Date of publication MAR-31, 2025, date of current version MAR-16, 2025.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.1.3880

A Symmetric Cryptoalgorithm Based on a Hierarchical Residue Number System

IHOR YAKYMENKO[1], MYKHAILO KASIANCHUK[1], OLESIA MARTYNIUK[2], SERHII MARTYNIUK[3]

[1]Department of Cyber Security West Ukrainian National University, 46003, Ternopil, Ukraine
[2]Department of Applied Mathematies West Ukrainian National University, 46003, Ternopil, Ukraine
[3]Department of Computer Science and Teaching Methods Ternopil Volodymyr Hnatiuk National Pedagogical University, 46015, Ternopil, Ukraine
(e-mail: iyakymenko@ukr.net, kasyanchuk@ukr.net, o.martyniuk@wunu.edu.ua, sergmart65@tnpu.edu.ua)

Corresponding author: Ihor Yakymenko (e-mail: iyakymenko@ukr.net).

ABSTRACT The paper develops a symmetric cryptoalgorithm based on a hierarchical system of remainder classes that allows to efficiently encrypt text messages using the remainders from dividing the numerical form of the plaintext into the corresponding modules. The peculiarity of this algorithm is its stepwise structure, which allows to gradually reduce the bit depth of modules and operands at each level. The software implementation and relevant experimental studies have shown that the abovementioned algorithm is highly resistant to cryptanalytic attacks due to the multi-level encryption structure and the use of large primes as modules at the first levels. It is established that the cryptographic strength increases with the number of modules, their bit depth, and hierarchical levels. A comparative analysis of the stability of the proposed algorithm and the AES-256 algorithm is carried out. It is determined at which values of the input parameters (bit depth of the modules, number of modules and hierarchy levels) the proposed algorithm demonstrates stability comparable to AES-256, while providing greater flexibility of settings and computational efficiency. The proposed methodology allows changing the number and bit depth of modules, the number of hierarchy levels, and other parameters to achieve the required degree of protection, making the algorithm versatile for different attacks and computing resources. This allows to adaptively adjust the system parameters to achieve the optimal ratio between the level of cryptographic strength and the speed of computation.

KEYWORDS symmetric cryptoalgorithm, hierarchical residue number system, module, remainder, Chinese remainder theorem, coding rules, cryptographic strength, hierarchical levels, bit depth of modules.

I. INTRODUCTION

Modern development of information technologies and globalization of the information space create new challenges for data protection [1, 2]. The number of cyberattacks is increasing every year [3], aimed at stealing confidential information, financial resources and personal data [4]. In this context, cryptographic methods remain the main tool for ensuring information security [5-7]. Particular attention is drawn to symmetric encryption methods [8] that provide a balance between the speed of data processing and the level of data protection [9, 10].

Existing modern algorithms, such as AES, have demonstrated high efficiency in countering cryptanalytic attacks [11-13]. However, the development of quantum computing [14] and the increase in the computing power of modern systems require the study of new approaches to creating cryptographically secure systems. In this context, the use of the residue number system (RNS) [15, 16] and its forms [17] is a promising direction, as it allows developing algorithms [18, 19] that have high speed and efficiency while maintaining the required level of resistance to attacks [20].

The relevance of the study is determined by the need to strengthen information security in the face of increasing complexity and scale of cyberattacks. The proposed approach is based on the use of the RNS mathematical apparatus, which combines high encryption efficiency with adaptability to different levels of threats [21-23]. In addition, it is important to find alternatives to traditional cryptographic methods to create new standards for information security in a rapidly changing digital environment.

The purpose of this article is to develop a symmetric cryptoalgorithm based on the use of hierarchical residue number system (HRNS) and to evaluate its stability. The proposed algorithm allows to adaptively adjust the system parameters to achieve an optimal ratio between the level of cryptographic strength and the speed of computation.

II. OVERVIEW OF EXISTING SOLUTIONS

In the field of cryptography, there is a constant need to improve algorithms and methods to ensure a high level of

security and increase the speed of processing confidential information. One way to enhance the efficiency of cryptographic operations is by using HRNS. Its foundations were laid in [24]. The HRNS proposed by the authors is based on representing large modules as subsystems of residues with smaller modules. This process can be repeated until the modules reach an acceptable length. Calculations can be performed within large dynamic ranges with several small modules, simplifying the transformation of numbers from residue format to positional representation.

In [25], a new method for performing base extensions using a hierarchical approach in RNS is proposed. For certain parameters, this significantly reduces calculations.

Several studies presented in the article [26] examine the development methods of HRNS and their application in asymmetric cryptography. The authors demonstrate that HRNS can effectively be used for data encryption and signing, providing a large dynamic range and high security level.

Additionally, work [27] shows that using specially selected modules of a two-level RNS compared to a conventional one with the same number of modules and the same dynamic range results in hardware savings of 25 to 40% and a complexity reduction of up to 80% in accordance with the RNS.

The work [28] indicates the possibility of expanding the dynamic range of the RNS by adding virtual layers, which contributes to the efficient implementation of modular operations for large numbers. This approach can find its application in large cryptographic systems, such as modern RSA encryption algorithms, and underscores the importance of applying a hierarchical approach in RNS for cryptographic tasks. Through it, computational costs can be significantly reduced, and a high level of parallelism can be achieved, which is crucial for the efficiency of cryptographic operations.

The paper [29] discusses new symmetric crypto algorithms based on ordinary integer RNS and its modified perfect form. The authors propose two methods. The first involves encrypting by transforming plaintext into a set of residues using corresponding modules (keys) and its subsequent restoration in decimal form using the Chinese remainder theorem (CRT). In the second method, the plaintext is divided into corresponding modules into smaller sub-blocks, which act as ciphertext. Decryption is performed by searching for residues of the ciphertext using corresponding modules. The authors also conduct research on the cryptographic strength of the proposed crypto algorithms based on the asymptotic distribution law of prime numbers and the use of the Euler function.

A logical continuation of this work is the article [30], which develops symmetric encryption algorithms based on RNS. The essence of encryption is that when restoring the decimal representation of a number from its residues using CRT, multiplication is performed not on found base numbers but on arbitrarily selected, mutually prime with modules, coefficients (keys) of symmetric encryption in RNS, allowing to enhance the cryptographic strength of algorithms.

The application of HRNS in cryptography and their research opens new perspectives for creating more efficient and secure cryptographic systems, which can find their implementation in a wide range of information technologies, including confidential information protection, and ensuring information and cyber security.

III. A HIERARCHICAL SYMMETRIC CRYPTOALGORITHM BASED ON A RESIDUE NUMBER SYSTEM

Any decimal number N in the RNS can be given by the set of residues b_i from dividing this number by selected modules p_i, which must be pairwise prime [31]:

$$b_i = N \bmod p_i. \tag{1}$$

The CRT allows to establish a mutually unambiguous correspondence between an integer N in the decimal system from the range $[0, P)$, where $P = \prod_{i=1}^{l} p_i$, $N < P$, and its residues:

$$N = \left(\sum_{i=1}^{l} b_i M_i m_i \right) \bmod P, \tag{2}$$

where $M_i = \dfrac{P}{p_i}$, m_i is determined by the expression $m_i = M_i^{-1} \bmod p_i$, l — number of modules.

Another way is to add the product of the previously considered modules sequentially. For example, the module p_1 is added to the residue b_1 until the congruence $N_1 \bmod p_2 = b_2$ is satisfied ($N_1 = b_1 + \gamma_1 p_1$, γ_1 is the number of additions of the module p_1). Next, the product of the modules $p_1 p_2$ is added to N_1 until the equality $N_2 \bmod p_3 = b_3$ ($N_2 = N_1 + \gamma_2 p_1 p_2$, γ_2 is the number of additions of $p_1 p_2$) is satisfied. At the i-th step ($i = 1, \ldots, l-1$), the congruence $N_i \bmod p_{i+1} = b_{i+1}$ ($N_i = N_{i-1} + \gamma_i p_1 p_2 p_3 \ldots p_i$) must be fulfilled. Figure 1 shows the scheme of restoring the decimal representation of a number by its residues based on the addition of the product of modules.

It is important to note that when using this approach, the results of intermediate calculations will not go beyond the set range P, which makes it impossible to overflow the bit grid and eliminates the need to perform the operation of finding the remainder modulo P. This method is similar to the Garner algorithm [32], according to which to calculate the coefficients γ_i, it is needed to use methods of finding the inverse element by the corresponding module:

$$\gamma_i = \left(b_{i+1} - N_i\right) \cdot \left(\left(p_1 p_2 \ldots p_i\right) \bmod p_{i+1} \right)^{-1} \bmod p_{i+1}.$$

$$N = \underbrace{b_1 + \gamma_1 p_1}_{\substack{\mid \bmod p_2 \\ \mid\mid\mid \\ \mid b_2}} + \gamma_2 p_1 p_2 + \gamma_3 p_1 p_2 p_3 + \ldots + \gamma_{l-1} p_1 p_2 p_3 \ldots p_{l-1},$$

$$\underbrace{}_{\substack{\mid \bmod p_3 \\ \mid\mid\mid \\ \mid b_3}}$$

$$\underbrace{}_{\substack{\mid \bmod p_4 \\ \mid\mid\mid \\ \mid b_4}}$$

$$\underbrace{}_{\substack{\mid \bmod p_l \\ \mid\mid\mid \\ \mid b_l}}$$

Figure 1. Scheme for restoring the decimal representation of a number by its remainders based on adding the product of modules.

IV. RULES FOR ENCODING TEXT

To encrypt textual information, it must first be converted into a numerical form. This is usually done using well-known tables of correspondences between letters and numbers (for example, ASCII codes, letter numbers in the corresponding alphabet). However, when using the decimal system, it is inconvenient when the letter number starts from zero (for example, $a = 00$, $b = 01$, etc.). Therefore, for transcoding letters of the Ukrainian and English alphabets, numbers, and some special characters, the principle is used when the character numbering is represented by a three-digit number starting from 100. This allows to effectively apply the proposed symmetric encryption/decryption method in a stepped RNS. Table 1 shows the character-number correspondence for encoding 167 characters. Obviously, Table 1 can be extended to 900 characters.

Table 1. Encoding of letters of the Ukrainian and English alphabets, special characters and numbers

а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131
	\n	.	,	:	!	?	%	#	@	№	;	^	*	(	)
132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147
-	+	=	_	<	>	{	}	[	]	|	/	`	~	\	"
148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163
\\	$	&	a	b	c	d	e	f	g	h	i	j	k	l	m
164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
n	o	p	q	r	s	t	u	v	w	x	y	z	A	Б	B
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М	Н	О
196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	A	B	C
212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243
T	U	V	W	X	Y	Z	1	2	3	4	5	6	7	8	9
244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259
0	—	–	«	»	ґ	Я									
260	261	262	263	264	265	266									

V. THEORETICAL FOUNDATIONS OF SYMMETRIC ENCRYPTION IN HIERARCHICAL RESIDUE NUMBER SYSTEM

The increase in the volume of information processing, transmission and storage inevitably leads to an increase in the number of modules and their size. This, in turn, causes hardware to become more complex and operations to take longer. Reducing the size of modules and, accordingly, numerical operands allows the HRNS.

Let the system of first-level modules $p_1, p_2, ..., p_l$, arranged in ascending order, provide the possibility of plaintext blocks N in the range $[0, P)$, where $P = \prod_{i=1}^{l} p_i$. The residues b_i are calculated using formula (1). Next, for each module p_i of the first level, a new system of modules of the second level is selected: $q_{i1}, q_{i2}, ..., q_{il}$, assuming for simplicity that the number of modules in each system at any level is the same and equal to l. Accordingly, the residues $b_{ij} = b_i \bmod q_{ij}$ are calculated.

Then, in turn, the residues of the second level are written in the same way in the system of modules of the third level, subject to the relevant requirements. This procedure continues until the last, k-th level.

Thus, each residue of the r-th level corresponds to l systems with l^{r-1} residues. Thus, l^r residues ($l > 1$, $r = 1, 2,..., k–1$) are transmitted to the $r + 1$ level and the ciphertext consists of l^k numbers, which are the residues of the last level of the HRNS. If the number of levels is 1, then the usual symmetric encryption in the RNS takes place. The module sets are known to the sender and the receiver.

As a rule, the number of levels is determined by the conditions of a particular task. Such a process of transition to smaller modules greatly simplifies the implementation of an elementary arithmetic device and reduces the time for performing arithmetic operations. Figure 2 shows a diagram of processing residues at the appropriate levels.

It should be noted that for cryptography tasks, it is advisable to choose different numbers of modules at different levels. This will significantly increase the complexity of cryptanalysis of such an encryption system, although it will complicate the hardware implementation.

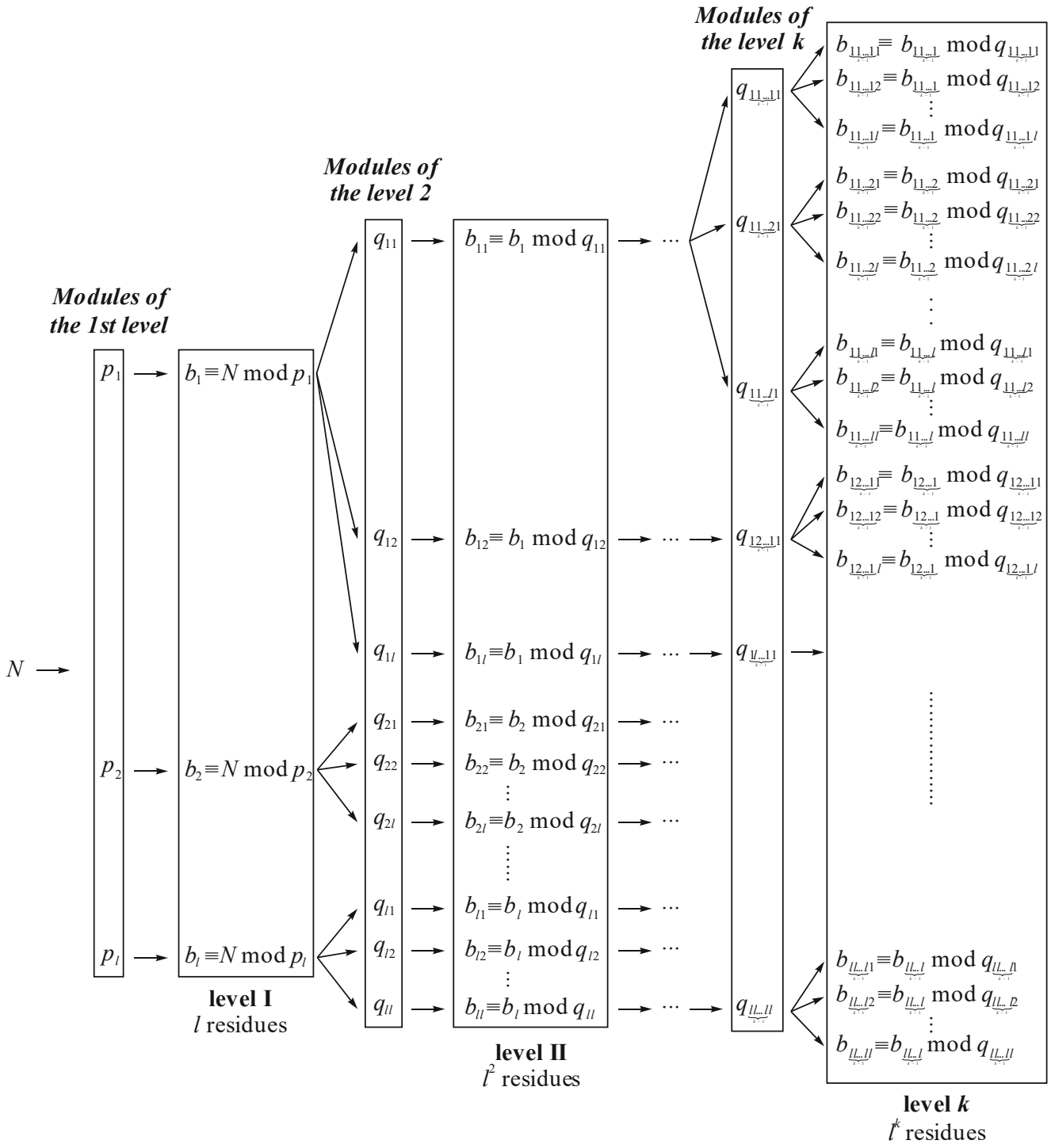


Figure 2. Scheme of transfer of residues by levels.

If the plaintext message converted into numerical form does not satisfy the condition $N < P$, it must be divided into blocks that meet the specified inequality.

Decryption in an HRNS is performed in the reverse order. The entire ciphertext is divided into blocks, the number of which is l^k. Then, using one of the methods of recovering a number from its residues, for example, the CRT (2), the residues of a higher level are obtained. At each level of decryption, the number of residues is reduced by a factor of l.

At the last level of decryption, the numerical value of the input message is obtained. Figure 3 shows a general scheme of symmetric encryption based on the HRNS.

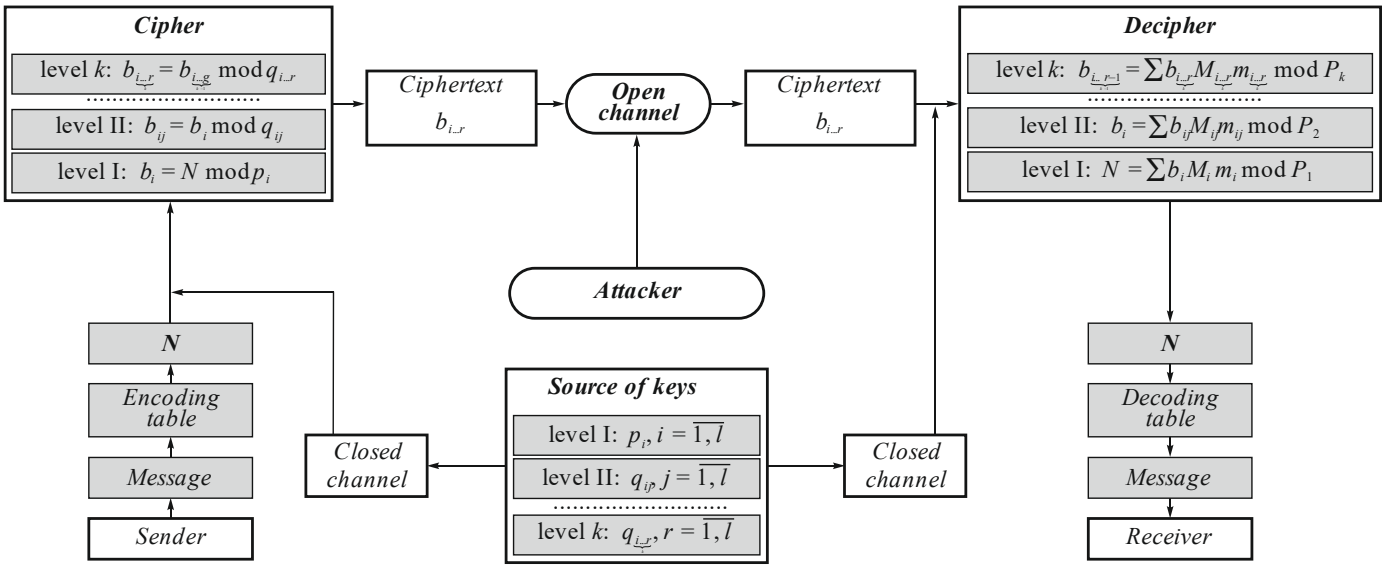


Figure 3. Scheme of symmetric encryption and decryption based on HRNS

VI. AN EXAMPLE OF A SYMMETRIC ENCRYPTION METHOD IN AN HIERARCHICAL RESIDUE NUMBER SYSTEM

For example, let's take a two-level HRNS with the following sets of modules of the 1st and 2nd levels, respectively: $p_1 = 164359$, $p_2 = 1287493$, $p_3 = 1702861$; $q_{11} = 167$, $q_{12} = 331$, $q_{13} = 599$, $q_{21} = 137$, $q_{22} = 283$, $q_{23} = 677$, $q_{31} = 157$, $q_{32} = 367$, $q_{33} = 709$. Suppose that it is needed to encrypt the plaintext “looks cool”, which is divided into two blocks of 5 characters each, including spaces: S_1 = “looks”, S_2 = “ cool”. According to Table 1, two 15-digit numeric blocks are obtained: $N_1 = 178181181177185$, $N_2 = 132169181181178$. The results of the hierarchical encryption of blocks N_1 and N_2 are shown in Tables 2-3.

Table 2. Encryption N_1

$N_1 = 178181181177185$				
$p_1 = 164359$	$b_1 = 10\,326$	$q_{11} = 167$	$b_{11} = 139$
		$q_{12} = 331$	$b_{12} = 65$
		$q_{13} = 599$	$b_{13} = 143$
$p_2 = 1287493$	$b_2 = 1109499$	$q_{21} = 137$	$b_{21} = 73$
		$q_{22} = 283$	$b_{22} = 139$
		$q_{23} = 677$	$b_{23} = 573$
$p_3 = 1702861$	$b_3 = 1145503$	$q_{31} = 157$	$b_{31} = 31$
		$q_{32} = 367$	$b_{32} = 96$
		$q_{33} = 709$	$b_{33} = 468$

Thus, the blocks S_1 = “looks” and S_2 = “ cool” correspond to the residue sets (139, 65, 143, 73, 139, 573, 31, 96, 468) and (63, 133, 264, 37, 251, 403, 37, 260, 656), respectively. The full ciphertext is obtained by concatenating both sets of residues: (139, 65, 143, 73, 139, 573, 31, 96, 468, 63, 133, 264, 37, 251, 403, 37, 260, 656).

Table 3. Encryption N_2

$N_2 = 132169181181178$				
$p_1 = 164359$	$b_1 = 136\,836$	$q_{11} = 167$	$b_{11} = 63$
		$q_{12} = 331$	$b_{12} = 133$
		$q_{13} = 599$	$b_{13} = 264$
$p_2 = 1287493$	$b_2 = 1\,074\,802$	$q_{21} = 137$	$b_{21} = 37$
		$q_{22} = 283$	$b_{22} = 251$
		$q_{23} = 677$	$b_{23} = 403$
$p_3 = 1702861$	$b_3 = 136\,784$	$q_{31} = 157$	$b_{31} = 37$
		$q_{32} = 367$	$b_{32} = 260$
		$q_{33} = 709$	$b_{33} = 656$

The decryption process is carried out in reverse order using the CRT, starting from the second level with 9 modules: $q_{11} = 167$, $q_{12} = 331$, $q_{13} = 599$, $q_{21} = 137$, $q_{22} = 283$, $q_{23} = 677$, $q_{31} = 157$, $q_{32} = 367$, $q_{33} = 709$. After recovering the residuals of the second level, the decryption is performed using three modules of the first level: $p_1 = 164359$, $p_2 = 1287493$, $p_3 = 1702861$.

Thus, the ciphertext is divided into two blocks of 9 residues each. The first of them (139, 65, 143, 73, 139, 573, 31, 96, 468) is divided into sub-blocks of three residues, to which the CRT (2) with second-level modules is applied. The resulting three numbers (remainders) are again subjected to the CRT with first-level modules. The procedure is similar with the second block of the ciphertext. The process and results of decrypting each block are presented in Tables 4 and 5, respectively.

Table 4. Decryption of the first block

		I			II			III		
II level	b_{ij}	$b_{11} = 139$	$b_{12} = 65$	$b_{13} = 143$	$b_{21} = 73$	$b_{22} = 139$	$b_{23} = 573$	$b_{31} = 31$	$b_{32} = 96$	$b_{33} = 468$
	q_{ij}	$q_{11} = 167$	$q_{12} = 331$	$q_{13} = 599$	$q_{21} = 137$	$q_{22} = 283$	$q_{23} = 677$	$q_{31} = 157$	$q_{32} = 367$	$q_{33} = 709$
	$M_{ij} = \frac{\prod_{j=1}^{l} q_{ij}}{q_{ij}}$	198 269	100 033	55 277	191 591	92 749	38 771	260 203	111 313	57 619
	$m_{ij} = M_{ij}^{-1} \bmod q_{ij}$	71	14	319	78	83	93	32	213	153
	$b_i = \left(\sum_{j=1}^{l} b_{ij} M_{ij} m_{ij}\right) \bmod \prod_{j=1}^{l} q_{ij}$	10326			1 109 499			1 145 503		
I level	p_i	$p_1 = 164359$			$p_2 = 1287493$			$p_3 = 1702861$		
	$M_i = \frac{\prod_{i=1}^{l} p_i}{p_i}$	2 192 421 617 473			279 880 531 099			211 611 061 987		
	$m_i = M_i^{-1} \bmod p_i$	59400			1 108 855			1 323 710		
	$N_1 = \left(\sum_{i=1}^{l} b_i M_i m_i\right) \bmod \prod_{i=1}^{l} p_i$	178 181 181 177 185								
	S_1	«looks»								

Table 5. Decryption of the second block

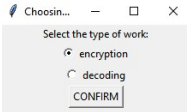
		I			II			III		
II level	b_{ij}	$b_{11} = 63$	$b_{12} = 133$	$b_{13} = 264$	$b_{21} = 37$	$b_{22} = 251$	$b_{23} = 403$	$b_{31} = 37$	$b_{32} = 260$	$b_{33} = 656$
	q_{ij}	$q_{11} = 167$	$q_{12} = 331$	$q_{13} = 599$	$q_{21} = 137$	$q_{22} = 283$	$q_{23} = 677$	$q_{31} = 157$	$q_{32} = 367$	$q_{33} = 709$
	$M_{ij} = \frac{\prod_{j=1}^{l} q_{ij}}{q_{ij}}$	198 269	100 033	55 277	191 591	92 749	38 771	260 203	111 313	57 619
	$m_{ij} = M_{ij}^{-1} \bmod q_{ij}$	71	14	319	78	83	93	32	213	153
	$b_i = \left(\sum_{j=1}^{l} b_{ij} M_{ij} m_{ij}\right) \bmod \prod_{j=1}^{l} q_{ij}$	136 836			1 074 802			136 784		
I level	p_i	$p_1 = 164359$			$p_2 = 1287493$			$p_3 = 1702861$		
	$M_i = \frac{\prod_{i=1}^{l} p_i}{p_i}$	2 192 421 617 473			279 880 531 099			211 611 061 987		
	$m_i = M_i^{-1} \bmod p_i$	59400			1 108 855			1 323 710		
	$N_2 = \left(\sum_{i=1}^{l} b_i M_i m_i\right) \bmod \prod_{i=1}^{l} p_i$	132 169 181 181 178								
	S_2	« cool»								

Thus, after combining the decrypted texts of both blocks, the incoming message “looks cool” is obtained.

VII. SOFTWARE IMPLEMENTATION OF THE METHOD

The implementation of the proposed encryption method is based on the Python software. The calculations in the study were performed using HP Pavilion Gaming Laptop 16-a0xxx/Intel(R) Core(TM) i7-10870H CPU @ 2.20 GHz, 2208 Mhz, 8 Core(s)/32 Gb/1 Tb. At the initial stage, the necessary libraries were exported, a table of encoding letters, numbers and symbols was created, functions for displaying text during encryption and decryption, as well as auxiliary functions for displaying windows on the screen, etc. A description of the main functions of the program implementation and the output of the result of execution on the screen for encryption and decryption is given in Tables 6 and 7, respectively.

Table 6. Description of the main functions of the encryption software implementation and the result of displaying on the screen

Function description	*The result of the screen display*
Create and display a job selection window	
Create and display a window for entering text and checking if all characters are in the alphabet	

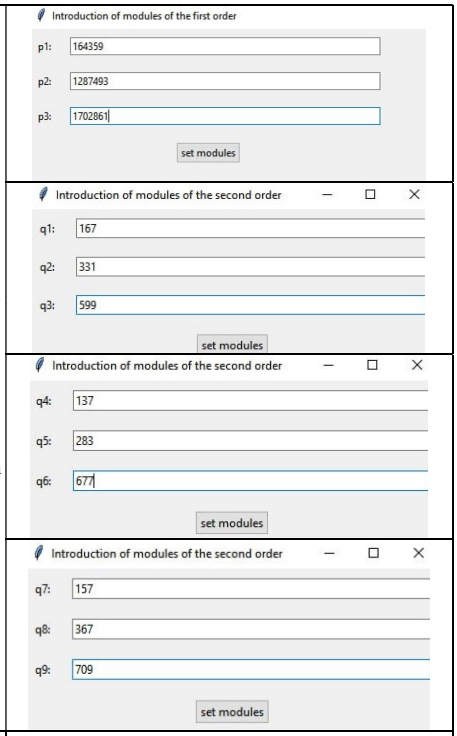
Creating and displaying a window for entering first- and second-order modules and checking the conditions for the entered data, namely, mutual primes, belonging to a certain range	
A timer for evaluating the program's performance, block encryption of the text, writing data to a file for transmission to the recipient, and additional text output to a window with the ability to copy	

Table 7. Description of the main functions of the decryption software implementation and the result of displaying on the screen

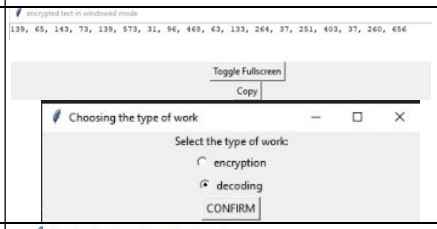
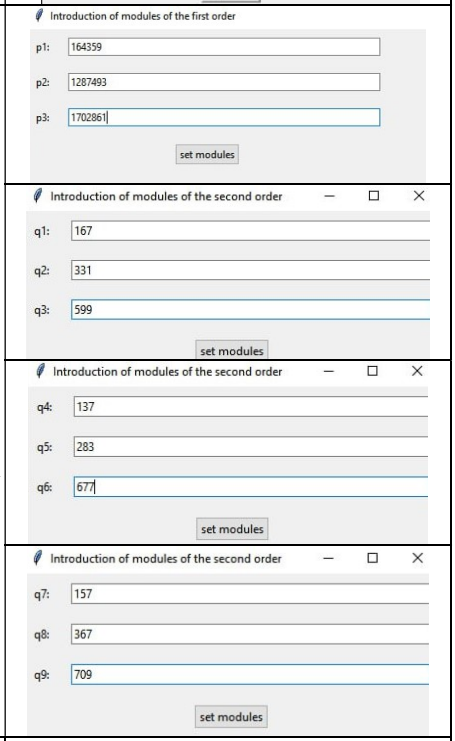
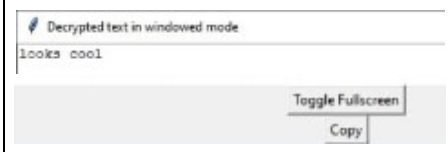
Code description	*The result of the screen display*
Create and display a window for setting the cryptotext and selecting the type of work	
Creating and displaying a window for entering modules of the first and second levels and checking the conditions for the entered data, namely: mutual simplicity, belonging to a certain range	
Reading data, block decryption, creating a file with decrypted text, displaying it on the screen and estimating the operating time of each level	

Table 8 shows the encryption and decryption time for messages of different lengths at each level.

Table 8. Encryption and decryption time for messages of different lengths at each level

Text size, bits	Encryption time,		Decryption time	
	I level	II level	I level	II level
8	7.799943E–06	4.300033E–06	2.659997E–05	1.419999E–05
16	1.279998E–05	7.799943E–06	5.199993E–05	2.020004E–05
32	1.630012E–05	1.369999E–05	4.129997E–05	2.959999E–05
64	2.010015E–05	1.389976E–05	5.489995E–05	4.520011E–05
128	2.790009E–05	2.320018E–05	1.246998E–04	9.370025E–05
256	5.609961E–05	6.760005E–05	6.430999E–04	2.128998E–04
512	1.146998E–04	9.070023E–05	5.221999E–04	4.703999E–04
1024	2.345002E–04	1.778999E–04	7.691999E–04	7.490998E–04
2048	4.413006E–04	3.627992E–04	2.385101E–03	2.479399E–03
4096	9.392999E–04	7.505005E–04	3.148402E–03	3.015401E–03

It is expected that as the message dimension increases from 8 bits to 4096 bits, the time of cryptographic transformations increases by about 100 times. With the same bit depths, the decryption time is longer than the encryption time. This is due to the use of computationally complex CRT operations during decryption. In addition, the time of cryptographic transformations at the second level, which uses smaller modules, is usually less than at the first level.

VIII. EVALUATION OF CRYPTOGRAPHIC STRENGTH OF A CRYPTOGRAPHIC ALGORITHM IN A HIERARCHICAL SYSTEM OF RESIDUAL CLASSES

As noted in [29], the complexity of cryptanalysis of a symmetric encryption method in an integer RNS, provided that the crypto-transformation modules are primes, is $O\left(\left(\frac{2^{n+1}}{n}\right)^{l} n^2\right)$, where n — bit depth of the modules. When studying the cryptographic resistance of the developed algorithm using the HRNS, it is necessary to take into account the number of levels k and the change in the bit depth of the modules at each level. Therefore, the total time complexity of cryptanalysis at all levels is calculated according to the following formula:

$$O(n,k,l)=\prod_k\left(\frac{2^{n_k+1}}{n_k}\right)^{l_k} {n_k}^2, \qquad (3)$$

where $n_k = n - k + 1$ — is the bit depth at the k-th level, $l_k = l^k$ is the number of modules at the k-th level.

Taking into account the values of the parameters n_k and l_k, formula (3) will take the following form:

$$O(n,k,l)=\prod_k\left(\frac{2^{n-k+1}}{n-k+1}\right)^{l^k}(n-k+1)^2. \qquad (4)$$

For example, the cryptographic strength of the proposed algorithm for a k-level HRNS with the number of modules on the first level $l = 3$ is described by the expression:

$$O(n,k)=\prod_k\left(\frac{2^{n-k+1}}{n-k+1}\right)^{3^k}(n-k+1)^2. \qquad (5)$$

Figure 4 shows a graph that displays the logarithmic dependence of the complexity of cryptanalysis on the number of bits n, the number of modules l, and the levels k. It is clearly observed that with the growth of these parameters, the complexity of cryptanalysis increases.

The results of experimental studies indicate that the stability of the proposed symmetric cryptosystem based on the use of a hierarchical system of residual classes increases with the number of levels and the bit depth of the input parameters (modules).

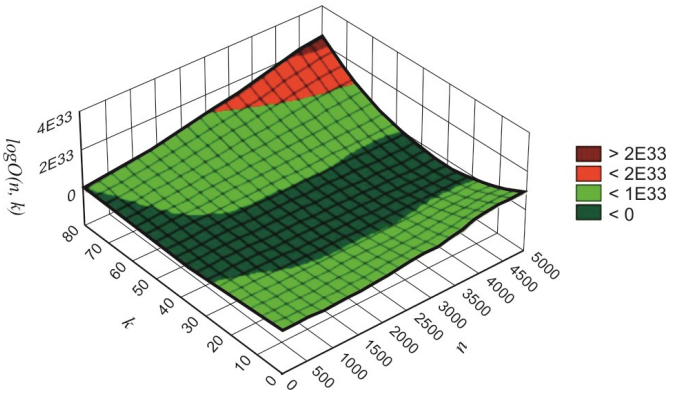


Figure 4. Dependence of the logarithm of cryptanalysis complexity on the number of bits n, the number of modules l and the levels k.

From the literature review, it is known [33, 34] that the cryptanalysis of the modern symmetric cipher AES-256 is 2^{255} bit operations. From the equality $\prod_k\left(\frac{2^{n-k+1}}{n-k+1}\right)^{l^k}(n-k+1)^2 = 2^{255}$ we can find the bits, the number of levels, and the number of RNS modules that provide the same robustness as the AES-256 algorithm.

After logarithmizing in base 2 and performing elementary arithmetic transformations, we can get

$$\sum_k\left(l^k(n-k+1)+(2-l^k)\log_2(n-k+1)\right)=255. \qquad (6)$$

Table 9 shows the values of cryptographic strength for different values of n, l and k.

Table 9. Cryptographic resistance of the developed algorithm in HRNS at different module bit depths, numbers of modules, and levels

$n = 8$					
k	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
1	16	21	26	31	36
2	38.38529	64.34852	98.69703	141.4308	192.5499
3	70.87552	161.7245	322.4294	573.4804	935.368
4	118.3685	383.2921	1012.66	2251.919	4410.793
5	186.3685	873.2921	3064.66	8505.919	19966.79
6	280.1008	1908.024	8863.823	30619.05	85989.95
$n = 12$					
k	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
1	24	32.41504	40.83007	49.24511	57.66015
2	61.08114	107.199	168.398	244.6782	336.0395
3	121.1496	294.1508	602.4385	1086.081	1785.147
4	220.7706	772.7267	2101.278	4736.218	9347.264
5	386.7706	1993.727	7227.278	20367.22	48233.26

Table 9 demonstrates that the proposed symmetric cryptographic algorithm based on the HRNS provides a level of cryptographic strength comparable to the AES-256 algorithm at certain values of the parameters n, l, k.

In particular, for 8-bit modules with $l = 2$, 6 levels are required to exceed the AES-256 strength. Increasing the number of modules leads to a decrease in encryption levels (if $l = 3$, then $k = 4$; if $l = 4$-6, then $k = 3$).

For 12-bit modules with $l = 2$, the strength of the proposed

cryptoalgorithm exceeds the strength of AES-256 at the fifth level. With an increase in the number of modules (l = 3-5), the condition of exceeding the AES-256 strength is fulfilled at the third level, and at l = 6 – at the second level.

Thus, varying the bit depth of the modules, their number, and hierarchy levels allows us to achieve the appropriate cryptographic strength of the proposed algorithm depending on the specific task.

IX. CONCLUSIONS

For the first time, a symmetric cryptoalgorithm based on the HRNS has been proposed that allows to efficiently convert text messages into numerical form and ensure their encryption using the remaining modules. A feature of the algorithm is its stepwise structure, which makes it possible to consistently reduce the bit depth of the modules at each level and ensure the preservation of the original information.

Experimental studies have demonstrated that the proposed algorithm is highly resistant to attacks due to the use of large prime numbers in the system of modules and a multi-level encryption structure. It is established that cryptographic resistance increases with the number of modules, their bit depth, and hierarchy levels.

A comparative analysis of the stability of the proposed algorithm and the AES-256 algorithm is carried out. It is indicated at what values of the input parameters (bit depth of the modules, number of modules, and hierarchy levels) the algorithm based on the RNS demonstrates stability comparable to AES-256, while providing greater flexibility of settings and computational efficiency.

The proposed methodology allows changing the bit depth of the modules, the number of hierarchy levels, and other parameters to achieve the required level of protection. This makes the algorithm universal for use in conditions of different levels of threats and computer system resources.

In general, the conducted studies confirm the effectiveness and prospects of using a symmetric cryptographic algorithm based on the HRNS to ensure the protection of information in the modern digital environment. The proposed approach opens up new opportunities for the development of cryptography that meet the challenges of the digital age.

REFERENCES

[1] D. Le and A. Zincir-Heywood, "Exploring anomalous behaviour detection and classification for insider threat identification," *Int. J. Netw. Manag.*, vol. 31, no. 2, p. e2109, 2020. https://doi.org/10.1002/nem.2109.

[2] S. Kostoudas, O. Markovskyi, N. Doukas, & N. Bardis, "Secure and encrypted communication system on mobile devices," *In 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, December 2022, pp. 1-6. https://doi.org/10.1109/DESSERT58054.2022.10018747.

[3] M. Dawood, S. Tu, C. Xiao, H. Alasmary, M. Waqas, and S. Rehman, "Cyberattacks and security of cloud computing: A complete guideline," *Symmetry*, vol. 15, no. 11, p. 1981, 2023. https://doi.org/10.3390/sym15111981.

[4] R. Leszczyna, "Review of cybersecurity assessment methods: Applicability perspective," *Comput. Secur.*, vol. 108, p. 102376, 2021. https://doi.org/10.1016/j.cose.2021.102376.

[5] N. Valarmathy and P. Vishnupriya, "Network security and cryptography techniques," *Netw. Commun. Eng. J.*, vol. 9, no. 9, pp. 229–231, 2017. [Online]. Available at: https://www.ciitresearch.org/dl/index.php/nce/article/view/NCE112017004.

[6] M. Marwaha, R. K. Bedi, A. Singh, and T. Singh, "Comparative analysis of cryptographic algorithms," *Int. J. Adv. Eng. Technol.*, vol. 9, pp. 16–18, 2013. [Online]. Available at: https://api.semanticscholar.org/CorpusID:14369407.

[7] A. G. Khan, S. Basharat, and M. U. Riaz, "Analysis of asymmetric cryptography in information security based on computational study to ensure confidentiality during information exchange," *Int. J. Sci. Eng. Res.*, vol. 9, no. 10, pp. 992–999, 2018. [Online]. Available at: https://doi.org/10.13140/RG.2.2.30495.61602.

[8] M. Kasianchuk, I. Yakymenko, and Y. Nykolaychuk, "Symmetric cryptoalgorithms in the residue number system," *Cybern. Syst. Anal.*, vol. 57, no. 2, pp. 329–336, 2021. https://doi.org/10.1007/s10559-021-00358-6.

[9] C. Ubochi, B. Olaniyi, K. Ukagwu, and S. Nnamchi, "A comparative analysis of symmetric cryptographic algorithm as a data security tool: A survey," *J. Sci. Technol. Res.*, vol. 5, no. 3, pp. 144–168, 2023.

[10] O. G. Abood and S. K. Guirguis, "A survey on cryptography algorithms," *Int. J. Sci. Res. Publ.*, vol. 8, no. 7, pp. 410–415, 2018. https://doi.org/10.29322/IJSRP.8.7.2018.p7978.

[11] M. N. A. Wahid, A. Ali, B. Esparham, and M. Marwan, "A comparison of cryptographic algorithms: DES, 3DES, AES, RSA and BlowFish for guessing attacks prevention," *J. Comp. Sci. Appl. Inform. Technol.*, vol. 3, no. 2, pp. 1–7, 2018. [Online]. Available at: https://api.semanticscholar.org/CorpusID:52035951.

[12] P. Princy, "A comparison of symmetric key algorithms DES, AES, BlowFish, RC4, RC6: A survey," *Int. J. Comput. Sci. Eng. Technol.*, vol. 6, no. 5, pp. 328–331, 2015. [Online]. Available at: https://api.semanticscholar.org/CorpusID:61177706.

[13] M. Mathur and A. Kesarwani, "Comparison between DES, 3DES, RC2, RC6, BlowFish and AES," *Proc. Nat. Conf. New Horiz. IT-NCNHIT*, vol. 3, pp. 143–148, 2013. [Online]. Available at: https://api.semanticscholar.org/CorpusID:18620923.

[14] M. Lakhan and A. Ospanova, "Quantum computer," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 11, no. 4, pp. 3372–3376, 2022. doi: 10.15680/IJIRSET.2022.1104029.

[15] P. Ananda Mohan, *Residue number systems: theory and applications*, Birkhäuser, 2016, p. 351. https://doi.org/10.1007/978-3-319-41385-3.

[16] A. Omondi and B. Premkumar, *Residue number systems: theory and implementation*, Imperial College Press, 2007, p. 296. https://doi.org/10.1142/9781860948671.

[17] M. Kasianchuk, Y. Nykolaychuk, and I. Yakymenko, "Theory and methods of constructing of modules system of the perfect modified form of the system of residual classes," *J. Autom. Inf. Sci.*, vol. 48, no. 8, pp. 56–63, 2016. doi: 10.1615/JAutomatInfScien.v48.i8.60.

[18] N. Vivek and K. Anusudha, "Design of RNS based addition subtraction and multiplication units," *Int. J. Eng. Trends Technol.*, vol. 10, no. 12, pp. 593-596, 2014. https://doi.org/10.1615/JAutomatInfScien.v48.i8.60.

[19] K. V. Lalitha and V. Sailaja, "High performance adder using residue number system," *Int. J. VLSI Embedded Syst.*, vol. 5, pp. 1323-1332, 2014. [Online]. Available at: https://api.semanticscholar.org/CorpusID:41207527.

[20] D. Schoinianakis, "Residue arithmetic systems in cryptography: A survey on modern security applications," *J. Cryptogr. Eng.*, vol. 10, no. 3, pp. 249–267, 2020. https://doi.org/10.1007/s13389-020-00231-w.

[21] I. R. Fadulilahi, E. K. Bankas, and J. B. A. K. Ansuura, "Efficient algorithm for RNS implementation of RSA," *Int. J. Comput. Appl.*, vol. 127, no. 5, pp. 14-19, 2015. https://doi.org/10.5120/ijca2015906381.

[22] M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, and K. Navi, "Efficient RNS implementation of elliptic curve point multiplication GF(p)," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 21, pp. 1545–1549, 2013. https://doi.org/10.1109/TVLSI.2012.2210916.

[23] A. Kar et al., "Security in cloud storage: An enhanced technique of data storage in cloud using RNS," in *Proc. IEEE 7th Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf. (UEMCON)*, New York, USA, 2016, pp. 1–4. https://doi.org/10.1109/UEMCON.2016.7777905.

[24] H. M. Yassine, "Hierarchical residue numbering system suitable for VLSI arithmetic architecture," *Circuits Syst. (ISCAS '92): Proc. IEEE Int. Symp.*, San Diego, CA, USA, 1992, pp. 811–814. https://doi.org/10.1109/ISCAS.1992.230098.

[25] L. Djath, L. Bigou, and A. Tisserand, "Hierarchical approach in RNS base extension for asymmetric cryptography," *ARITH: 2019 IEEE 26th Symp. Comput. Arithmetic*, Kyoto, Japan, Jun 2019. https://doi.org/10.1109/ARITH.2019.00016.

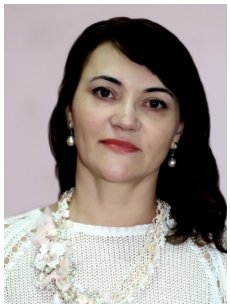
[26] T. Tomczak, "Hierarchical residue number systems with small moduli and simple converters," *Int. J. Appl. Math. Comput. Sci.*, vol. 21, no. 1, pp. 173-192, 2011. https://doi.org/10.2478/v10006-011-0013-2.

[27] A. Skavantzos and M. Abdallah, "Implementation issues of the two-level residue number system with pairs of conjugate moduli," *IEEE*

Trans. Signal Process., vol. 47, no. 3, pp. 826-838, 1999. https://doi.org/10.1109/78.747787.
[28] H. D. L. Hollmann, R. Rietman, S. de Hoogh, and L. Tolhuizen, "A Multi-layer recursive residue number system," *IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 1460–1464. https://doi.org/10.1109/ISIT.2018.8437612.
[29] S. Zawislak, M. Kasianchuk, I. Yakymenko, and D. Jancarczyk, "Methods of crypto-stable symmetric encryption in the residual number system," *Proc. 26th Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst. (KES 2022)*, *Procedia Comput. Sci.*, vol. 207, pp. 128–137, 2022. https://doi.org/10.1016/j.procs.2022.09.045.
[30] Y. Nykolaychuk, I. Yakymenko, N. Vozna, and M. Kasianchuk, "Residue number system asymmetric cryptoalgorithms," *Cybernetics Syst. Anal.*, vol. 58, no. 4, pp. 611–618, 2022. https://doi.org/10.1007/s10559-022-00494-7.
[31] N. Singh, "An overview of Residue Number System," in *Devices, Circuits & Commun.: Proc. Nat. Seminar*, 2008, pp. 132–135. [Online]. Available at: https://www.researchgate.net/publication/307174628.
[32] Y. Li, L. Xiao, A. Liang, Y. Zheng, and L. Ruan, "Fast Parallel Garner Algorithm for Chinese Remainder Theorem," in *9th Int. Conf. Netw. Parallel Comput. (NPC)*, Gwangju, South Korea, 2012, pp. 164–171. https://doi.org/10.1007/978-3-642-35606-3_19.
[33] G. V. Bard, *Algebraic Cryptanalysis*, Springer, Boston, MA, USA, 2009, p. 392. https://doi.org/10.1007/978-0-387-88757-9.
[34] H. Nover, "Algebraic Cryptanalysis of AES: An Overview," University of Wisconsin, Madison, WI, USA, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:11862091.

***IHOR YAKYMENKO** in 1998 graduated from the Taras Shevchenko National University of Kyiv on specialty Mathematics. In 2012 defended his dissertation for the Candidate degree (Ph.D.) of Technical Sciences. In 2018 received the title of Associate Professor of the Department of Computer Engineering. Currently, is the Dean of the Faculty of Computer Information Technologies of the West Ukrainian National University and works as an Associate Professor of the Department of Computer Engineering. Scientific interests: remainder class system, research of mathematical operations in cryptographic methods of information protection, asymmetric systems.*

***MYKHAILO KASIANCHUK** in 1994 graduated from the Ternopil State Pedagogical Institute on specialty Mathematics and Physics. In 2001 defended his dissertation for the Candidate degree (Ph.D.) of Physical and Mathematical Sciences. In 2005 received the diploma of Associate Professor on the Department of Information Technology Security. In 2020 defended his dissertation for the degree of Doctor of Technical Sciences, and in 2022 received the academic title of Professor. At the present time working as an professor of the Department of Cyber Security in West Ukrainian National University. Scientific interests: the system of residual classes, research of mathematical operations in cryptographic methods.*



***OLESIA MARTYNIUK** in 1989 graduated from the Ternopil State Pedagogical Institute on specialty Mathematics and Physics. In 1992 defended her dissertation for the Candidate degree (Ph.D.) of Physical and Mathematical Sciences. In 1997 received the diploma of Associate Professor on the Department of Mathematics. At the present time is working as an Associate Professor of the Applied Mathematies in West Ukrainian National University. Scientific interests: numerical computing methods, optimization methods, mathematical statistics, computer modeling.*

***SERHII MARTYNIUK** in 1989 graduated from the Ternopil State Pedagogical Institute on specialty Mathematics and Physics. In 1991 defended his dissertation for the Candidate degree (Ph.D.) of Physical and Mathematical Sciences. In 1996 received the diploma of Associate Professor on the Department of Computer Science and Teaching Methods. At the present time is working as an Associate Professor on the Department of Computer Science and Teaching Methods Ternopil Volodymyr Hnatiuk National Pedagogical University. Scientific interests: numerical computing methods, optimization methods, e-learning.*