Information Protection and Recovery Hamming Codes Based' Hash Technique

Serhii Mitsenko[b], Serhii Naumenko[a], Inna Rozlomii[a], Andrii Yarmilko[a]

[a] *Bohdan Khmelnytsky National University of Cherkasy, 81, Shevchenko Blvd., Cherkasy, 18031, Ukraine*
[b] *State University of Trade and Economics, 19, Kyoto str., Kyiv, 02156, Ukraine*

Abstract

Among the important tasks faced by users of information systems, one of the main ones is ensuring data security, particularly their integrity, and restoring damaged data. Among the methods of solving these problems, hash function schemes are of the most significant interest. However, the favorable properties of this approach are accompanied by high redundancy, especially when monitoring the integrity of a small-sized message. The article discusses the method of constructing a generalized cryptographic hashing method for integrity control and data recovery with the introduction of minimal redundancy. The proposed solutions for building interference-resistant systems for encoding and decoding digital data use self-controlled and self-adjusted linear block codes. It is suggested to use a hash code system to control data integrity. They are built according to rules similar to the rules for building linear redundant codes. In this case, the rules for constructing Hamming codes are applied. The focus of attention was on the rational selection of the necessary redundant code in order, on the one hand, to ensure the essential reliability of the information and, on the other hand, to avoid burdening the communication channels with a large amount of redundant data. The main advantage of the proposed method is the implementation of information integrity control and defect correction for a given level of security with minimal redundancy and the possibility of localizing integrity violations and correcting a given number of errors. It is shown that the rules for constructing linear codes are similar to those for building linear redundant codes, in particular Hamming codes, which makes it possible to adapt the theory of linear redundant codes to the problems of information protection in systems with limited resource capabilities. The obtained results provide scientific and engineering tools for monitoring and ensuring data integrity with the possibility of checking their authenticity after restoration in the event of an integrity violation. They also provide the conditions for creating promising and improving existing data storage systems. The method has the potential to be implemented in embedded systems, in particular, in IoT systems.

Keywords

Data integrity control, data recovery, hash function, matrix crypto transformations, redundant codes.

1. Introduction

Currently, users of many information systems are faced with the task of protecting the data processed by them. One of the measures to ensure data security in such systems is to protect their integrity [1]. Solving the problem of data integrity protection is particularly relevant in functioning data processing centers that include storage systems. These systems have different construction

ICST-2023: Information Control Systems & Technologies, September 21-23, 2023, Odesa, Ukraine.

EMAIL: smitsenko@gmail.com (S. Mitsenko); naumenko.serhii1122@vu.cdu.edu.ua (S. Naumenko); inna-roz@ukr.net (I. Rozlomii); a-ja@ukr.net (A. Yarmilko)
ORCID: 0000-0002-9582-7486 (S. Mitsenko); 0000-0002-6337-1605 (S. Naumenko); 0000-0001-5065-9004 (I. Rozlomii); 0000-0003-2062-2694 (A. Yarmilko)

structures and operating principles that provide for operation in the conditions of random errors and destructive influences of an intruder (unauthorized modification of data or decommissioning of part of the medium or individual cells, sectors). In addition to data integrity control, it is also necessary to ensure the recovery of data whose integrity has been violated.

There are various methods of solving the task of control and ensuring data integrity. Among them, schemes of hash functions are of the most significant interest. They are successfully used in the localization of defective blocks, but they are not without drawbacks. The main one is high redundancy when monitoring the integrity of sequences of small message blocks. Considering the mentioned shortcoming of the existing solutions, it is urgent to find ways to reduce the redundancy introduced for a given level of data security in the conditions of random errors and destructive influences of the attacker. In addition, despite the widespread use of hash functions, they must be more researched. Practical proposals for their use are mostly reduced to finding ways to increase their crypto resistance [2]. Proposals regarding the use of hash functions, which would allow reducing the introduced redundancy for a given level of data security, are very few (practically non-existent).

The research aims to investigate the efficacy of utilizing Hamming codes in conjunction with a novel hash technique for information protection and recovery. This involves the development of a robust and efficient method to encode and store data using Hamming codes while enhancing data integrity through a unique hash technique. The study will delve into the implementation, performance evaluation, and comparative analysis of the proposed approach against existing methods for information protection and recovery. Hamming codes have been widely used for error detection and correction, and hash techniques play a crucial role in verifying data authenticity. Combining these two concepts can potentially offer a more comprehensive approach to information protection and recovery. This research is relevant to various domains, including data storage, network communication, cybersecurity, and digital forensics, where data integrity and recovery are critical concerns. The proposed research introduces a novel amalgamation of Hamming codes and hash techniques, which has not been extensively explored in existing literature. While both Hamming codes and hash techniques are well-established individually, their synergistic application for information protection and recovery presents a novel approach. The integration of error-detection and correction capabilities of Hamming codes with the data verification strengths of hash techniques brings a unique angle to data integrity and recovery solutions. The research will pioneer the exploration of this combined method and provide insights into its effectiveness, thereby contributing to the advancement of information security methodologies.

2. Related works

In modern society's dependence on information technologies, any failure in operating information or communication systems can lead to losses. The causes and sources of potential threats to information security, risk assessment, and their classification have been the object of research for many years. They have been covered in scientific publications [3-4]. The results obtained for the current period make it possible to reasonably formulate requirements for means of information protection, in particular, its reliability and integrity control. A proven means of managing the reliability of information transmission is interference-resistant coding [5]. A distinction is made between error-detecting codes and error-correcting codes. C. Shannon pointed out the theoretical possibility of using interference-resistant coding to detect and correct errors as early as 1948.

The most popular self-monitoring and self-correcting linear block code is the Hamming code [6]. Today, Hamming codes are fundamental to building interference-resistant systems for coding and decoding digital data. The general principles of classical use of Hamming codes for single-bit error correction are presented in [7-8]. Research [9-10] in which modifications of Hamming codes and areas of their practical application are described show the considerable interest of the scientific community in Hamming codes. In [11], the performance of Hamming codes was evaluated using an artificial neural network. In work [12], formalization of error-correcting code (ECC) was developed using SSReflect extension of Coq proof-assistant. In addition to the famous Hamming codes, the authors investigated modern low-density parity-check codes (LDPC). In particular, in [13] the authors proposed efficient codes with low redundant code that can correct any combination of insertions and deletions in almost linear time.

A significant part of the attention of scientists is focused on the (7,4) Hamming code. In particular, [14-15] describe image hiding methods using the (7,4) Hamming code and corresponding operations with pixels. In addition, the Hamming code was used to control the integrity of blocks of electronic documents [16].

Available studies and publications provide a sufficient basis for generalizing ideas about the potential of tamper-resistant coding and hashing methods [17]. They confirm that the theory of interference-resistant coding, particularly of correcting Hamming codes, is adapted to solving data integrity problems in any application area.

3. Proposed technique

There are several types of error-correcting codes that are used for detecting and correcting errors in data transmission:

1. Hamming codes. They are among the most common error-correcting codes. They allow for the detection and correction of one or more single-bit errors in transmitted data. Hamming codes are used in many systems to ensure data integrity and reliability during transmission.
2. BCH codes (Bose-Chaudhuri-Hocquenghem). BCH codes are widely used error-correcting codes in many communication systems. They are capable of detecting and correcting multiple errors, including both single and multiple errors. BCH codes are typically used in systems with low noise levels or on channels with a high probability of errors.
3. Reed-Solomon codes. They are powerful error-correcting codes capable of detecting and correcting errors in data blocks of any size. They are widely used in digital communications, magnetic storage devices, supercomputers, and other systems where data transmission reliability is crucial.
4. Low-Density Parity Check (LDPC) codes. These codes are used in modern communication standards such as Wi-Fi, satellite communication, and mobile networks. They provide a high level of error correction while requiring minimal resource allocation. LDPC codes are based on low-density matrices, which determine their speed and high throughput capacity. They effectively handle large data volumes and noise in communication channels.
5. Turbo codes. They are among the most powerful error-correcting codes used in modern communication systems. They are based on the principle of iterative decoding, which ensures high error correction capability. Turbo codes are used in cellular communication, satellite systems, digital television, and other applications where data transmission reliability is crucial.

Each of these types of error-correcting codes has its own advantages and limitations, and the choice of a specific code depends on the specific application and the requirements for data transmission reliability (Table 1).

In this research, the choice of Hamming codes as a method for information protection and recovery is based on several key factors:

1. Error correction efficiency. Hamming codes have the property of detecting and correcting single-bit errors in transmitted data. This makes them effective for error protection in communication systems where data integrity is crucial.
2. Simplicity and low computational complexity. Hamming codes have a simple structure and are easily implemented. The algorithms for error detection and correction in Hamming codes are relatively simple, allowing for easy integration into resource-constrained systems such as embedded systems and IoT devices.
3. Flexibility. Hamming codes can be adapted to different requirements and application scenarios. They can have various levels of error correction, depending on the system's needs. This allows for the use of Hamming codes based on the specific level of protection required for the information resources.
4. Application in resource-constrained systems. Hamming codes demonstrate good efficiency in resource-constrained systems such as embedded systems and IoT devices. They do not require

large memory capacities or high computational power, making them suitable for use in such systems.

Table 1
Comparison of error-correcting codes properties

Code Type	Code Properties	Usage
Hamming codes	Detection and correction of single errors	Memory, data transmission networks, data storage systems
BCH codes	Detection and correction of multiple errors	Satellite communication, optical networks, digital television
Reed-Solomon codes	Detection and correction of errors in data blocks	Magnetic storage devices, digital communications
LDPC codes	High error correction capability with low matrix density	WiFi, satellite communication, mobile communication
Turbo codes	High error correction capability, iterative decoding	Cellular communication, satellite systems, digital television

Furthermore, Hamming codes have a solid mathematical foundation and find wide application in the theory of error correction and coding, providing additional confidence in their reliability and effectiveness. They can be successfully utilized in communication clusters of robotic systems and Industry 4.0 systems, where ensuring data integrity and the ability to detect and correct errors under resource constraints is crucial. The choice of Hamming codes offers a scientific and engineering toolset for securing and ensuring the reliability of information systems in the modern digital environment.

3.1. Analysis of existing methods of information integrity controls

The use of hashing algorithms to ensure the integrity of information has several advantages, including:

- relatively low redundancy;
- a small number of cryptographic transformations;
- the ability to control the length of the hash code.

Different methods of checking the integrity of information using hash functions differ in the composition and sequence of calculations. The generalized scheme of using hash functions is shown in Figure 1.

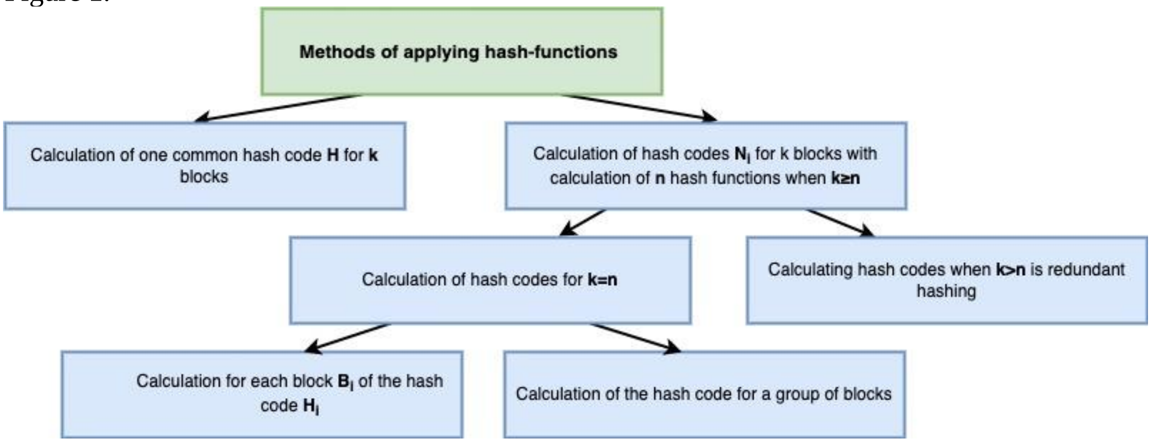

Methods of applying hash-functions
Calculation of one common hash code H for k blocks
Calculation of hash codes N$_i$ for k blocks with calculation of n hash functions when k≥n
Calculation of hash codes for k=n
Calculating hash codes when k>n is redundant hashing
Calculation for each block B$_i$ of the hash code H$_i$
Calculation of the hash code for a group of blocks

Figure 1: A generalized scheme for using hash functions

Let us take a closer look at each option for calculating the hash function. For this, we introduce the notation: h – hash function, B_i – information block, k – total number of blocks, n – number of hash functions.

1. Calculating one common hash code H for k blocks of data, represented by binary vectors $B_i (i = 1,2,...,k)$, result $h(B_1, B_2,..., B_k) = H$ (Figure 2). The checksum is obtained due to the execution of the hashing algorithm, which fully characterizes the entire set of blocks. Such a scheme, according to the properties of hash functions, allows you to control the integrity, but at the same time, there is no possibility of localizing a defective data block, which is represented by a binary vector B_i.

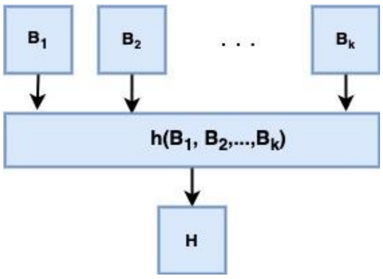


Figure 2: Calculating a common hash function for k blocks

In the case of $k \geq n$ – for each data block represented by a binary vector B_i a hash value is calculated H_i, Figure 3.

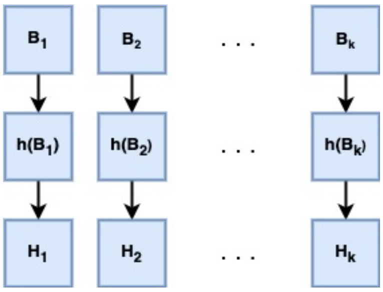


Figure 3: Calculating a hash function for each block

A method of applying a fully connected hashing network, in which each hash code H_i is calculated from the entire set of data, represented by binary vectors B_i :

$$h(B_{i1} \| B_{i2} \| ... \| B_{ik}), \qquad (1)$$

where || – concatenation operation (Figure 4).

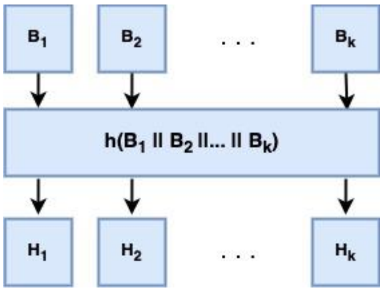


Figure 4: Calculating a hash function for groups of blocks

In this method, the sequence of data blocks, represented by binary vectors, is of great importance B_i, and takes part in concatenation. According to Figure 4, firstly the binary vector is concatenated B_i, the number of which coincides with the number of the hash code being calculated H_i (due to this, their values will differ).

2. Ways of using hash functions at $k > n$ are of greatest interest because they allow integrity control and detection of defective data blocks. Methods are presented in [16-17] that allow you to localize errors and repair damaged blocks. The methods are based on cryptographic methods for calculating the hash function based on matrix crypto transformation operations. Aggregation of hash function calculation methods based on matrix primitives and linear systems of hash codes built according to principles similar to the rules for constructing linear redundant Hamming codes.

3.2. Method development

The introduction of redundant information into the information transmitted by the network provides the possibility of detecting and correcting errors on the side of the recipient of the message. The mathematical theory of building redundant (interference-resistant) codes now has great achievements. However, there is a big gap between the level of theoretical achievements of the theory of interference-resistant coding and the level of results of practical use of this theory.

The introduction of redundancy makes it possible to detect and correct errors in the information that is transmitted and can be changed during transmission. There are codes that detect errors and correcting codes that, in addition to detecting an error, correct it. The easiest ways to detect errors are checksumming and parity checking. However, they are not reliable enough, especially when a large number of errors occur. Since whole fragments can be falsified in information messages, such mechanisms cannot fully solve the problem of their forgery. As you know, the history of the emergence and development of the theory and practice of interference-resistant coding to correct errors and thereby ensure the reliability of transmitted data begins with the works of Shannon [18]. However, Shannon did not show how to build tamper-resistant codes but only proved their existence. Not long after, Hamming developed the theory of linear block codes. Hamming introduced and defined the basic parameters of block codes and developed encoding and decoding devices for his codes.

The proposed method of data integrity control uses a system of hash codes. They are built according to rules similar to the rules for building linear redundant codes. In this case, according to the rules for constructing Hamming codes. Such a system is a linear hash code system [19]. It is represented by a set of hash codes obtained using a standard procedure for implementing a hash function from a set of message blocks in the order determined by a particular block selection procedure based on the mathematical apparatus of linear algebra. For the standard procedure of implementing a hash function, in particular, methods of calculating a hash function based on matrix primitives can be used, presented in [20].

To control and ensure message integrity B, it must be represented as a set of fixed-length blocks $B = \{B_1, B_2, ..., B_n\}$. Blocks are interpreted as a sequence of n information blocks to which s control (additional) blocks are added in the amount necessary to protect data integrity. As a result, a code sequence will be obtained $(n, n+s)$, Figure 5.

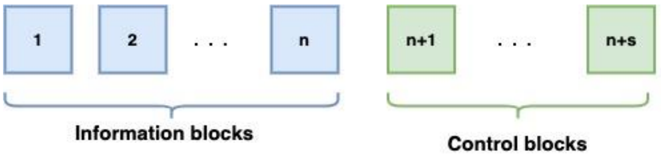


Figure 5: Scheme of code representation (n, n+s)

The addition of control blocks is performed according to the rules for building redundant codes, depending on the need for corrective properties of the received code (Figure 6).

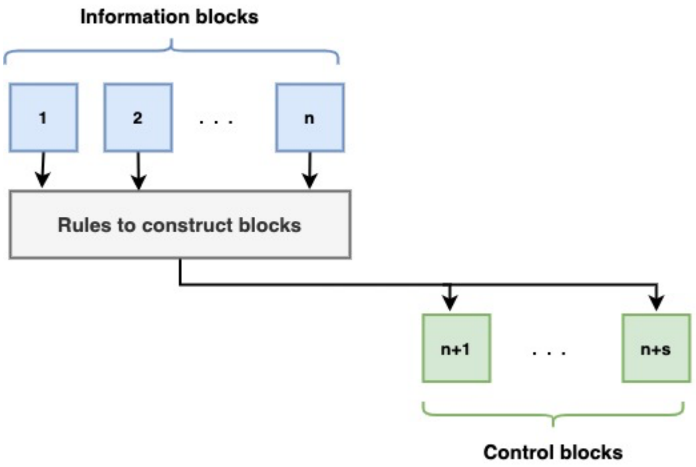


Figure 6: Scheme of redundant code formation

To evaluate the correcting ability of codes, Hamming introduced the codename d and minimal codename d_0 distance and showed their dependence on code length, and introduced redundancy [21]. Hamming proved that the minimum code distance characterizes the correcting properties of an interference-resistant code. It was proved [22] that if two code sequences differ from each other in t $(t \geq 1)$ positions (bits, characters), and will differ from all other code sequences of this code set in more than t positions, then to correct t errors it is necessary to ensure the minimum code distance:

$$d_0 \geq 2 * t + 1. \qquad (2)$$

If condition (2) is satisfied, then the fault-tolerant code can be guaranteed to correct the following number of erroneous characters:

$$t_{correct} \leq \frac{d_0 - 1}{2} \qquad (3)$$

or detect $t_{\det ect} \leq d_0 - 1$ false binary characters.

So, to build (n, n+s)-code, let's use formulas (2-3). The resulting (n, n+s)-code will be used to restore damaged information blocks.

Let us dwell more on the fact that any $G_{(n,k)}$ Hamming code can be given in its general form by the generating matrix:

$$G_{(n,k)} = \begin{vmatrix} 1\,0\,0\,0 \ldots 0 \; b_{11}\; b_{12}\; b_{13}\; b_{14} \ldots b_{1r} \\ 0\,1\,0\,0 \;\ldots 0\; b_{21}\; b_{22}\; b_{23}\; b_{24} \ldots b_{2r} \\ 0\,0\,1\,0 \;\ldots 0\; b_{31}\; b_{32}\; b_{33}\; b_{34} \ldots b_{3r} \\ 0\,0\,0\,1 \;\ldots 0\; b_{41}\; b_{42}\; b_{43}\; b_{34} \ldots b_{4r} \\ .\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;. \\ .\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;.\;\;. \\ 0\,0\,0\,0 \;\ldots 1\; b_{k1}\; b_{k2}\; b_{k3}\; b_{k4} \ldots b_{kr} \end{vmatrix} \qquad (4)$$

To determine the values of the verification elements of the right part of the matrix, it is necessary to proceed from the main properties of systematic codes.

Since each row of the unit matrix $k \times k$ has only one unit, the weight of each row of the assigned matrix must not be less than $d - 1$, namely, the mod2 of the two rows must not be less than $d - 2$ for guaranteed single error correction. In addition, the combinations of the right-hand side of the matrix must be linearly independent.

Since it is considering problems of integrity violations in message blocks of the communication system, we denote the message as $\vec{A} = (\vec{a}_1, \vec{a}_2, \vec{a}_3, ..., \vec{a}_n)$, where $\vec{a}_1, \vec{a}_2, \vec{a}_3, ..., \vec{a}_n$ – set of corresponding binary vectors (blocks of information of arbitrary size). By the set $\vec{F} = (\vec{f}_1, \vec{f}_2, \vec{f}_3, ..., \vec{f}_n)$ denote the values of hash functions of a fixed size, calculated by $f_i = h(a_i)$, where $i \in [1, n]$.

A set of possible block hashing schemes $a_1, a_2, a_3, ..., a_n$ can be represented in the form of a binary matrix:

$$F = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{pmatrix}, \qquad (5)$$

where each row corresponds to a defined hashing scheme. The conditions for forming the matrix rows are described in [17]. Given the properties of the generating matrix, the rules for building linear codes make it possible to build systems of hash codes. A system of hash codes is a set of hash codes obtained by implementing any algorithm for calculating a hash function in the order determined by a particular procedure for selecting records (blocks of information) based on the mathematical apparatus of linear algebra.

3.3. Algorithm for building hash codes

Hashing of the original block of information can be represented as an expression $(\vec{a}_1\, \vec{a}_2 ... \vec{a}_{n+1}\, \vec{a}_{n+l}) \rightarrow (\vec{a}_1\, \vec{a}_2 ... \vec{a}_{n+1}\, \vec{a}_{n+l}\ \vec{f}_{n+l+1}\ \vec{f}_{n+r})$, where → – special multidimensional non-commutative hashing operation.

Then, as a result of hashing, the secured block will look like this:

$$(\vec{a}_1\, \vec{a}_2 ... \vec{a}_{n+1}\, \vec{a}_{n+l}) \otimes \begin{vmatrix} 1\,0\,0\,0 \ldots 0 \; b_{11}\, b_{12}\, b_{13}\, b_{14} \ldots b_{1r} \\ 0\,1\,0\,0 \ldots 0 \; b_{21}\, b_{22}\, b_{23}\, b_{24} \ldots b_{2r} \\ 0\,0\,1\,0 \ldots 0 \; b_{31}\, b_{32}\, b_{33}\, b_{34} \ldots b_{3r} \\ 0\,0\,0\,1 \ldots 0 \; b_{41}\, b_{42}\, b_{43}\, b_{34} \ldots b_{4r} \\ .\ .\ .\ .\ .\ .\ .\ .\ .\ . \\ .\ .\ .\ .\ .\ .\ .\ .\ .\ . \\ 0\,0\,0\,0 \ldots 1 \; b_{k1}\, b_{k2}\, b_{k3}\, b_{k4} \ldots b_{kr} \end{vmatrix} = (\vec{a}_1\, \vec{a}_2 ... \vec{a}_{n+1}\, \vec{a}_{n+l}\ \vec{f}_{n+l+1}\ \vec{f}_{n+r}), \qquad (6)$$

where $\vec{f}_{n+r} = h(b_0 \vec{a}_n \,||\, b_1 \vec{a}_{n+1} \,||\, ... \,||\, b_l \vec{a}_{n+l}),\ b_l \in \{1, 0\}$ or $\vec{A}_v \otimes G_{(n,k)} = (\vec{a}_1\, \vec{a}_2 ... \vec{a}_{n+1}\, \vec{a}_{n+l}\ \vec{f}_{n+l+1}\ \vec{f}_{n+r})$, symbol $\otimes$ – a special multidimensional non-commutative operation of hashing information blocks of an electronic document. The algorithm for building redundant hash codes is shown in Figure 7.

The syndrome concept is used to control the integrity of information in the theory of linear codes. A syndrome in coding theory means a set of signs characteristic of a particular phenomenon. The syndrome of a vector that can have errors makes it possible to recognize the most likely nature of these errors.

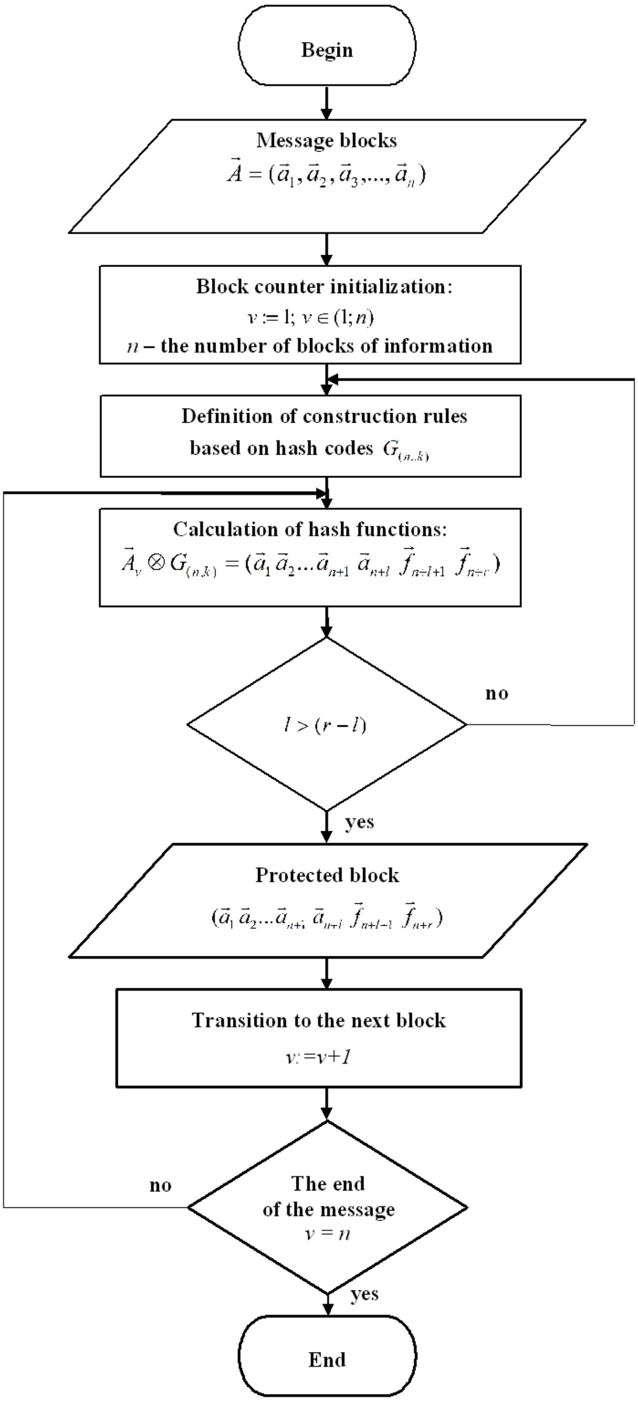
By an error in the protected block $(\vec{a}_1\, \vec{a}_2 ... \vec{a}_{n+1}\, \vec{a}_{n+l}\ \vec{f}_{n+l+1}\ \vec{f}_{n+r})$, we will understand the result of the discrepancy of the binary vector with the result obtained as a result of the syndrome check.

Checking the integrity of data in blocks of information includes the following steps:

- we have a block of data at the entrance $(\vec{a}^*{}_1\, \vec{a}^*{}_2 ... \vec{a}^*{}_{n+1}\, \vec{a}^*{}_{n+l}\ \vec{f}^*{}_{n+l+1}\ \vec{f}^*{}_{n+r})$, which is checked for integrity, a block hashing operation must be performed;
- calculate the syndrome that corresponds to the value of the predicate:

$$P(\vec{a}_n)=\begin{cases}1, if \ \vec{f}_n^* = \vec{f}_n; \\ 0, if \ \vec{f}_n^* \neq \vec{f}_n.\end{cases}, \quad (7)$$

● according to the table of syndromes, it is necessary to correct errors in the blocks of information.



Begin
Message blocks
$\vec{A}=(\vec{a}_1,\vec{a}_2,\vec{a}_3,...,\vec{a}_n)$
Block counter initialization:
$v:=1;\ v\in(1;n)$
n – the number of blocks of information
Definition of construction rules based on hash codes $G_{(n,k)}$
Calculation of hash functions:
$\vec{A}_v\otimes G_{(n,k)}=(\vec{a}_1\vec{a}_2...\vec{a}_{n+1}\ \vec{a}_{n+l}\ \vec{f}_{n+l+1}\ \vec{f}_{n+r})$
$l>(r-l)$
no
yes
Protected block
$(\vec{a}_1\vec{a}_2...\vec{a}_{n+1}\ \vec{a}_{n+l}\ \vec{f}_{n+l-1}\ \vec{f}_{n+r})$
Transition to the next block
$v:=v+1$
no
The end of the message
$v=n$
yes
End

Figure 7: Algorithm for calculating hash functions to ensure message integrity

The block diagram of the integrity check algorithm in blocks of information is shown in Figure 8.

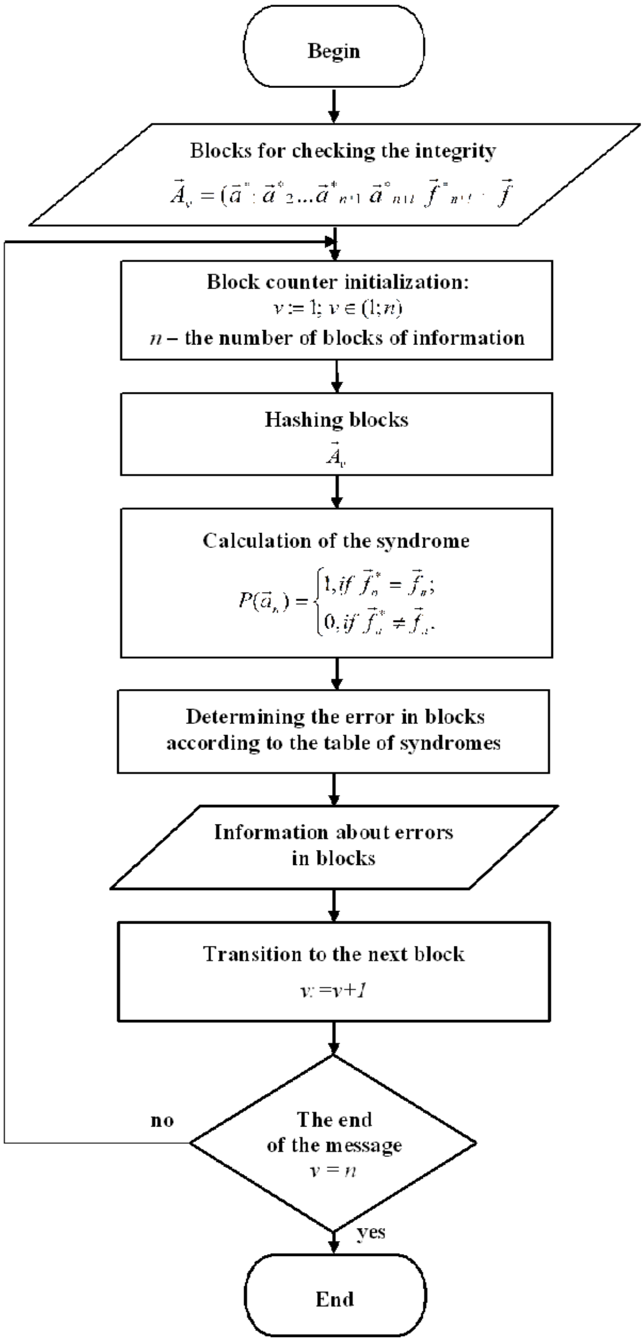

Begin
Blocks for checking the integrity
$\vec{A}_v = (\vec{a}^*_1\ \vec{a}^*_2 \ldots \vec{a}^*_{n-1}\ \vec{a}^*_{n}\ \vec{f}^*_{n-1} \ldots \vec{f}$
Block counter initialization:
$v := 1;\ v \in (1;n)$
n – the number of blocks of information
Hashing blocks
$\vec{A}_v$
Calculation of the syndrome
$P(\vec{a}_n) = \begin{cases} 1, if\ \vec{f}^*_n = \vec{f}_n; \\ 0, if\ \vec{f}^*_n \neq \vec{f}_n. \end{cases}$
Determining the error in blocks according to the table of syndromes
Information about errors in blocks
Transition to the next block
$v := v+1$
The end of the message
$v = n$
no
yes
End

Figure 8: Error detection algorithm in blocks of information

3.4. Data integrity control based on the rules for constructing linear hash codes

To construct an interference-resistant code $G_{(9,4)}$ message $\vec{A} = (\vec{a}_1, \vec{a}_2, \vec{a}_3, \vec{a}_4)$ we will use the theory of linearly independent vectors, which is used in vector theory to construct a Hamming code (9,4):

$$(\vec{a}_1, \vec{a}_2, \vec{a}_3, \vec{a}_4) \otimes \left|\begin{array}{c|c} 1000 & 00010 \\ 0100 & 00101 \\ 0010 & 11011 \\ 0001 & 11100 \end{array}\right| = (\vec{a}_1, \vec{a}_2, \vec{a}_3, \vec{a}_4, \vec{f}_1, \vec{f}_2, \vec{f}_3, \vec{f}_4, \vec{f}_5) \cdot$$

The resulting scheme for building an interference-resistant code is presented in Figure 9.

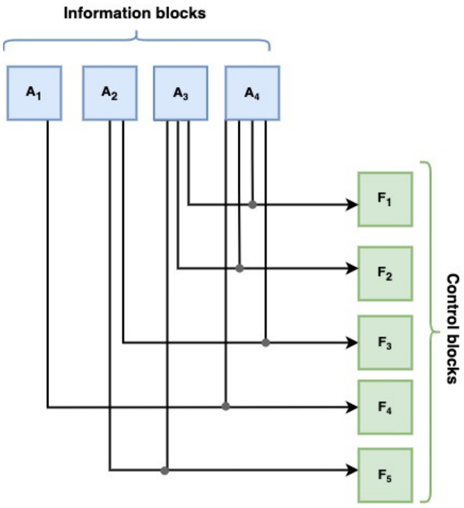
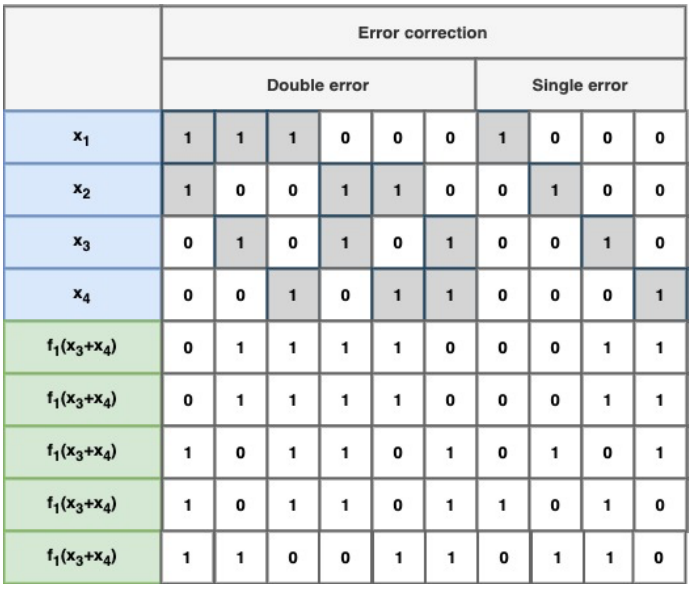


Figure 9: Rules for the formation of control blocks

To control the integrity of the protected data block according to (6) $(\vec{a}^*{}_1,\vec{a}^*{}_2,\vec{a}^*{}_3,\vec{a}^*{}_4,\vec{f}^*{}_1,\vec{f}^*{}_2,\vec{f}^*{}_3,\vec{f}^*{}_4,\vec{f}^*{}_5)$ the syndrome is calculated $\vec{F}=(f_1,f_2,f_3,f_4,f_5)$, what corresponds to the predicate (7). Figure 10 shows the results of integrity violations in a protected data block determined by syndromes $(\vec{a}_1,\vec{a}_2,\vec{a}_3,\vec{a}_4,\vec{f}_1,\vec{f}_2,\vec{f}_3,\vec{f}_4,\vec{f}_5)$.



	Error correction									
	Double error						Single error			
x_1	1	1	1	0	0	0	1	0	0	0
x_2	1	0	0	1	1	0	0	1	0	0
x_3	0	1	0	1	0	1	0	0	1	0
x_4	0	0	1	0	1	1	0	0	0	1
$f_1(x_3+x_4)$	0	1	1	1	1	0	0	0	1	1
$f_1(x_3+x_4)$	0	1	1	1	1	0	0	0	1	1
$f_1(x_3+x_4)$	1	0	1	1	0	1	0	1	0	1
$f_1(x_3+x_4)$	1	0	1	1	0	1	1	0	1	0
$f_1(x_3+x_4)$	1	1	0	0	1	1	0	1	1	0

Figure 10: Error correction in message blocks (falsified blocks are highlighted in gray)

4. Discussions

Developed linear hash codes, built by analogy with Hamming codes, allowing correcting errors in message blocks. Still, the number of mistakes that can be rectified (corrective property of the code) depends on the size of the redundant code (control blocks of information). It is necessary to rationally choose the necessary redundant code to ensure, on the one hand, the required reliability of the information and, on the other hand, to avoid burdening the communication channels with a large amount of redundant data. In other words, it is necessary to ensure the integrity of information with a minimum amount of redundant code.

The proposed solutions can be applied in traditional information systems and implement intelligent procedures with hybrid human-machine intelligence. An example is modern concepts of creating production systems (in particular, Industry 4.0 and Industry 5.0), which focus on using artificial intelligence as a safe, reliable, and responsible component of a single human-machine functional and communication space. The intellectualization of human-machine interaction, the cooperative nature of the activity of intelligent agents, and their interaction in an available open slot, undoubtedly exacerbate such systems' efficiency, safety, and predictability. The severity of these problems, directly related to the state of communications, will become more acute in connection with the exit of such technologies from the category of unique and experimental projects to the variety of mass utilitarian and applied applications. Since the typical solution for creating cooperative system modules of this type is their reproduction on the platform of embedded systems, the functioning of security subsystems will be directly affected by the presence of constrained devices in their composition and battery life limitations. Therefore, the problem of security in such systems must be solved comprehensively. Priority is given to methods and algorithms that are appropriate in terms of basic functionality and economical in terms of the resources of the technical platform. In general, the practical implementation of proper software tools should increase trust between participants in communication processes due to the possibility of identifying and restoring damaged fragments of messages in the information flow. Because of the above, the proposed method of ensuring the integrity of notes follows the general requirements for such means.

There are several ways to extend and modify the proposed hashing method based on Hamming codes for detecting and correcting faulty information blocks exchanged among participants in a communication cluster of a robotic system. Some of these possibilities include:

1. Using hybrid methods. It is possible to combine Hamming codes with other error detection and correction methods, such as BCH (Bose-Chaudhuri-Hocquenghem) codes or Reed-Solomon codes. This will allow the creation of a hybrid system that combines the advantages of different coding methods and ensures a high level of error correction and error recovery.
2. Using optimized algorithms. It is possible to conduct research and development of optimized algorithms for computing hash codes based on Hamming codes. This may involve the use of fast computation methods, memory optimization, and other techniques to enhance the performance of the hashing method.
3. Expanding the application scope of the method. The hashing method based on Hamming codes can be extended for use in various domains of communication clusters in robotic systems. For example, exploring the potential of this method for data protection in unmanned vehicles, where ensuring reliable transmission and data integrity between the propulsion mechanism and control systems is crucial, would be worthwhile. Additionally, attention should be given to the possibility of specializing the method for Industry 4.0 applications, where robotic systems and other "smart" devices interact in a manufacturing environment. Expanding the application scope of the method can also include the Internet of Things (IoT), where numerous devices are connected to the network and exchange data. The optimal utilization of Hamming codes for detecting and correcting defective data blocks can provide an additional level of protection and reliability in these domains, facilitating data recovery and preventing the transmission of erroneous information.

5. Conclusion

The article proposes a cryptographic hashing method based on Hamming codes for information protection and recovery. Using the mathematical apparatus of the theory of vector systems, an algorithm for building linear hash codes was developed to ensure data integrity in information systems. The rules for building hash codes are affected by the given (or necessary) level of security of information resources. The redundancy of control information depends on the need for curative properties.

It is shown that the rules for constructing linear hash code systems are similar to the rules for constructing Hamming codes. Thus, the well-developed theory of linear redundant codes can be used in the new field of constructing linear hash code systems.

The main advantage of the proposed method is the implementation of information integrity control and defect correction for a given level of security with minimal redundancy and the possibility of localizing integrity violations and correcting a given number of errors.

The obtained results provide a scientific and engineering toolkit for monitoring and ensuring data integrity with the possibility of checking their authenticity after restoration in case of integrity violation and provide the conditions for creating promising and improving existing data storage systems.

6. References

[1] M. E. Whitman, H. J. Mattord, Principles of information security, 7th. ed., Cengage Learning, Boston, MA, 2021.

[2] R. Alguliyev, Y. Imamverdiyev, L. Sukhostat, Cyber-physical systems and their security issues, Computers in Industry 100 (2018) 212-223.

[3] C. K. Yee, M. F. Zolkipli,Review on Confidentiality, Integrity and Availability in Information Security, Journal of ICT in Education, 8 2 (2021) 34-42.

[4] M. Nieles, K. Dempsey, V. Y. Pillitteri, An introduction to information security, NIST special publication, 800 12 (2017) 101.

[5] J. Sima, J. Bruck, On optimal k-deletion correcting codes, IEEE Transactions on Information Theory, 67 6 (2020) 3360-3375.

[6] J. Van Wonterghem, A. Alloum, J. J. Boutros, M. Moeneclaey, Performance comparison of short-length error-correcting codes, in: 2016 Symposium on Communications and Vehicular Technologies, SCVT, IEEE, 2016, pp. 1-6. doi.org/10.1109/scvt.2016.7797660.

[7] A. K. Singh, Error detection and correction by hamming code, in: International Conference on Global Trends in Signal Processing, Information Computing and Communication, ICGTSPICC, IEEE, 2016, pp. 35-37.

[8] P. Kumar, A. K. Ahuja, R. Chakka, BCH/hamming/cyclic coding techniques: Comparison of PAPR-reduction performance in OFDM systems, in: International Conference on Intelligent Computing and Applications, ICICA, Springer, 2018, pp. 557-566.

[9] U. Martínez-Peñas, Hamming and simplex codes for the sum-rank metric, Designs, Codes and Cryptography, 88 8 (2020) 1521-1539.

[10] N. Sridevi, K. Jamal, K. Mannem, Implementation of error correction techniques in memory applications, in: 5th International Conference on Computing Methodologies and Communication, ICCMC, IEEE, 2021, pp. 586-595.

[11] A. C. Vaz, C. G. Nayak, D. Nayak, Hamming code performance evaluation using artificial neural network decoder, in: 15th International Conference on Engineering of Modern Electric Systems, EMES, IEEE, 2019, pp. 37-40.

[12] R. Affeldt, J. Garrigue, Formalization of error-correcting codes: from Hamming to modern coding theory, in: Interactive Theorem Proving: 6th International Conference, ITP 2015, Nanjing, China, August 24-27, Springer International Publishing, 2015, pp. 17-33.

[13] J. Brakensiek, V. Guruswami, S. Zbarsky, Efficient low-redundancy codes for correcting multiple deletions, IEEE Transactions on Information Theory, 64 5 (2017) 3403-3410.

[14] Z. Cao, Z. Yin, H. Hu, X. Gao, L. Wang, High capacity data hiding scheme based on (7, 4) Hamming code, SpringerPlus, 5 1 (2016) 1-13. doi: 10.1186/s40064-016-1818-0.

[15] B. Jana, D. Giri, S. K. Mondal, Partial reversible data hiding scheme using (7, 4) hamming code, Multimedia Tools and Applications, 76 (2017) 21691-21706.

[16] I. A. Rozlomii, V. N. Rudnitsky, E. S. Alekseeva, Using of hash function to identify counterfeit fragments of electronic document, Wschodnioeuropejskie Czasopismo Naukowe (East European Scientific Journal), 3 19 (2017) 68-72.

[17] A. Yarmilko, I. Rozlomii, H. Kosenyuk, Hash method for information stream’s safety in dynamic cooperative production system, in: S. Shkarlet et al. (Eds): Mathematical Modeling and Simulation of Systems, volume 344 of Lecture Notes in Networks and Systems, Springer, Cham, 2022, pp. 173-183. doi.org/10.1007/978-3-030-89902-8_14.

[18] J. Zhang, K. Feng, Relative generalized Hamming weights of cyclic codes, Finite Fields and Their Applications 50 (2018) 338-355.

[19] X. Li, Q. Yue, The Hamming distances of repeated-root cyclic codes of length 5ps, Discrete Applied Mathematics 284 (2020) 29-41.

[20] I. O. Rozlomii, Methods for calculating the hash function of an electronic document based on matrix cryptographic transformations, Bulletin of CSTU. Engineering sciences 4 (2016) 88-94.

[21] M. Shi, F. Özbudak, P. Solé, Geometric approach to b-symbol Hamming weights of cyclic codes, IEEE Transactions on Information Theory, 67 6 (2021) 3735-3751.

[22] W. Rurik, A. Mazumdar, Hamming codes as error-reducing codes, in: IEEE Information Theory Workshop, ITW, IEEE, 2016, pp. 404-408.