Data encryption method based on the redundant residue number system

Vasyl Yatskiv [1,†], Elena Nyemkova [2,†], Serhii Kulyna [1,*,†], Halyna Kulyna [1,†] and Stepan Ivasiev [1,†].

[1] *West Ukrainian National University, 11 Lvivska str., Ternopil, 46009, Ukraine*
[2] *Lviv Polytechnic National University, 12 Bandery str., Lviv, 79000, Ukraine*

Abstract

The subject of the article is the process of data encryption for building systems of distributed secure data storage. The aim of the work is to develop a method of data encryption based on Redundant Residue Number System (RRNS) and evaluate its cryptographic strength. The tasks include analyzing the existing method of data encryption using RRNS and investigating its cryptographic strength; constructing a structural diagram of the encryption process based on RRNS and shuffling method based on the M-sequence key; justifying the use of the conditions mentioned in the work to reduce the bit width of the used modules. The methods used in the research are: RRNS-based digital information protection method; methods of data transformation and processing in RRNS; methods for evaluating the cryptographic strength of encryption algorithms. The following results were obtained: an algorithm for data encryption based on RRNS was proposed. The existing method of data encryption based on RRNS is investigated and an improved method of data encryption based on RRNS and the shuffling method based on the M-sequence key is proposed. A formula was derived to estimate cryptographic strength in RRNS-based data encryption, allowing calculation of module number and bit width for protection comparable to AES 128 against brute-force attacks. Suggested was a minimum message length to decrease required module bit width for specified protection levels. Cryptographic strength comparisons were made between classical RRNS encryption and proposed RRNS with shuffling method using M-sequence key. Findings: Novelty lies in proposing RRNS-based data encryption employing an additional key for M-sequence key-based shuffling, ensuring heightened cryptographic security. A structural diagram for RRNS data encryption was devised, alongside a formula for estimating cryptographic strength in the proposed RRNS method. The conducted research showed that the proposed method with a module bit rate of 8-16 bits provides 11-15 times higher cryptographic strength compared to the classical method of encryption in RRNS, and with a module bit rate of 8-16 bits, and with a module bit rate of 96-128 bits, on average, it is 3 times higher which makes it suitable for building secure distributed data storage systems.

Keywords

redundant residue number system, distributed data storage, protected data storage systems, data recovery. cipher text, difficulty rating

IntelITSIS'2024: 5th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 28, 2024, Khmelnytskyi, Ukraine

[*] Corresponding author.
[†] These authors contributed equally.

jazkiv@ukr.net (V. Yatskiv); cyberlbi12@gmail.com (E. Nyemkova); sersks@gmail.com (S. Kulyna); gkulyna@gmail.com (H. Kulyna); stepan.ivasiev@gmail.com (S. Ivasiev).
0000-0001-9778-6625 (V. Yatskiv); 0000-0003-0690-2657 (E. Nyemkova); 0000-0002-6162-9457 (S. Kulyna); 0000-0001-7035-315X (H. Kulyna); 0000-0003-2243-5956 (S. Ivasiev).

1. Introduction

To backup data, distributed systems can be used that make it difficult for intruders to access data. Distributed data storage systems allow splitting data into fragments which can be located on physically remote devices or cloud services that complicate physical access to information. In the case when an attacker gets access to part of the data or one of the data storage devices, the use of encryption, in particular, the block type of encryption and maintaining sequential blocks on different devices allows for additional data protection and makes it impossible to decrypt it.

There are a number of methods for splitting data into fragments; one of them is the use of the Residue Number System. According to RRNS, data is split into fragments due to the predetermined modules, which reduces the bit-width of the number and increases the speed of performing arithmetic operations, especially when processing long-length messages, for example, during hardware implementation of cryptographic algorithms [1]. The inverse transform is performed on all residues simultaneously, so in case of interception of some of them by an attacker, recovery is impossible.

2. Related works

Despite a large number of existing data protection methods and algorithms, the number of cyber threats to intelligent and computer networks is increasing [2]. As a result, public attention to cryptography is growing. Most of the modern scientific and technical problems that need to be solved are related to data protection [3].

Data encryption methods are developed with the aim of detecting modifications along with ensuring security to prevent an intruder from getting access to data or modifying it when transmitting [4].

Common data protection methods include the use of long symmetric and asymmetric keys [5], as well as various key generation algorithm functions [6]. The use of new types of quantum generators of pseudorandom signals [7] makes it possible to ensure the true randomness of the generated keys. Application of multilevel encryption methods [8] can be efficient, but such solutions lead to a significant decrease in the performance of cryptographic algorithms [9].

In turn, the use of non-positional number systems, an example of which is the RRNS [10], makes it difficult for an attacker to decode data, and the use of simple algorithms allows the implementation of multilevel encryption methods [11], which, in combination with other common solutions, significantly improves data protection [12]. In [13], a special system of modules for building correction codes based on the RRNS was considered. Their use is one of the ways to increase the reliability and security of data storage systems [14] and distributed storage of residue files allows increasing trust in cloud storage providers [15].

The purpose of the research is to develop a data encryption method based on the RRNS and estimate its cryptographic strength.

3. Research on encryption in RRNS

3.1. Data encryption algorithm in RRNS

Data conversion in RRNS is carried out according to the following formula:

$$x_i = X \bmod p_i , \tag{1}$$

where $(x_1, x_2, ..., x_i, ... x_n)$ is the sequence of residues; X – data given in the positional number system; p_i - relatively prime modules.

The received residues x_i are written into files and stored in the network or distributed data storage. Data conversion from the RRNS into the positional number system is carried [14]:

$$X = \left(\sum_{i=1}^{n} x_i \cdot b_i \right) \bmod P , \tag{2}$$

where P is the total range of the system and determined by the following formula:

$$P = \prod_{i=1}^{n} p_i ;$$

b_i are the base numbers of the RRNS, which are determined according to formula:

$$b_i = \frac{P}{p_i} \cdot m_i \equiv 1 \bmod p_i ,$$

where m_i – is a set of coefficients that ensure the conversion orthogonally and satisfy the condition $0<m_i<p_i$.

To detect errors in the RRNS, an extended system of modules (p_1, p_2, ..., p_i, ..., p_k, ..., p_n) is used, where k is the number of modules of the working range.

To detect errors, the following concept of the working range is introduced:

$$H = \prod_{i=1}^{k} p_i .$$

If an error occurs in the residue of one of the modules (x_1, x_2, ..., x_i^*, ..., p_n) the value X^* obtained as a result of the inverse transform will be outside the working range, i.e. $H<X^*$.

Data encryption algorithm in the RRNS consists of the following steps:

1. Choosing the size of the encryption block.
2. Selecting the number of relatively prime modules.
3. Selecting the value of relatively prime modules.
4. Converting the data block into decimal format.
5. Converting the data block into the RRNS according to formula (1).
6. Coding residues into the optimal volume format.
7. Recording the residues xi into n files to store them on distributed media.

The data encryption key in the RRNS is presented by the values of modules pi and their number n.

3.2. Data encryption conditions in RRNS

It should be noted that conducting theoretical research, we are not restricted to any conditions when selecting modules, but they must be taken into account during implementation. The first condition is that the smallest addressable memory cell is a byte.

Accordingly, the values of modules should be selected so that the working range meets the following condition:

$$H \geq 2^{8*c} + 1,$$

where H – is the working range; c – is an integer that determines the amount of stored data.

It is also necessary to take into account that relatively prime numbers are used as modules in RRNS, the values of which affect the size of the working range [16].

For example, when c=2, the value of the working range must satisfy the condition H≥65537. To fulfill this condition, one of the options is a set of modules: {41, 43, 47}, with the value H=82861, and another one is as follows: {3, 5, 7, 11, 13, 17}, H= 255255.

As it is seen, both values satisfy the condition, however, in the second case, the number of modules is 2 times larger and, accordingly, the number of memory cells for storing the residues is also required to be 2 times larger.

Due to this comparison, it is possible to derive the condition for selecting the optimal set of modules, namely, the dependence on the number and bit-size of modules.

We cannot unlimitedly increase the number of modules, because it complicates the operations of writing and reading data from memory, and the speed of performing arithmetic operations with modules of the same bit-size takes the same amount of time [17].

When comparing sets of modules {41, 43, 47} and {3, 5, 7, 11, 13, 17} which have the same bit-size, i.e., 1 byte and 2 bytes of the working range, the 1st set of modules is 2 times more effective, since its residues are 3 bytes when calculating and the 2nd set of residues is 6 bytes.

When attempting unauthorized data decryption (2), the intruder needs to determine the number n and the value of modules p_i, which have been used to encrypt data in RRNS.

Thus, the problem of estimating the cryptographic strength of data encryption based on the RRNS is still relevant.

According to the asymptotic distribution of prime numbers, their amount in the interval from 0 to some q is approximately determined by the following formula:

$$\pi(q) = \frac{q}{\ln q - 1{,}07}. \tag{3}$$

When using s-bit numbers, the range q can be represented as $q=2^s$, and formula 3 is as:

$$\pi(s) = \frac{2^s}{(s \cdot \ln 2) - 1{,}07}. \tag{4}$$

When selecting a system of n modules, the approximate number of options can be determined by the following formula:

$$L(n,s) = \prod_{\pi(s)-n}^{\pi(s)} \pi(s). \tag{5}$$

For small values of s, this notation is more accurate. However, under the condition of using modules with the bit-size s≥16, formula 5 can be represented as follows:

$$L(n,k)=\left(\frac{2^s}{(s\cdot\ln2)-1{,}07}\right)^n. \quad (6)$$

Accordingly, the total complexity of direct and inverse transforms based on the RRNS is determined due to the following formula [17]:

$$O_{RRNS}(n^2\cdot2^{2\cdot s}). \quad (7)$$

And the total computational complexity of the cryptanalysis algorithm based on the RRNS is calculated as follows:

$$O\left(\left(\frac{2^s}{s\cdot\ln2}\right)^n\cdot n^2\cdot2^{2\cdot s}\right). \quad (8)$$

When determining the complexity of the cryptanalysis algorithm, it is advisable to round down the power values, which makes it possible to compare the algorithm strength with common solutions based on modern symmetric algorithms [18].

3.3. Research the complexity of the cryptanalysis algorithm

The smallest acceptable number of modules in a redundant RRNS should consist of two information modules and one check modulus, i.e., n=3.

Taking into account this condition, the total computational complexity of the cryptanalysis algorithm with the specified set of modules (formula 8) is determined as:

$$O\left(\left(\frac{2^s}{s\cdot\ln2}\right)^3\cdot9\cdot2^{2\cdot s}\right).$$

The calculation results of the dependence of the cryptanalysis algorithm total complexity on the bit-size of the residues with two information modules and one check modulus are presented in Table 1.

As it can be seen from the above calculations (Table 1), when the bit-size of the modulus increases by 8, the bit-size of the cryptanalysis algorithm computational complexity Q(n, s) increases on average by 24 orders, which shows a direct linear dependence, and the overall system efficiency gradually decreases, since the percentage growth falls. To study the dependence of the computational complexity of the cryptanalysis algorithm on the bit-size, $n\in[4,8]$ and $s\in[8,128]$ are substituted into formula 6.

The results of this study show that the dependence of the computational complexity of the cryptanalysis algorithm O(n, s) does not change and stays linear when the bit-size of residues increases.

In general, an increase in the bit-size of modules leads to an increase in the cryptanalysis complexity by approximately 2 times when the number of bits increases from 8 to 16, with a further decrease in the efficiency of each subsequent one to 1.07 times when the bit-size of modules is 120-128 bits.

In cryptography, one of the methods used to estimate cipher strength is the computational complexity of sorting through all candidates, and this method is called a brute -force attack. Currently, one of the most common data encryption algorithms is AES – a symmetric encryption algorithm with a key length of 128 bits [18].

Based on the conducted research (formula 8), let us select the modules whose bit-size is similar to the use of the AES 128 algorithm (Table 2).

Table 1
Dependence of the total complexity of the cryptanalysis algorithm on the bit-size of modules, when n=3

№	2s	O(n, s)	Q(n, s)	Efficiency
	28	5,67E+07	225	---
	216	4,75E+14	248	1,92
	224	5,32E+21	272	1,50
	232	6,69E+28	295	1,32
	240	8,98E+35	2119	1,25
	248	1,26E+43	2143	1,20
	256	1,81E+50	2166	1,16
	264	2,65E+57	2190	1,14
	272	3,95E+64	2214	1,13
	280	5,97E+71	2238	1,11
	288	9,10E+78	2262	1,10
	296	1,40E+86	2286	1,09
	2104	2,17E+93	2310	1,08
	2112	3,38E+100	2333	1,07
	2120	5,29E+107	2357	1,07
	2128	8,32E+114	2381	1,07

where 2s – is the bit-size of modules that can be used; Q(n, s) – full digits, which are provided by O(n, s).

Table 2
Bit-size of modules with a key length of more than 128 bits

n	3	4	5	6	7	8
2s	248	240	232	224	224	224
Q(n, s)	2143	2155	2152	2134	2154	2174

In order to estimate the cryptographic strength of the ciphertext transmitted to the remote storage device, it is proposed to take into account not only the overall complexity of the cryptanalysis algorithm based on the RRNS, but also the size of the files of residues, since when intercepting a message, an intruder does not know the bit-size of the selected modules. Therefore, to decipher the message, it is necessary to sort through all the prime numbers that can be used as modules during encryption.

According to the asymptotic distribution of prime numbers (formula 4), the amount of prime numbers in the file of residues is determined by the following formula:

$$S(f) = f \cdot \sum_{i=3}^{f} \frac{1}{i} \tag{9}$$

where f is the bit-size of the files with residues.

Since the development of a system with a redundancy of more than 100% is not advisable, therefore the sum of the sizes of all files with residues should not exceed twice the size of the initial file, and the size of the fragment itself is calculated according the following formula:

$$f = \frac{2 \cdot F}{n},$$

where n is a number of modules; F is the size of the initial file.

Based on formula 9, we determine the dependence of cryptographic strength on the size of the files with residues (Table 3).

Table 3
Dependence of cryptographic strength on the size of the files with residues

f	1 byte	1 Kb	1 MB	10 MB	50 MB
Sf	9	6,63E +04	1,26E +08	1,45E +08	8,91E +10
$S = \lfloor 2^{i} \rfloor$	23	216	226	227	236

Accordingly, both the overall complexity of the cryptanalysis algorithm (formula 8) and the dependence of cryptographic strength on the size of the files with residues are taken into account to determine the values of the modules of the intercepted message (Table 2). At the same time, the complexity of message decryption is calculated according to formula:

$$O\left(\left(\frac{2^{s}}{s\ln 2}\right)^{n} \cdot n^{2} \cdot 2^{2 \cdot s}\right) \cdot \left(f \cdot \sum_{i=3}^{f} \frac{1}{i}\right) = O\left(\sum_{i=3}^{f} \frac{1}{i} \cdot \left(\frac{2^{s}}{s\ln 2}\right)^{n} \cdot f \cdot n^{2} \cdot 2^{2 \cdot s}\right) \qquad (10)$$

Based on the calculated computational complexity of the cryptanalysis algorithm (formula 8) and the cryptographic strength (Table 3), the required bit-size of the modules to ensure the complexity of the message decryption is determined (Table 4).

Table 4
The complexity of message decryption with different bit-sizes of modules

n	3		4		5		6		7		8	
2s	240	240	232	232	224	224	224	224	224	224	216	216
Minf	1 Kb	1 MB	1 Kb	1 MB	1 Kb	1 MB	1 Kb	1 MB	1 Kb	1 MB	1 Kb	1 MB
Q(n, f, s)	2135	2145	2140	2150	2129	2139	2150	2160	2170	2180	2130	2140

According to Table 4, it can be concluded that consideration of the minimum length of the files of residues of 1 KB when determining the complexity of the message decryption allows reducing the bit-size of modules by 8 bits while ensuring the required level of protection.

4. Data encryption method based on RRNS and pseudorandom sequences

4.1. Scheme of the data encryption method in RRNS

In order to increase the level of the encryption algorithm cryptographic strength based on the RRNS, this paper proposes an improvement of the encryption method by changing the position of the residues x_i using pseudorandom sequences PRS (S.key).

As a PRS, we select an M-sequence that provides a high bit generation rate and the required key length [19, 20].

Figure 1 shows an encryption scheme based on the RRNS, where p_i is a set of modules, Cipher is an encryption block, and Decipher is a decryption block.

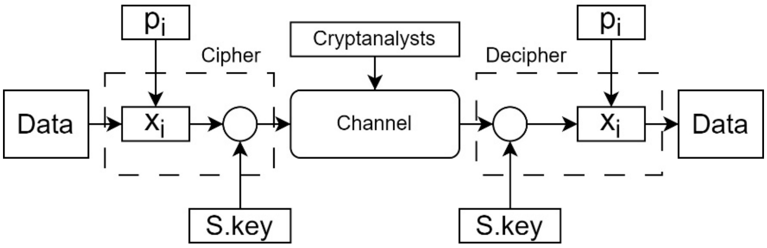

Figure 1: Scheme of the proposed encryption method based on RRNS.

It is proposed to write the calculated residue to the file of residues without change when the signal is '0'. In the case when the value of the sequence is equal to '1', then the residues are recorded into the following file of residues, and the last residue is recorded into the first file (Fig. 2).

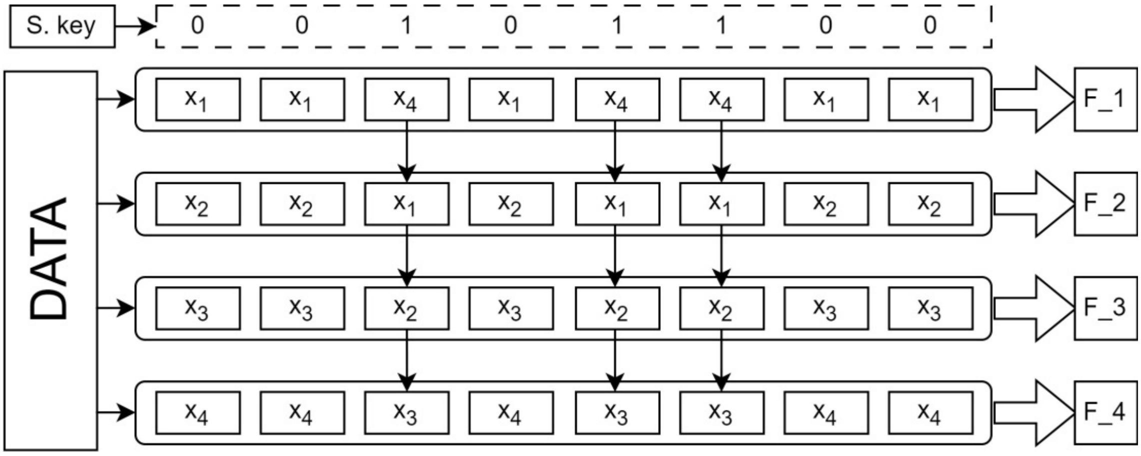

Figure 2: Data conversion using the shuffling method.

The residues are shifted according to the value of the PRS, which is notated as '0' and '1'.

During decryption, the inverse sequence of operations is performed, i.e., first, the contents of the files of residues (F_1 - F_4) are written into the corresponding residue arrays to which the shift key operation is subsequently applied.

Considering that the RRNS is a non-positional number system, the shift of the residues by one position makes it difficult to restore data. If the set of modules used to develop the system is unknown, an attacker will not be able to obtain the source file even after gaining access to all residue files.

4.2. Cryptoresistance of the data encryption method in RRNS

To recover the correct sequence of residues, an attacker needs to sort through each of the possible residues in all the residue files. It is possible to obtain the correct residue file if the sequence for all files is correctly selected. The number of residues depends on the number of modules bits, and the complexity of sorting possible candidates taking into account the shift, is determined by the following formula:

$$O(n, f, s, i) = \sum_{i=1}^{\frac{s}{8}} n!^{\frac{f}{8 \cdot i}} . \qquad (11)$$

As it has been proposed earlier to use a minimum size of the residue file of at least 1 KB, then by substituting the corresponding values into (11) and taking into account the data in Table 3, the complexity of sorting residues in the residue file can be determined (Table 5).

Table 5
The complexity of sorting residues by an attacker with unknown modules

n	3	4	5	6	7	8
2^{s}	2^{40}	2^{32}	2^{24}	2^{24}	2^{24}	2^{16}
Q(n, f, s, i)	2^{582}	2^{1064}	2^{1638}	2^{2106}	2^{2808}	2^{3010}

As it can be seen in Table 5, the complexity of sorting residues Q(n, f, s, i) taking into account the shift, is quite high, and its consideration makes it possible to increase the cryptographic strength of the developed system and reduce the bit-size of the modules.

Thus, taking into account the complexity of finding residues (formula 6), the cryptographic strength of the system (Table 2), the complexity of sorting the residues (Table 5) and messages whose residues will be at least 1 KB, it is possible to determine the complexity of breaking into the system when using a different number of modules with a bit-size of less than 16 bits (Table 6).

Table 6
Cryptographic strength of the system with up to 16-bit modules

n	3	4	5	6	7	8
Q(n, f, s, i)	2^{229}	2^{371}	2^{533}	2^{711}	2^{904}	2^{1109}

The results of the dependence of the system cryptographic strength on the bit-size of the residues with two information and one check modules are shown in Table 7.

As it can be seen from the above calculations (Table 7), when the bit-size of the modulus increases by 8, the bit-size of the computational complexity of the cryptanalysis algorithm Q(n, f, s, i) does not increase linearly, but it significantly decreases when the bit-size of modules increases up to 24 bits and higher.

4.3. The complexity of the cryptographic strength of the proposed method

A comparison of the dependence of the calculation complexity of the decryption algorithm O(n, k) (Table 1) and the developed data encryption method Q(n, f, s, i) (Table 6) on the bit-size of modules when n=3 is shown in Figure 3.

The proposed method provides 11-15 times higher cryptographic strength with 8-16 bit modules than the classical method, and on average 3 times higher with 96-128 bit modules.

It should be noted that this complexity of breaking into the system will arise only if the attacker does not know the bit-size and the modules themselves.

Table 7
Dependence of the system cryptographic strength on the bit-size of modules, when n=3

№	2^s	Q(n, f, s, i)	Efficiency
	2^8	2^{394}	---
	2^{16}	2^{559}	1,42
	2^{24}	2^{667}	1,19
	2^{32}	2^{749}	1,12
	2^{40}	2^{813}	1,09
	2^{48}	2^{867}	1,07
	2^{56}	2^{914}	1,05
	2^{64}	2^{954}	1,04
	2^{72}	2^{990}	1,04
	2^{80}	2^{1021}	1,03
	2^{88}	2^{1049}	1,03
	2^{96}	2^{1075}	1,02
	2^{104}	2^{1098}	1,02
	2^{112}	2^{1121}	1,02
	2^{120}	2^{1141}	1,02
	2^{128}	2^{1161}	1,02

However, when the system works for a long time, all official information usually becomes public knowledge.

In this case, an attacker can examine each digit position separately, checking whether the condition 0≤X≤M is fulfilled, and formula 11 is as follows:

$$O(n,f,s,i)=\sum_{i=1}^{\frac{s}{8}} n!\cdot\frac{f}{8\cdot i} \tag{12}$$

Taking into account the previously mentioned conditions and formula 12, the complexity of sorting through the fragments in the residue file with the modules known to an attacker is shown in Table 8.

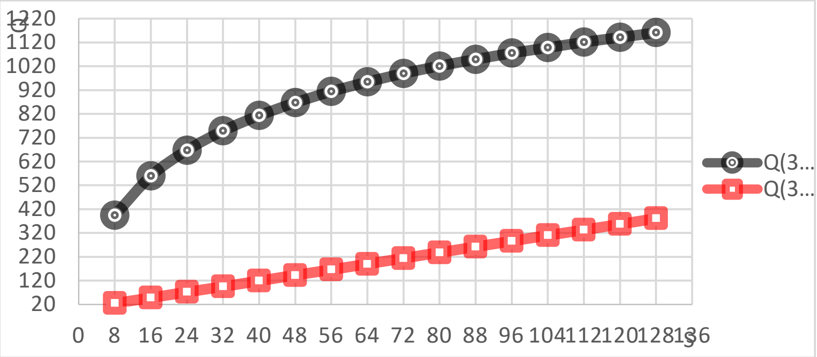

Figure 3: Comparison of the cryptographic strength of the RRNS O(n, k) and the developed method Q (n, f, s, i) due to the bit-size of k when n=3.

Table 8
Residue sorting complexity with known modules

n	3	4	5	6	7	8
2^s	2^{40}	2^{32}	2^{24}	2^{24}	2^{24}	2^{16}
Q (n, f, s, i)	2^7	2^9	2^{12}	2^{14}	2^{17}	2^{21}

As it is shown in Table 8, in the case when an attacker knows the set of modules p_i, which has been used to split the input data array into arrays of residues, the sorting complexity significantly reduces and largely depends on the length of the file and the key length.

Since it has been assumed that for calculations the residue files should be at least 1 KB, and the use of 1-byte modules is not advisable due to the increase in the time of the read/write operations, a 512-bit key is sufficient for data encryption.

5. Discussions

Among the results of this study, the following issues need to be discussed.

Firstly, when developing secure RRNS on the basis of redundant RRNS, the problem of justifying the number of check modules arises. According to the 2022 Backblaze Storage Cloud report [21], the percentage of failure of data storage devices (DSD) depends to a small extent on the size of the device itself and to a large extent it depends on the time of operation, but even for the devices that have been working continuously for more than 8 years, the percentage does not exceed 3.73% for small DSD (up to 10 TB), and for disks with a size of 12-16 TB, on average it is 1.07%. This proves the high reliability of modern DSD and allows the detection and correction of only a single error when calculating the number of RRNS check modules.

Secondly, we propose to use the bit-size of the residues not less than 1 KB. Table 3 shows the dependence of additional cryptographic strength on the size of the residue file.

Finally, a PRS of maximum length (M-sequence) is used as an additional key, because it provides a high bit generation rate and the required key length. However, other types of PRS can also be implemented.

6. Conclusion

The method for data encryption based on redundant RRNS and the residue shift by the M-sequence key is proposed and investigated in this paper.

The formula for estimating the cryptographic strength of data encryption based on the RRNS is created, which makes it possible to determine the number and bit-size of modules that provide a level of protection similar to the AES 128 algorithm in brute-force attacks.

It is proposed to reduce the minimum length of the message, which makes it possible to reduce the bit-size of modules necessary to achieve the specified level of protection.

A comparison of the cryptographic strength of the classical method of encryption in the RRNS and the proposed method based on the redundant RRNS and the residue shift by key is carried out.

The proposed method provides 11-15 times higher cryptographic strength using 8-16 bit modules than the classical method, and 3 times higher on average when using 96-128 bit modules.

In our opinion, the most promising areas of future research are the following ones:

- Studying the impact of other types of attacks on the proposed encryption method.
- Investigating the cryptographic strength of the algorithm when increasing the number of the RRNS modules.
- Studying the prospects of using the proposed approach to encryption of medical images.

References

[1] C. –H. Chang, A. S. Molahosseini, A. E. Zarandi and T. F. Tay, Residue Number Systems: A New Paradigm to Datapath Optimization for Low–Power and High–Performance Digital Signal Processing Applications. IEEE Circuits and Systems Magazine, Fourthquarter (2015) 26-44. doi: 10.1109/MCAS.2015.2484118.

[2] S. Lysenko, D. Sokalskyi & I. Mykhasko. Methods for cyberattacks detection in the computer networks as a mean of resilient it-infrastructure construction: state-of-art. Computer systems and information technologies, (2021) 31-35. doi: https://doi.org/10.31891/CSIT-2021-5-4

[3] H. R. Pawar, D. G. Harkut. Classical and Quantum Cryptography for Image Encryption & Decryption. 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), San Salvador, El Salvador, (2018) 1-4. doi: 10.1109/RICE.2018.8509035.

[4] A. Rao, D. Suma, Novel Image Encryption Algorithm with Image Integrity Check. 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, (2018) 98-104. doi: 10.1109/CSITSS.2018.8768797.

[5] J. D. Gaur, A. Kumar Singh, N. P. Singh and G. Rajan. Comparative Study on Different Encryption and Decryption Algorithm. 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, (2021) 903-908. doi: 10.1109/ICACITE51222.2021.9404734.

[6] B. Umapathy & G. Kalpana. A Key Generation Algorithm for Cryptographic Algorithms to Improve Key Complexity and Efficiency. In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, (2023) 647-652. doi: 10.1109/ICSSIT55814.2023.10060906.

[7] A. Sen, A. Ghosh and A. Nath. Bit level symmetric key cryptography using genetic algorithm. 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), Nagpur, India, (2017) 193-199. doi: 10.1109/CSNT.2017.8418536.

[8] T. Tabassum and M. A. Mahmood. A Multi-Layer Data Encryption and Decryption Mechanism Employing Cryptography and Steganography. 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), Bangladesh, (2020) 1-6. doi: 10.1109/ETCCE51779.2020.9350908.

[9] E. Ochoa-Jiménez, L. Rivera-Zamarripa, N. Cruz-Cortés and F. Rodríguez-Henríquez. Implementation of RSA Signatures on GPU and CPU Architectures. in IEEE Access, vol. 8, (2020) 9928-9941. doi: 10.1109/ACCESS.2019.2963826.

[10] Ananda Mohan, P.V. Residue Number Systems: Theory and Applications. Birkhäuser, Cham, Switzerland, 2016.

[11] E. Y. Baagyere, P. A. -N. Agbedemnab, Z. Qin, M. I. Daabo and Z. Qin. A Multi-Layered Data Encryption and Decryption Scheme Based on Genetic Algorithm and Residual Numbers. In IEEE Access, vol. 8, (2020) 100438-100447. doi: 10.1109/ACCESS.2020.2997838.

[12] P. A. -N. Agbedemnab, E. Y. Baagyere and M. I. Daabo. A New Image Encryption and Decryption Technique using Genetic Algorithm and Residual Numbers. 2019 IEEE AFRICON, Accra, Ghana, (2019) 1-9. doi: 10.1109/AFRICON46755.2019.9133919.

[13] V. Yatskiv, A. Sachenko, N. Yatskiv, P. Bykovyy and A. Segin. Compression and Transfer of Images in Wireless Sensor Networks Using the Transformation of Residue Number System. 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, (2019) 1111-1114. doi: 10.1109/IDAACS.2019.8924372..

[14] V. Yatskiv, S. Kulyna, P. Bykovyy, T. Maksymyuk and A. Sachenko. Method of Reliable Data Storage Based on Redundant Residue Number System. 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Dortmund, Germany, (2020) 1-4. doi: 10.1109/IDAACS-SWS50031.2020.9297052.

[15] A. Kar et al. Secuirity in cloud storage: An enhanced technique of data storage in cloud using RNS. 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, (2016) 1-4. doi: 10.1109/UEMCON.2016.7777905.

[16] E. Vassalos, & D. Bakalis. Residue-to-Binary Converter for the New RNS Moduli {22n−2, 2n−1, 2n+1}. 2019 Panhellenic Conference on Electronics & Telecommunications (PACET), Volos, Greece, (2019) 1-4. doi: 10.1109/PACET48583.2019.8956249.

[17] S. Kulyna. Evaluation of the reverse transformation methods complexity of the residual number system for secure data storage. Scientific Journal of TNTU, Tern.: TNTU, vol. 107, iss 3, (2022) 21-28. doi: 10.33108/visnyk_tntu2022.03.021.

[18] O. G. Abood, S. K. Guirguis. A survey on cryptography algorithms. International Journal of Scientific and Research Publications, vol. 8, iss 7, (2018) 410-415. doi: 10.29322/IJSRP.8.7.2018.p7978.

[19] M. Shirvanimoghaddam. On the Hamming Weight Distribution of Subsequences of Pseudorandom Sequences. 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, (2021) 1671-1675. doi: 10.1109/ISIT45174.2021.9517946.

[20] García, J. Espinosa, G. Cotrina, A. Peinado, A. Ortiz. Security and efficiency of linear feedback shift registers in GF (2n) using n−bit grouped operations. Mathematics, vol. 10, iss 6, (2022) 996. doi: 10.3390/math10060996.

[21] Backblaze Drive Stats for 2022, 2022. URL: https://www.backblaze.com/company/about.html.