accuracy)) of data integrity is performed by means of cryptographic methods.

1 Introduction

At present, users of various information systems are facing the tasks of protecting the data processed in them. One of the measures to ensure the security of data processed in information systems is the protection of their integrity [ISO05].

The problem solution of data integrity protection becomes especially urgent during the operation of widely created data processing centers when using different processing facilities in their composition with different building structures and operating principles under conditions of both random errors and errors generated through deliberate actions of an attacker (unauthorized data modification (for example, through the action of malicious code) or the failure of a part of the media (for example, individual cells, sectors)).

The challenge of data integrity protection is complicated because of its complexity, as it involves not only data integrity control, but also its provision, which means the restoration of data whose integrity has been violated for some reason.

There are various ways of solving the problem of control and assurance of data integrity, among which the following are of the greatest interest.

In: Marco Schaerf, Massimo Mecella, Drozdova Viktoria Igorevna, Kalmykov Igor Anatolievich (eds.): Proceedings of REMS 2018 – Russian Federation & Europe Multidisciplinary Symposium on Computer Science and ICT, Stavropol – Dombay, Russia, 15–20 October 2018, published at http://ceur-ws.org

2 Analysis of existing solutions for control and assurance of data integrity

There are known ways to control the data integrity by calculating checksum values and comparing them with reference values, as well as methods based on the use of cryptographic methods: key and keyless hashing, means of electronic signature [Knu73, Men96, Bih07, Bel06]. The disadvantage of these methods is the lack of the ability to insure their integrity without introducing an additional data recovery mechanism.

There are known ways to ensure the integrity of data through the use of various types of reservation (using hardware or software implementation of RAID technology (Redundant Array of Independent Disks) (RAID arrays)), duplication methods, redundant coding methods [Hen13, Mor06, Ham80]. The disadvantage of these methods is high redundancy.

The presented solutions show that some of the methods allow to control the data integrity by comparing the values of the reference and calculated hash-codes of the hash function (checksums) when requesting the use of processed or stored data, but the lack of a mechanism for their recovery does not allow their integrity assurance. Other methods, on the other hand, provide data integrity by restoring them, for example, from a backup copy, but their practical use without data integrity control is ineffective. Individual methods allow for control and ensure the data integrity, however, of valuable high redundancy.

The most popular solutions are the complex protection of data integrity associated with the simultaneous solution of control tasks and ensuring data integrity, which is achieved by consistently applying first the cryptographic transformation to data, and then applying the technology of data backup.

At the same time, data integrity protection is relevant both for systems of RAID type, where all media are located in one constructive block, and for distributed storage systems, that is, for network storage.

Thus, in order to protect the data integrity, when considering this notion in a complex, it is necessary to aggregate existing solutions. Combining the known methods in one allows you to control and ensure the data integrity.

3 Choosing ways to control the data integrity and recovery to share them while ensuring integrity

A method [App05] is known where, before writing to a RAID array (after reading), the data is encrypted (decrypted) by a dedicated device connected to the PCI-tyre, the encryption key being read from an external storage device and/or requested from the user. In [Pat12], before writing to the array, the data is divided into several segments, after which the checksums are calculated separately from the data from each segment. The data segments and checksums are further distributed over the disks of the RAID array.

In [Pat10], a method of protecting data in a network storage is proposed, where a user's request for reading (writing) data first passes the authorization procedure, and only if the operation is allowed, the data on the network storage is decrypted (encrypted) accordingly. The keys of encryption (decryption) are stored on the client side.

Another version of combined protection is proposed in [App11], where the data is stored in the cloud, and the encryption module is stored not on the client side, but on the side of the cloud storage provider. This solution is intended, as a rule, to protect the backup copies of data in the cloud, although the original data is stored on the client side in its original form. In order to protect the data, the data file is first divided into parts, and then each part is transformed using a cryptographic algorithm and written to one or more media in the cloud. Protection is provided when data is lost on the client side. In this case, the backup is restored from the cloud.

The disadvantage of the presented combined methods is the high redundancy, as well as the lack of the possibility, without the introduction of an additional monitoring mechanism, to verify the validity (reliability, accuracy) of the recovered data while ensuring their integrity.

In order to eliminate the drawbacks of the known combined methods, a solution is proposed in which cryptographic methods are chosen to perform data integrity control, in particular, a hash function designed specifically for this purpose, and the data recovery procedure is performed by using redundant residue number system codes, the application of the mathematical apparatus of which allows to provide minimal redundancy, and most importantly, provides, when used together with cryptographic methods the construction of unique scheme which allows to verify the validity (reliability, accuracy) of recovered data while ensuring its integrity in case of violation.

4 Structural-parametric synthesis of the system of parallel control and assurance of data integrity

For control and integrity purposes, the data blocks M_i ($i = 1, 2, \ldots, n$), to be protected are represented in the form of sub-blocks of fixed length $M_i = \{m_{i,1}||m_{i,2}||\ldots||m_{i,n}\}$, where $0 \le m_{i,j} < p_{i,j}$ ($i, j = 1, \ldots, n$; $p_{i,j} \in \mathbb{N}$); "||" — is the concatenation operation, n — is the number of data blocks M_i, to be protected, and also fixed-length sub-blocks in each data block under consideration M_i. And the length of the data blocks M_i.

Obtaining the matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{bmatrix}.$$

To implement integrity control, a hash function is applied to the data blocks M_i, the construction rules of which are defined in [ISO08]. The received hash-codes S_i (or MAC) hash functions $h(M_i)$ (or $h_k(M_i)$, k — secret key) from data blocks M_i will be the reference codes, we obtain the matrix $\mathbf{\Psi}$:

$$\mathbf{\Psi} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} & \to & s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} & \to & s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} & \to & s_{n,1} & s_{n,2} & \cdots & s_{n,n} \end{bmatrix},$$

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

Now consider the data blocks M_j ($j = 1, 2, \ldots, n$) represented by sub-blocks $m_{1,1}, m_{2,1}, \ldots, m_{n,1}$; $m_{1,2}, m_{2,2}, \ldots, m_{n,2}; \ldots; m_{1,n}, m_{2,n}, \ldots, m_{n,n}$. The sub-blocks of the $m_{i,j}$ data blocks under consideration M_j are interpreted as the minimum nonnegative deductions from the generically ordered, mutually simple modules $p_{i,j}$, and form an information super-block of the residue number system (RNS) codes.

As a result of the base extension, we obtain redundant sub-blocks $m_{n+1,1}, m_{n+2,1}, \ldots, m_{k,1}$; $m_{n+1,2}, m_{n+2,2}, \ldots, m_{k,2}; \ldots; m_{n+1,n}, m_{n+2,n}, \ldots, m_{k,n}$, the set of which together with the sub-blocks forming a single super-block of elements form a vector of redundant residue number system (RRNS) codes.

We get the matrix $\mathbf{\Upsilon}$ with redundant sub-blocks of the vector of RRNS codes:

$$\mathbf{\Upsilon} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} & s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} & s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} & s_{n,1} & s_{n,2} & \cdots & s_{n,n} \\ \downarrow & \downarrow & \vdots & \downarrow & & & & \\ m_{n+1,1} & m_{n+1,2} & \cdots & m_{n+1,n} & & & & \\ m_{n+2,1} & m_{n+2,2} & \cdots & m_{n+2,n} & & & & \\ \vdots & \vdots & \ddots & \vdots & & & & \\ m_{k,1} & m_{k,2} & \cdots & m_{k,n} & & & & \end{bmatrix}.$$

Let: $s_{i,j} = \sum_{t=0}^{-1+\log p_{i,j}} \varsigma_{i,j}^{(t)} 2^t$; $m_{i,j} = \sum_{t=0}^{-1+\log p_{i,j}} \mu(t)_{i,j} 2^{i-n-1}$; $g_{i,j} = \sum_{t=0}^{-1+\log p_{i,j}} \zeta_{i,j}^{(t)} 2^{j-1}$; $\mu_{i,j}, \varsigma_{i,j}, \zeta_{i,j} \in \{0, 1\}$. Then the numbers $m_{i,j}$, $s_{i,j}$ and $g_{i,j}$ can be mapped to binary vectors:
$\mathbf{s}_{i,j} = [\ \varsigma_{i,j}^{(0)} \quad \varsigma_{i,j}^{(1)} \quad \ldots \quad \varsigma_{i,j}^{(-1+\log p_{i,j})}\ ]$; $\mathbf{m}_{i,j} = [\ \mu_{i,j}^{(0)} \quad \mu_{i,j}^{(1)} \quad \ldots \quad \mu_{i,j}^{(-1+\log p_{i,j})}\ ]$;
$\mathbf{g}_{i,j} = [\ \zeta_{i,j}^{(0)} \quad \zeta_{i,j}^{(1)} \quad \ldots \quad \zeta_{i,j}^{(-1+\log p_{i,j})}\ ]$. We add the i-th sub-blocks of hash-codes S_i with j-th redundant sub-blocks of data blocks M_j^* of the vector of RRNS codes:

$$\mathbf{G}_i = \mathbf{S}_i \oplus \mathbf{M}_j^* = [\ \mathbf{s}_{i,1} \oplus \mathbf{m}_{n+1,j} \quad \mathbf{s}_{i,2} \oplus \mathbf{m}_{n+2,j} \quad \ldots \quad \mathbf{s}_{i,n} \oplus \mathbf{m}_{k,j}\ ]$$

where the sign "$\oplus$" denotes the summation in the $\mathbb{GF}(2)$;

$$\mathbf{S}_i = [\ \mathbf{s}_{i,1} \quad \mathbf{s}_{i,2} \quad \ldots \quad \mathbf{s}_{i,n}\ ];\ \mathbf{M}_j^* = [\ \mathbf{m}_{n+1,j} \quad \mathbf{m}_{n+2,j} \quad \ldots \quad \mathbf{m}_{k,j}\ ];\ \mathbf{G}_i = [\ \mathbf{g}_{i,1} \quad \mathbf{g}_{i,2} \quad \ldots \quad \mathbf{g}_{i,n}\ ].$$

We obtain the matrix $\mathbf{\Omega}$:

$$\mathbf{\Omega} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} & g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} & g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} & g_{n,1} & g_{n,2} & \cdots & g_{n,n} \end{bmatrix}. \tag{1}$$

At the end of the preparatory stage of the construction of the system (Figure 1), the data subject to protection is presented in the form (1), which will allow control and ensuring their integrity.

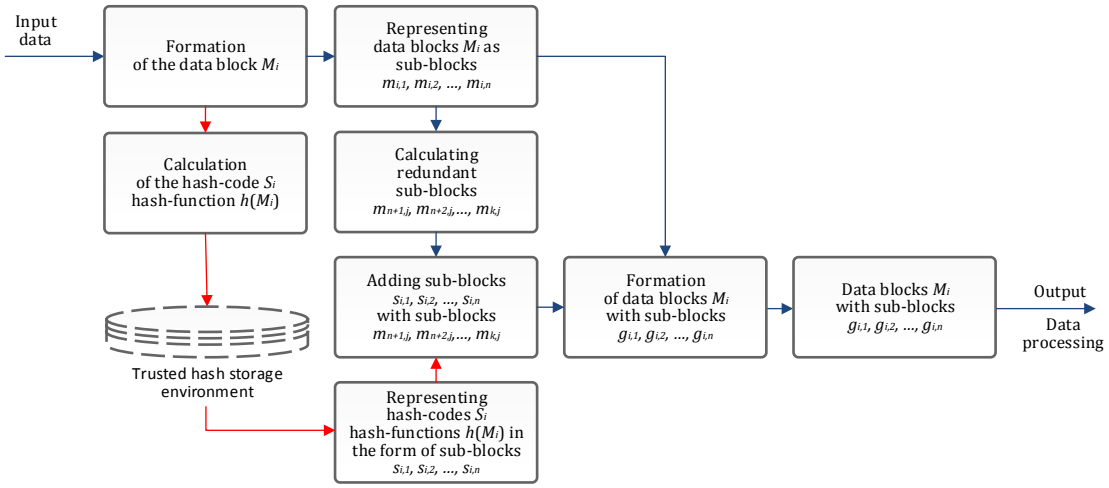

Figure 1: A diagram explaining the preparatory stage of the system construction

5 Procedure for control of data integrity

When requesting the use of data (the main stage) to be protected, they are control for their integrity, which can be ensured by performing the base extension the information super-block of RRNS codes [Baj04, Baj05], with this redundant sub-blocks are created $m'_{n+1,1}, m'_{n+2,1}, \ldots, m'_{k,1}; m'_{n+1,2}, m'_{n+2,2}, \ldots, m'_{k,2}; \ldots$ $\ldots; m'_{n+1,n}, m'_{n+2,n}, \ldots, m'_{k,n}$ of the data blocks $M_j^{*\prime}$ of the vector of the RRNS codes, where "$\bullet'$" denotes that changes could occur in sub-blocks $m'_{1,1}, m'_{1,2}, \ldots, m'_{1,n}; m'_{2,1}, m'_{2,2}, \ldots, m'_{2,n}; \ldots; m'_{n,1}, m'_{n,2}, \ldots, m'_{n,n}$ of data blocks M'_i.

The matrix $\mathbf{\Omega}$ with the redundant sub-blocks of the vector of RRNS codes takes the form:

$$\mathbf{\Omega}' = \begin{bmatrix} m'_{1,1} & m'_{1,2} & \cdots & m'_{1,n} & g'_{1,1} & g'_{1,2} & \cdots & g'_{1,n} \\ m'_{2,1} & m'_{2,2} & \cdots & m'_{2,n} & g'_{2,1} & g'_{2,2} & \cdots & g'_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ m'_{n,1} & m'_{n,2} & \cdots & m'_{n,n} & g'_{n,1} & g'_{n,2} & \cdots & g'_{n,n} \\ \downarrow & \downarrow & \vdots & \downarrow & & & & \\ m'_{n+1,1} & m'_{n+1,2} & \cdots & m'_{n+1,n} & & & & \\ m'_{n+2,1} & m'_{n+2,2} & \cdots & m'_{n+2,n} & & & & \\ \vdots & \vdots & \ddots & \vdots & & & & \\ m'_{k,1} & m'_{k,2} & \cdots & m'_{k,n} & & & & \end{bmatrix}.$$

We perform the inverse transformation (here the bit vectors $\mathbf{g}'_{i,j}$, $\mathbf{m}'_{i,j}$ is equivalent to the numbers $g'_{i,j}, m'_{i,j}$):

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

where $\mathbf{S}'_i = [\ \mathbf{s}'_{i,1} \quad \mathbf{s}'_{i,2} \quad \dots \quad \mathbf{s}'_{i,n}\ ]$; $\mathbf{M}^{*\prime}_j = [\ \mathbf{m}'_{n+1,j} \quad \mathbf{m}'_{n+2,j} \quad \dots \quad \mathbf{m}'_{k,j}\ ]$; $i = j$.

Compare the values of the hash-codes obtained $\mathbf{S}'_i$ to the values of the previously calculated hash-codes $\mathbf{S}''_i$ (or $\mathbf{S}_i$) hash function $h(\mathbf{M}'_i)$. Based on the results of the comparison, let's make a conclusion:

- about the absence of violation of data integrity, at $\mathbf{S}'_i = \mathbf{S}''_i$ (or $\mathbf{S}'_i = \mathbf{S}_i$);
- about data integrity violation, when $\mathbf{S}'_i \neq \mathbf{S}''_i$ (or $\mathbf{S}'_i \neq \mathbf{S}_i$).

6 Procedure of ensuring the data integrity

If the values of the hash-codes of the hash function compared with each other are different, which will be characterized by the occurrence of an error (violation of integrity) in the data being processed, we shall perform its localization.

The localization of the detected error (sub-blocks $\tilde{m}_{i,j}$ with integrity violation) is performed initially on the rows of the matrix $\mathbf{\Omega}'$ (the i-th data block with the integrity violation, which includes the sub-block $\tilde{m}_{i,j}$ is determined), and then on the columns (the j-th data block with integrity violation, which includes the sub-block $\tilde{m}_{i,j}$ is determined).

A data block $\tilde{M}_i$ with integrity violation, whose sub-blocks are located along the row of the matrix $\mathbf{\Omega}'$, is determined from the results of a comparison of the calculated and reference hash-codes of the hash function. A data block $\tilde{M}_j$ with an integrity violation whose sub-blocks are arranged along the column of the matrix $\mathbf{\Omega}'$ is determined by means of a mathematical apparatus RRNS codes based on the fundamental provisions of the Chinese remainder theorem.

In accordance with the mathematical apparatus of RNS [Baj04], in which the tested data block M_j will be interpreted as a nonnegative integer A_j unambiguously represented by a set of residues on RNS basis $p_{1,j}, p_{2,j}, \dots, p_{n,j} < p_{n+1,j} < \dots < p_{k,j}$:

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

where $P_{n,j} = p_{1,j}p_{2,j}\dots p_{n,j} > A_j$; $\alpha_{i,j} = |A|_{p_{i,j}}$; $|\bullet|_p$ — is the smallest nonnegative residue of the number "$\bullet$" modulo p; $j = 1, 2, \dots, n, n+1, \dots, k$; $i = 1, 2, \dots, n$; $p_{1,j}, p_{2,j}, \dots, p_{n,j} < p_{n+1,j} < \dots < p_{k,j}$ — are pairwise simple.

The resulting residues $\alpha_{i,j}$ will be interpreted as sub-blocks $m_{i,j}$ of the data block M_j, that is, the remnants of the RNS $\alpha_{1,j}, \alpha_{2,j}, \dots, \alpha_{n,j}$ will be interpreted as sub-blocks $m_{1,j}, m_{2,j}, \dots, m_{n,j}$ and will be considered informational (informational group n sub-blocks), and $\alpha_{n+1,j}, \dots, \alpha_{k,j}$ — interpreted as sub-blocks $m_{n+1,j}, \dots, m_{k,j}$ and considered as control (redundant) (control (redundant) group $(k-n)$ sub-blocks). The RNS itself is in this case extended, where $P_{k,j} = P_{n,j}p_{n+1,j}\dots p_{k,j}$, and covers the complete set of states represented by all k deductions. This area will be the full range of the RNS $[0, P_{k,j})$ and consist of a working range $[0, P_{n,j})$, where $P_{n,j} = p_{1,j}p_{2,j}\dots p_{n,j}$, is defined by nonredundant of the RNS bases (sub-blocks $m_{1,j}, m_{2,j}, \dots, m_{n,j}$), and a range $[P_{n,j}, P_{k,j})$ defined by redundant of the RNS bases (sub-blocks $m_{n+1,j}, \dots, m_{k,j}$) and representing invalid area. This means that operations on the number A_j are performed in the range $[0, P_{k,j})$, and if the result of the RNS operation goes beyond the $P_{n,j}$, then there is a conclusion about the calculation error. Checking this rule allows you to localize the error in the data block $\tilde{M}_j$ of the matrix $\mathbf{\Omega}'$.

Example 1

Choose a base system $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$ for which the operating range is $P_4 = p_1p_2p_3p_4 =$ $= 2 \cdot 3 \cdot 5 \cdot 7 = 210$. Then introduce the control bases $p_5 = 11$, $p_6 = 13$, then the full range is defined as $P_6 = P_4p_5p_6 = 210 \cdot 11 \cdot 13 = 30030$.

Let us calculate the orthogonal bases of the system: $B_1 = (1, 0, 0, 0, 0, 0) = 15015$; $B_2 = (0, 1, 0, 0, 0, 0) =$ $= 20020$; $B_3 = (0, 0, 1, 0, 0, 0) = 6006$; $B_4 = (0, 0, 0, 1, 0, 0) = 25740$; $B_5 = (0, 0, 0, 0, 1, 0) = 16380$; $B_6 = (0, 0, 0, 0, 0, 1) = 6930$.

Given a number $A = (1, 2, 2, 3, 6, 4) = 17$. Instead of it, after data processing we received $\tilde{A} = (1, 2, 2, 3, 1, 4)$. To localize the error, calculate the value of the number $\tilde{A}$:

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

The resulting number is incorrect ($\tilde{A} > 210$), which indicates an error in the processing of data. As a result of localization, it was determined that the number $\tilde{\alpha}_5$ on the base $p_5 = 11$ was wrong.

After determining the data blocks $\tilde{M}_i$ and $\tilde{M}_j$ with broken integrity, a decision is made that an error occurred in the sub-block $\tilde{m}_{i,\,j}$, located at the intersection of the localized row and column of the matrix $\mathbf{\Omega}'$ an error occurred (data integrity violation). After localizing the error (finding the sub-block $\tilde{m}_{i,\,j}$ with integrity violation), we perform a reconfiguration, the possibility of which is provided by RRNS codes [Yan01].

The reconfiguration is performed by calculating A^* from the system of equations:

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

on the "correct" bases of the RNS:

$$A^* = |\tilde{\alpha}_1 B_{1,\,r} + \ldots + \tilde{\alpha}_n B_{n,\,r} + \ldots + \tilde{\alpha}_k B_{k,\,r}|_{P_r}\,, \tag{2}$$

where $\tilde{\alpha}_i$ — residue with error; $B_{i,\,r}$ — orthogonal bases; $i,\ r = 1,\ \ldots,\ n,\ \ldots,\ k;\ i \neq r$; $B_{i,\,r} = \dfrac{P_r \mu_{i,\,r}}{p_i}$; $P_r = \dfrac{P_k}{p_r}$; $\mu_{i,\,r}$ is chosen so that the following comparison takes place: $\dfrac{P_r \mu_{i,\,r}}{p_i} \equiv |1|_{p_i}$.

Let's compile Table 1 containing the values of the recalculated orthogonal bases and modules of the system, provided that a single error occurs on each basis of the RNS, respectively.

Table 1: Table of values of orthogonal bases and modules of the system

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

After calculating A^* on the correct bases of the system, we calculate α_i instead of the previously excluded from the calculation residue with error $\tilde{\alpha}_i$:

$$\alpha_i = |A^*|_{p_i}\,. \tag{3}$$

Example 2

In accordance with (2) we calculate A^* (*the initial data from Example 1*), using Table 1, we obtain

$$A^* = |\alpha_1 B_{1,\,r} + \ldots + \tilde{\alpha}_5 B_{5,\,r} + \alpha_6 B_{6,\,r}|_{P_5} = |1 \cdot B_{1,\,r} + \ldots + 0 \cdot B_{5,\,r} + 4 \cdot B_{6,\,r}|_{P_5} = 17.$$

In accordance with (3), we calculate α_i, we obtain

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

In the proposed system, a set of sub-blocks $m_{1,\,j},\ m_{2,\,j},\ \ldots,\ m_{n,\,j},\ m_{n+1,\,j},\ \ldots,\ m_{k,\,j}$, which is interpreted as RRNS codes that allow to detect an error at any stage of their processing (provided that the multiplicity of the guaranteed error to be detected $t_{\text{det}} = d_{\min} - 1$, where $d_{\min}$ — is the minimum code distance).

Restoration of data blocks M'_j in case of their integrity violation is possible by excluding from the recovery process any r sub-blocks without sacrificing the unambiguous representation (where $r = k - n$ — is the number of additional sub-blocks), so that the system of sub-blocks of data blocks M'_j will be interpreted as nonsystematic code, or an inseparable code, and then the sub-block is calculated $m_{i,\,j}$ instead of the previously excluded sub-block $\tilde{m}_{i,\,j}$ with the detected error.

Thus, the integrity of the data block M_i was ensured by control and restoring the data sub-block $\tilde{m}_{i,\,j}$ with broken integrity. Performing the verification of the data validity (reliability, accuracy) after recovery while

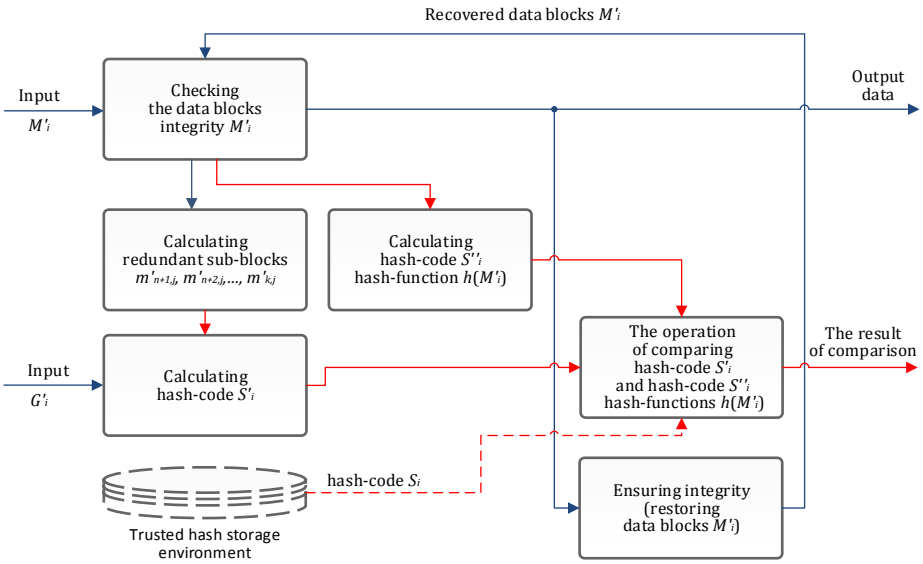

Figure 2: Scheme explaining the main stage of the system construction

ensuring their integrity in case of violation is performed by comparing the value of the previously calculated hash-code S_i' with the value of the calculated hash-code S_i'' hash function $h(M_i')$ already from the restored data block M_i' (Figure 2).

The general scheme of the developed method of two-dimensional control and data integrity in information systems based on RNSC and cryptographic hash functions is shown in the Figure 3.

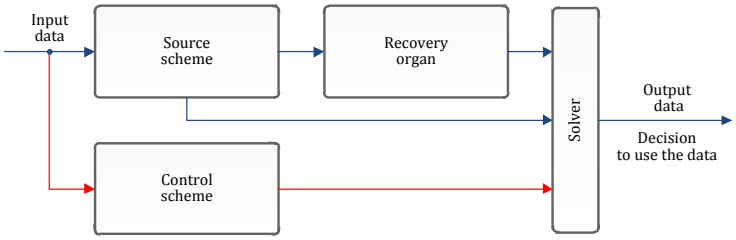

Figure 3: The general scheme of the developed method

7 Evaluation of the developed method

Evaluation of the developed method is carried out in comparison with the most popular of existing solutions integrated protection of data integrity, which consistently applies first cryptographic data transformation to control their integrity, and then a backup technology copy data to restore them in case of violation of integrity.

The indicator of quality is the redundancy factor K_{red}, which is calculated by the formula:

$$K_{\text{red}} = \frac{V_{\text{red.d}}^{(\text{con})} + V_{\text{red.d}}^{(\text{ass})}}{V_{\text{prot.d}}}, \tag{4}$$

where $V_{\text{red.d}}^{(\text{con})}$ — is the amount of redundant data entered to control the integrity of the protected data, $V_{\text{red.d}}^{(\text{ass})}$ — is the amount of redundant data entered to assurance the integrity of the protected data, $V_{\text{prot.d}}$ — is the amount of data to be protected. The criterion of quality is $K_{\text{red}} \to \min$.

Since the amount of redundant data $V_{\text{red.d}}^{(\text{con})}$ introduced to control integrity in the developed method and the

existing solution are equal, then (4) takes the form:

$$K_{\text{red}} = \frac{V_{\text{red.d}}^{(\text{ass})}}{V_{\text{prot.d}}}. \tag{5}$$

In accordance with (5) for the existing solution $K_{\text{red}} = 1$, since the amount of input redundancy is equal to the amount of data being protected ($V_{\text{red.d}}^{(\text{ass})} = V_{\text{prot.d}}$).

At the same time, to provide a level of data security, implemented in the technology of backup, in case of violation of integrity up to 2 sub-blocks of the data block you need to use the RRNS codes with two excess bases, in this case the redundancy of the control information is reduced from 100% (with backup technology) to 30-40% (RNS).

8 Conclusion

The results obtained provide scientific and engineering tools for control and ensuring the data integrity with the ability to verify their validity (reliability, accuracy) after recovery in case of violation of their integrity and provide the necessary conditions for creating promising and improving existing information systems for various purposes.

References

[ISO05] ISO/IEC 17799:2005. Information technology — Security techniques — Code of practice for information security management, 2005.

[Knu73] D. E. Knuth. *The Art of Computer Programming — Volume 3 / Sorting and Searching.* Addison-Wesley, 1973.

[Men96] A. J. Menezes, P. Oorschot, S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Inc, 1996.

[Bih07] E. Biham, O. Dunkelman. A framework for iterative hash functions. — HAIFA. ePrint Archive, Report 2007/278, 2007.

[Bel06] M. Bellare. New Proofs for NMAC and HMAC: Security without Collision-Resistance. CRYPTO, ePrint Archive, Report 2006/043, 2006.

[Hen13] S. Henry, Jr. Warren. *Hacker's delight.* Addison-Wesley, 2nd edn, 2013.

[Mor06] R. H. Morelos-Zaragoza. *The Art of Error Correcting Coding.* Addison-Wesley, 2nd edn, 2006.

[Ham80] R. Hamming. *Coding and Information Theory.* Prentice-Hall, 1980.

[App05] US Patent Application N20050081048, publ. 14/04/2005.

[Pat12] US Patent N8209551, publ. 26/06/2012.

[Pat10] US Patent N7752676, publ. 06/07/2010.

[App11] US Patent Application N20110107103, publ. 05/05/2011.

[ISO08] ISO/IEC 14888–1:2008. Information technology — Security techniques — Digital signatures with appendix — Part 1: General, 2008.

[Baj04] J.-C. Bajard, T. Plantard. RNS bases and conversions, SPIE Annual Meeting, Advenced Signal Proc. Alg., Architectures, and Implementation XIV. 60–69, https://www-almasty.lip6.fr/bajard/MesPublis/Spie2004.pdf, 2004.

[Baj05] J.-C. Bajard, N. Meloni, T. Plantard. Efficient RNS bases for Cryptography IMACS'05: World Congress: Scientific Computation, Applied Mathematics and Simulation. Paris, https://www-almasty.lip6.fr/bajard/MesPublis/IMACS2005.pdf, 2005.

[Yan01] L.-L. Yang, L. Hanzo. Coding Theory and Performance of Redundant Residue Number System Codes, Vehicular Technology Conference Fall. IEEE VTS 54th, https://pdfs.semanticscholar.org/e1f3/a5e67c24b5865990af3e8ba0a54bd6a86067.pdf, 2001.