Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп'ютерних систем і мереж

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем
Назва теми

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

КРМКІ. 015047.19.01.14 ПЗ

Виконав: студент 2 курсу, група КІ1м-19-1 Підпис Гончар Р.М.

Керівник доц., к. т. н, доцент кафедри КбКСМ Підпис Чешун В.М.

Нормоконтролер доц., к. т. н, доцент кафедри КбКСМ Підпис Муляр І.В.

До захисту допускаю:
Зав. кафедри КбКСМ, к.т.н., доцент Підпис Кльоц Ю.П.

4 12 2020 р.

Хмельницький, 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем

Кафедра Кафедра кібербезпеки та комп'ютерних систем і мереж

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма Програмування та захист комп'ютерних систем і мереж

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П.Кльоц

" 01 " 09 2020 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гончару Ростиславу Михайловичу

Прізвище, ім'я, по батькові студента

1. Тема роботи Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем

Керівник роботи Чешун Віктор Миколайович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

кандидат технічних наук, доцент

Затверджена наказом № 118 ректора університету додаток №23 від 01.09.2020

2. Строк подання студентом проекту (роботи) на кафедру 20.11.2020

3. Вихідні дані до проекту (роботи) криптографія, шифри зсуву, коди Хаффмена, комп'ютерні системи

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ. Дослідження властивостей способів кодування і методів захисту ресурсів комп'ютерних систем. Математична модель методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена. Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем. Апробація ефективності запропонованого методу і алгоритмів його реалізації. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Загальна характеристика магістерської роботи. Дослідження способів і задач кодування. Дослідження шифрів зсуву і оптимальних кодів Хаффмена. Математична модель методу. Основні положення методу. Алгоритмічна реалізація методу. Апробація методу – вхідні дані. Апробація методу - результати моделювання. Висновки.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Муляр І.В. доцент кафедри КбКСМ		

7. Дата видачі завдання «_2_» _____вересня_____ 2020р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напряму дослідження та узгодження тематики КРМ з керівником	2.02.2020	
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	2.03.2020	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	1.04.2020	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	1.05.2020	
5	Робота над науковою публікацією	1.06.2020	
6	Уточнення і затвердження теми	1.09.2020	
7	Робота над розділом 3 – розробка алгоритмів та технологій, їх аналіз	2.09.2020	
8	Робота над розділом 4 – апробація запропонованих рішень	1.10.2020	
9	Узгодження отриманих; оформлення пояснювальної записки згідно вимог	1.11.2020	
10	Оформлення графічної частини	8.11.2020	
11	Попередній захист роботи	10.11.2020	
12	Захист роботи на засіданні ЕК	5.12.2020	

Студент Гончар Р.М.
Підпис Ініціали, прізвище

Керівник роботи Чечун В.М.
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем

Автор роботи: Гончар Ростислав Михайлович

Керівник роботи: к.т.н., доц. Чешун Віктор Миколайович

Загальний обсяг роботи: 101 сторінка, 18 рисунків, 11 таблиць, 5 додатків, 26 посилань.

КРИПТОГРАФІЯ, ШИФРИ ЗСУВУ, КОДИ ХАФФМЕНА, КОМП’ЮТЕРНІ СИСТЕМИ

Метою кваліфікаційної роботи є підвищення ефективності захисту інформаційних ресурсів комп’ютерних систем за рахунок зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Дана кваліфікаційна робота присвячена розробці методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем, особливістю якого є можливість збільшення стійкості алгоритмів шифрування попередньою підготовкою вхідних даних оптимальним кодуванням. Оптимальне кодування в методі дозволяє не тільки стискати дані, а і забезпечує зміну складу кодового алфавіту алгоритму шифрування і його статистичні характеристики, що ускладнює злам зашифрованих даних.

Дата 20.11.2020р. Підпис студента

ANNOTATION

a qualification work of Honchar Rostyslav

entitled «A method of creating a shift cipher using optimal Huffman coding to increase the efficiency of protecting computer systems.».

Mentor: Ph.D. Cheshun Viktor

Total volume of work: 101 pages, 18 figures, 11 tables, 5 appendices, 26 references.

CRYPTOGRAPHY, SHIFT CODES, HUFFMAN CODES, COMPUTER SYSTEMS

The aim of the thesis is to increase the efficiency of protection of information resources of computer systems by changing the statistical properties of the alphabet codes for cryptographic shift codes using the optimal Huffman coding.

This thesis is devoted to the development of a method of generating shift ciphers using optimal Huffman coding to improve the protection of computer systems, the feature of which is the ability to increase the stability of encryption algorithms by pre-preparation of input data by optimal coding. Optimal encoding in the method allows not only to compress the data, but also provides a change in the composition of the code alphabet of the encryption algorithm and its statistical characteristics, which complicates the hacking of encrypted data.

Date 20.11.2020p. Signature

ЗМІСТ

ВСТУП

Актуальність дослідження. В умовах стрімкого розвитку інформаційних технологій, постійного збільшення обсягів інформації в кіберпросторі і зростання її цінності, а також через появу нових загроз щодо її цілісності і конфіденційності надзвичайної актуальності набувають заходи кібербезпеки.

Одним із базових заходів є криптографічний захист даних, про що свідчить поява і масштабне використання великої кількості методів та алгоритмів симетричного й асиметричного шифрування з різними функціональними можливостями і принципами дії (алгоритми DES-базовий, подвійний і потрійний DES, IDEA, ГОСТ 28147, Діффі-Хелмана, RSA тощо [1]) та спроби їх постійного вдосконалення.

Підвищення криптостійкості алгоритмів шифрування можна досягти попередньою підготовкою вхідних даних, в ході якого забезпечується порушення статистичних даних повторюваності символів вхідного тексту, тобто, збільшення характеристик його ентропії. Одним із варіантів такої підготовки вхідного тексту може бути застування методів оптимального нерівномірного кодування.

Дослідження присвячене питанню розробки методу формування шифрів зсуву із застосуванням оптимального нерівномірного кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Об'єктом дослідження є процес криптографічного шифрування даних в комп’ютерних системах із застосуванням криптографічних шифрів зсуву.

Предметом дослідження є методи, алгоритми і засоби підвищення криптостійкості криптографічних алгоритмів шифрування у застосуванні до криптографічних шифрів зсуву.

Мета кваліфікаційної роботи полягає в підвищенні ефективності захисту інформаційних ресурсів комп’ютерних систем за рахунок зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Для реалізації програми досліджень знадобилося:

а) дослідити використовувані в комп’ютерних системах способи і методи кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів. На підставі виявлених закономірностей визначити перспективні напрями наукових досліджень і сформулювати постановку задачі;

б) розробити математичну модель для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем;

в) визначити основні положення методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем та розробити алгоритми його реалізації;

г) обґрунтувати ефективність запропонованого методу і алгоритмів його реалізації моделюванням.

Методи досліджень базуються на основних положеннях теорії ймовірності та математичної статистики, теорії інформації та кодування, криптографії та прикладної криптології.

Наукова новизна отриманих результатів:

1. Розроблено нову математичну модель, яка враховує особливості сумісного використання методів оптимального кодування та шифрів зсуву для підвищення криптостійкості систем захисту інформаційних ресурсів комп’ютерних систем;

2. Запропоновано спосіб застосування методу оптимального кодування Хаффмена для зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву, який став основою для розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Практична цінність розробленого методу полягає у обґрунтуванні можливості підвищення ефективності криптографічного захисту інформаційних ресурсів комп’ютерних систем застосуванням методів оптимального кодування перед реалізацією криптографічних методів захисту, а також у розробці алгоритмів реалізації методу.

Апробація роботи. Наукові результати і основні положення кваліфікаційної роботи магістра доповідались і обговорювались на всеукраїнській і міжнародній науково-практичних конференціях.

Публікації. За темою кваліфікаційної роботи опубліковано 1 теза доповідей всеукраїнської науково-практичної конференції і 1 стаття у збірнику наукових праць молодих науковців і студентів.

1 ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ СПОСОБІВ КОДУВАННЯ І МЕТОДІВ ЗАХИСТУ РЕСУРСІВ КОМП'ЮТЕРНИХ СИСТЕМ

1.1 Кодування як спосіб захисту інформаційних ресурсів комп'ютерних систем

1.1.1 Кодування як основа магістерського дослідження і загальна класифікація кодів

Дане дослідження присвячене розробці методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

Як видно із загальної спрямованості дослідження, з метою підвищення ефективності захисту комп'ютерних систем в ньому передбачено використовувати два види кодування:

- оптимальне кодування Хаффмена;
- формування шифрів зсуву.

Оптимальне кодування Хаффмена при цьому розглядається як інструмент підвищення криптостійкості шифрів зсуву, через що і планується досягти підвищення ефективності захисту комп'ютерних систем.

З цього можна зробити висновок, що позитивний результат даного наукового дослідження полягає у визначенні принципів поєднання різних видів кодування для підвищення ефективності захисту інформаційних ресурсів комп'ютерних систем, що зумовлює потребу узагальнюючого дослідження використовуваних в комп'ютерних системах способів кодування.

Згідно з наведеними в [2, 3, 4, 5] даними, використовувані в комп'ютерних системах різновиди кодування можна розділити на 4 класи:

– примітивне кодування;

– ефективне або оптимальне кодування;

– завадостійке кодування або кодування з виявленням і виправленням помилок;

– шифрування даних (криптографічні методи захисту).

1.1.2 Примітивні коди і їх призначення

Як видно з наведеної класифікації, найпростішим видом кодування є примітивне.

До визначення примітивного кодування існує 2 підходи.

Примітивне кодування – первинне представлення актуальних інформаційних властивостей певного об’єкта із застосуванням набору символів, що використовуються як засіб кодування [2].

В [5] примітивне кодування описується як безнадлишкове, що погоджує алфавіт джерела з алфавітом дискретного каналу, і, в певних випадках, може використовуватися для шифрування даних з метою захисту від несанкціонованого доступу, а також для підвищення стійкості роботи засобів синхронізації систем зв'язку.

Визначення [5] швидше підходить до опису не класу кодів, а складності їх формування: примітивними можуть вважатися коди, що формуються простими або примітивними процедурами.

В [4,5] також зазначається головна властивість примітивних кодів: окремі примітивних кодів кодові комбінації можуть відрізнятися один від одного лише одним елементом. Через це, навіть один прийнятий з помилкою елемент кодової комбінації призводить до заміни однієї кодової комбінації іншою, тобто, до неправильного прийому відісланої кодової комбінації і сприйняття її за іншу.

Аналіз вказаної властивості свідчить, що вона може бути характерною і для досить складних способів кодування, зокрема, для окремих алгоритмів економного кодування та для алгоритмів шифрування.

На підтвердження цього в [5] описується примітивне шифрування: шифрування (при примітивному кодуванні) - спосіб забезпечення секретності зв'язку, який запобігає розумінню повідомлення несанкціонованим користувачем і введення в систему помилкових повідомлень. У цьому випадку правило примітивного кодування

вибирається так, щоб мінімізувати імовірність одержання на виході кодера довгої послідовності тільки з одиниць або тільки з нулів,. Такий кодер ідентифікують як скремблер (від scramble - перемішувати).

Оскільки підхід [5] стосовно простоти формування і організації коду як ознаки примітивного кодування накладає сферу застосування терміну на інші коди (оптимальні, шифрувальні), цей варіант нас не влаштовує і за основу візьмемо перше розглянуте визначення.

Таким чином, примітивне кодування можна розглядати як базове кодування інформації у вигляді систематизованих даних із застосуванням певного алфавіту.

Примітивне кодування є характерним для людини здавна – кодування інформації із застосуванням символьного (абетка) та цифрового (десяткові числа) алфавітів.

В комп'ютерних системах обидва ці способи виявилися незручними і основою примітивного кодування комп'ютерних даних стала двійкова система.

В загальному, в ході примітивного кодування символів тексту символи кодованого алфавіту представляються із застосуванням набору цифр від 0 до (m-1), де m - основа системи числення. Фактично, символи просто нумеруються за правилами лічби у відповідній системі. Особливість порівняно з простою лічбою – рівний розмір (розрядність) кодових комбінацій. Тобто, Примітивні коди – рівномірні коди, кожен опис символу початкового алфавіту має однакову кількість кодових позицій або розрядів.

В такому випадку розрядність коду n дозволяє створити неповторні комбінації, кількість N яких визначається формулою:

$$N = m^n \qquad (1.1)$$

З формули (1.1) слідує базова формула теорії інформації і кодування, яка дозволяє визначити розрядність повідомлень для заданої кількості різновидів повідомлень джерела:

$$n = \log_m K \qquad (1.2)$$

Вираз (1.2) є не лише формулою обчислення розрядності – він є основою для обчислення кількості інформації в повідомленнях за умови рівної ймовірності їх надходження

Примітивні коди використовується в комп'ютерних системах для представлення чисел, символів тексту, спеціальних знаків. Зокрема, популярною є система кодування ASCII (American Standard Code for Information Interchange - американська стандартна таблиця кодування) символів, що можуть вводитися з клавіатури.

На рис. 1.1. приведена таблиця кодів ASCII для латинської розкладки клавіатури, а на рис. 1.2 – для української.

Класична, американська стандартна таблиця кодування ASCII має розрядність 7 біт на символ, що дає за формулою (1.1) 128 можливих комбінацій двійкового коду. Для іноземних застосувань таблиця ASCII розширена до 8 біт на символ, що дає ще 128 додаткових можливих комбінацій двійкового коду [2].

Існують також національні варіанти таблиць кодів (Code Page або CP)для кирилиці розрядністю 8 біт: CP866 для операційної системи MS DOS, CP1251 для операційної системи Microsoft Windows, КОИ8 від розробників ЄС ЕОМ (в перших русифікованих версіях операційної системи UNIX), CP866 від IBM [6].

символ	10-й код	2-й код	символ	10-й код	2-й код	символ	10-й код	2-й код	символ	10-й код	2-й код
	32	00100000	8	56	00111000	P	80	01010000	h	104	01101000
!	33	00100001	9	57	00111001	Q	81	01010001	i	105	01101001
"	34	00100010	:	58	00111010	R	82	01010010	j	106	01101010
#	35	00100011	;	59	00111011	S	83	01010011	k	107	01101011
$	36	00100100	<	60	00111100	T	84	01010100	l	108	01101100
%	37	00100101	=	61	00111101	U	85	01010101	m	109	01101101
&	38	00100110	>	62	00111110	V	86	01010110	n	110	01101110
'	39	00100111	?	63	00111111	W	87	01010111	o	111	01101111
(	40	00101000	@	64	01000000	X	88	01011000	p	112	01110000
)	41	00101001	A	65	01000001	Y	89	01011001	q	113	01110001
*	42	00101010	B	66	01000010	Z	90	01011010	r	114	01110010
+	43	00101011	C	67	01000011	[	91	01011011	s	115	01110011
,	44	00101100	D	68	01000100	\	92	01011100	t	116	01110100
-	45	00101101	E	69	01000101	]	93	01011101	u	117	01110101
.	46	00101110	F	70	01000110	^	94	01011110	v	118	01110110
/	47	00101111	G	71	01000111	_	95	01011111	w	119	01110111
0	48	00110000	H	72	01001000	`	96	01100000	x	120	01111000
1	49	00110001	I	73	01001001	a	97	01100001	y	121	01111001
2	50	00110010	J	74	01001010	b	98	01100010	z	122	01111010
3	51	00110011	K	75	01001011	c	99	01100011	{	123	01111011
4	52	00110100	L	76	01001100	d	100	01100100	|	124	01111100
5	53	00110101	M	77	01001101	e	101	01100101	}	125	01111101
6	54	00110110	N	78	01001110	f	102	01100110	~	126	01111110
7	55	00110111	O	79	01001111	g	103	01100111	□	127	01111111

Рисунок 1.1 – Таблиця кодів ASCII для латинської розкладки клавіатури

Номер	Двійковий код	Символ	Номер	Двійковий код	Символ	Номер	Двійковий код	Символ
128	10000000	А	139	10001011	Л	150	10010110	Ц
129	10000001	Б	140	10001100	М	151	10010111	Ч
130	10000010	В	141	10001101	Н	152	10011000	Ш
131	10000011	Г	142	10001110	О	153	10011001	Щ
132	10000100	Д	143	10001111	П	154	10011010	Ї
133	10000101	Е	144	10010000	Р	155	10011011	І
134	10000110	Ж	145	10010001	С	156	10011100	Ь
135	10000111	З	146	10010010	Т	157	10011101	Є
136	10001000	И	147	10010011	У	158	10011110	Ю
137	10001001	Й	148	10010100	Ф	159	10011111	Я
138	10001010	К	149	10010101	Х	160	10100000	а

Рисунок 1.2 – Таблиця кодів ASCII для букв української розкладки клавіатури

Міжнародний орган стандартизації ISO затвердив стандартом російської мови стандарт кодування ISO 8859-5 [7].

Зрозуміло, що 256 можливих комбінацій не можуть охопити всі варіації алфавітів, а обмеженість одним алфавітом створює проблеми в епоху глобалізації економіки, стирання міждержавних меж та розвитку глобальних мереж.

З кінця минулого сторіччя для вирішення проблеми стандартизації символьного кодування введено новий міжнародний стандарт, який називається Unicode – шістнадцятирозрядна система кодування, в якій на кожен символ відводиться по два байти пам'яті. Розмір займаної пам'яті збільшується вдвічі, але така кодова таблиця передбачає кодування до 65536 символів. Стандарт Unicode (повна специфікація) включає всі існуючі, вимерлі і штучні алфавіти світу, а також знаки математичних, хімічних, музичних символів тощо [7].

Наведений огляд не є повним описом варіацій примітивного кодування символьних даних, але має за мету продемонструвати важливість досліджуваного питання, що дало велике різноманіття рішень.

Стосовно примітивного кодування цифрової інформації можна в комп’ютерних системах, то основою для такого кодування також є двійкова система, але, через занадто велику розрядність чисел в цій системі та різноманіття вирішуваних задач можна виділити декілька основних класів позиційних систем числення, що використовуються для кодування цифрових даних:

- двійкова система;
- системи з основою 2^n (вісімкова, шістнадцяткова системи тощо);
- десяткова система;
- двійково-десяткова система;
- інші позиційні системи з основою, відмінною від 2^n.

Окрім звичайних позиційних кодів лічби в комп’ютерних системах широко використовуються коди спеціального призначення:

- обернені двійкові коди;
- доповняльні двійкові коди;
- модифіковані двійкові коди;

– коди Грея;

– коди Айкена тощо.

Класифікацію примітивних кодів можна проводити нескінченно довго, але головне, що про них можна сказати:

– примітивні коди існують у величезній кількості;

– примітивні коди мають найрізноманітніше призначення і застосування в комп'ютерних системах;

– безпосередньо застосовувані в комп'ютерних системах коди для представлення будь-яких даних і викання операцій – двійкові або їх модифікації;

– цифро-кодована інформація, зокрема і примітивними кодами, є зручною для зберігання і транспортування;

– властивості примітивних двійкових кодів є основою для застосування найрізноманітніших методів перетворення даних та побудови систем захисту інформаційних ресурсів.

1.1.3 Ефективне кодування та його роль в захисті інформаційних ресурсів

Термін ефективне кодування виникає з положення, що коди цього класу забезпечують ефективне використання каналів зв'язку при передачі інформації.

Якщо говорити точніше, то ефективне кодування забезпечує збільшення ефективності використання ресурсів каналів зв'язку при передачі закодованої з їх використанням інформації, ніж це могло б бути без використання цих кодів.

Зрозуміло, що ефективне кодування не впливає на самі канали зв'язку, а працює лише представленою в певній системі кодування інформацією. Апріорно, це інформація, для кодування якої використано примітивне кодування.

Через відсутність процедур урахування статистичних властивостей символів (повідомлень), що проходять процедури примітивного кодування,

результат отримується дійсно «примітивним» для передачі інформації. Як зазначалося, міжнародний стандарт Unicode включає всі існуючі, вимерлі і штучні алфавіти світу, а також знаки математичних, хімічних, музичних символів тощо та інших символів, що досягається великою розрядністю кодових комбінацій. Звідси виникає питання, чи не доцільніше відкинути «мертві» в застосуваннях комп'ютерних систем символи задля зменшення розрядності коду і отримання більш ефективного способу кодування цих символів з точки зору ефективності використання ресурсів каналів зв'язку при передачі закодованої з їх використанням інформації.

Це примітивний приклад можливості ефективного кодування. Насправді, ефективне кодування значно складніше і враховує ступінь «мертвості» і «живучості» символів в текстах – їх статистичні характеристики. Це дозволяє робити інформацію «стиснутою», що і дає підвищення ефективності використання ресурсів каналів зв'язку при передачі закодованої інформації.

Оскільки в комп'ютерних системах інформація кодується двійковими кодами і стосовно двійкових кодів буде розглядатися метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, то зупинимось саме на аналізі особливостей утворення двійкових ефективних кодів.

Відношення (у відсотках) вихідного коду до вхідного визначають коефіцієнтом стиснення коду:

$$k = \frac{R_{\text{вих}}}{R_{\text{вх}}} * 100\%. \qquad (1.3)$$

Ефективні коди можна розділити на 2 категорії:

- ентропійні коди;
- словникові коди.

Хоча в деяких джерелах ентропійні коди розглядають як статистичні, з певної точки зору і деякі словникові ефективні коди оперують з статистичними даними і можуть розглядатися як статистичні.

Класичними (швидше, за тим, що їх історія виникнення базується на розробках класиків теорії інформації і кодування) ефективними кодами вважаються саме ентропійні коди.

Ентропійним кодуванням називається таке, при якому символи вихідного алфавіту кодують комбінаціями коду різної довжини так, щоб ентропію коду вихідних символів кодера наблизити до максимально можливої [7,8].

Згідно [9] економне кодування направлено на те, щоб переданий дискретний сигнал мав максимальну ентропію, тобто максимальну середню кількість інформації на символ. Тоді кажуть, що отримано повідомлення, в якому повністю усунуто надлишковість кодування. Для передачі отриманого повідомлення каналом знадобиться мінімальна смуга частот. Основна властивість повідомлення, в якому повністю усунена надлишковість кодування – рівна ймовірність і незалежність появи в послідовності символів.

Ентропія залежить від імовірності надходження символів алфавіту в кодованому повідомленні, тому кодові комбінації, що зустрічаються частіше кодуються коротшими кодовими комбінаціям, а ті, що зустрічаються рідше – довшими.

Ентропійні коди також називають префіксними.

Префіксність кодів забезпечує можливість розрізнення окремих символів в двійковому потоці закодованих даних.

Суть префіксності – жоден символ не повинен кодуватися кодом, який є початком коду іншого символу. Тобто, жоден код не повинен бути префіксом іншого коду.

До числа ентропійних (префіксних) ефективних кодів відносять три основних їх різновиди:

– код Шеннона-Фано;
– код Хаффмена;
– арифметичні коди.

Код Шеннона-Фано – один із перших ентропійних економних двійкових кодів.

Код Шеннона-Фано будується за наступним алгоритмом [7]:

– символи алфавіту упорядковують в порядку зменшення імовірностей появи;

– отриману упорядковану множину ділимо на 2 підмножини з максимально близькими сумами ймовірностей;

– символам однієї підмножини присвоюється символ «0», а іншої – символ «1».

– кожна підмножина повторно ділиться на 2 підмножини.

– процес повторюється, поки в кожній утвореній підмножині не залишиться 1 символ.

Кодування Шеннона-Фано зручно унаочнювати в таблицях перетворень.

Розглянемо кодування Шеннона-Фано для передачі повідомлень, що складаються з символів *А*, *Б*, *В*, *Г*, *Д*, *Е*. Імовірності появи символів приймемо рівними: $p(A) = 0.26$, $p(Б) = 0.21$, $p(B) = 0.10$, $p(Г) = 0.07$, $p(Д) = 0.14$, $p(E) = 0{,}25$.

Процес формування коду Шеннона-Фано за алгоритмом відповідно до поставлених умов відображено в таблиці 1.1 (символи, які додаються до кодів, підкреслено).

За таблицею 1.1 отримано кодові комбінації етропійного коду Шеннона-Фано: *А* – 00, *Б* – 10, *В* – 1110, *Г* – 1111, *Д* – 110, *Е* – 01.

Таблиця 1.1 – Формування коду Шеннона–Фано

Етап 1		Етап 2			Етап 3			Етап 4			Етап 5	
Символ	$p_{сим}$	$p_{сим}$	p_{Σ}	Код	$p_{сим}$	p_{Σ}	Код	$p_{сим}$	p_{Σ}	Код	$p_{сим}$ (p_{Σ})	Код
А	0,26	0,26	0,51	0	0,25	0,26	**00**					
Е	0,25	0,25		0	0,25	0,25	**01**					

Б	0,21	0,21	0,5	1	0,21	0,21	**10**					
Д	0,14	0,14		1	0,14	0,3	11	0,14	0,14	**110**		
В	0,10	0,10		1	0,10		11	0,10	0,14	111	0,10	**1110**
Г	0,07	0,04		1	0,04		11	0,04		111	0,04	**1111**

В [7] описується спосіб кодування кодом Шеннона-Фано за допомогою кодового дерева. Побудова дерева йде від кореня, який відповідає сукупності всіх символів, до листя, яке відповідає окремим символам алфавіту.

Розглянемо приклад побудови кодового дерева кодування Шеннона-Фано для передачі повідомлень, що складаються з символів *А*, *Б*, *В*, *Г*, *Д*. Імовірності появи символів приймемо рівними: $p(A) = 0.35$, $p(Б) = 0.17$, $p(B) = 0.17$, $p(Г) = 0.16$, $p(Д) = 0.15$.

Для формування коду символу необхідно йти від «кореня» до відповідного «листа». За деревом отримано кодові комбінації етропійного коду Шеннона-Фано: *А* – 00, *Б* – 01, *В* – 10, *Г* – 110, *Д* – 111.

Недоліком коду Шеннона-Фано є те, що він не завжди дає оптимальний варіант кодування, хоча і дозволяє отримати оптимальне рішення для більшості застосувань.

Через зазначений недолік код Шеннона-Фано зараз не використовується, чому сприяла поява більш досконалого ентропійного ефективного коду – коду Хаффмена.

Код Хаффмена також ідентифікують як вдосконалення коду Шеннона-Фано і називають кодом Шеннона-Фано-Хаффмена.

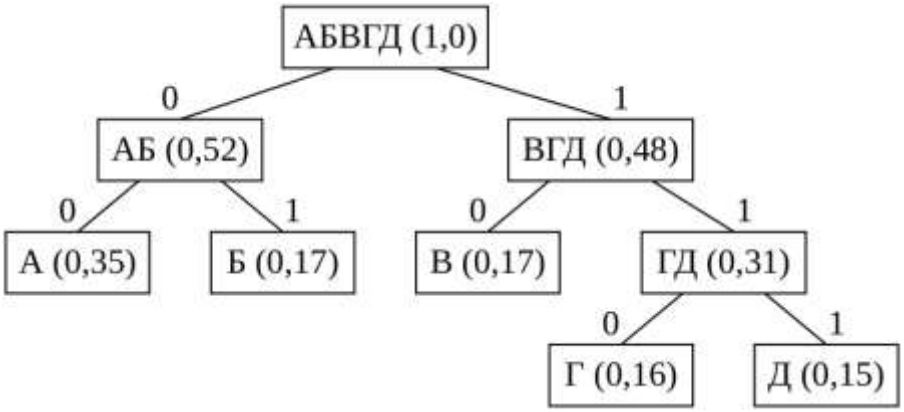


Рисунок 1.3 –Дерево для побудови етропійного коду Шеннона–Фано

Одна з відмінностей коду Хаффмена порівняно до коду Шеннона-Фано в тому, що побудова дерева йде не від кореня до листя, а в зворотному порядку – від листя до кореня.

Алгоритм кодування Хаффмена базується на наступних положеннях [7]:

– символи алфавіту утворюють список вузлів-листів дерева, вага вузлів-листів дорівнює ймовірності символу;

– два вільних вузла дерева з найменшими вагами об'єднуються в новий вузол (батьківський) з вагою, що є сумою ваг об'єднаних вузлів;

– батьківський вузол додається до переліку вільних вузлів, а його нащадки видаляються зі переліку;

– одній гілці, що відходить з батьківського вузла, присвоюється символ «0», а іншій – символ «1»;

– процес триває поки в списку не залишиться єдиний кореневий вузол дерева.

Головною відмінністю коду Хаффмена від коду Шеннона-Фано є те, що він завжди дає оптимальний варіант ентропійного кодування за статистичними даними.

Оскільки коду Хаффмена є базовим для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем, більш детальний його аналіз проведемо в подальшому.

Ще один спосіб ентропійного кодування – арифметичне [2, 12].

При арифметичному кодуванні вхідний текст представляється дійсними числами в межах від 0 до 1. В процесі кодування інтервал відображення тексту в діапазоні зменшується, а кількість біт його представлення збільшується. Чергові символи коду тексту скорочують величину інтервалу відповідно до з значень їх ймовірностей. Більш ймовірні

символи впливають на зміни менше, ніж менш ймовірні (вони додають менше розрядів до результату).

Методи Шеннона-Фено, Шеннона-Фено-Хаффмена, арифметичне кодування – це методи статистичного кодування.

Словникові методи базуються на інших принципах, а саме - на формуванні словників і посилань.

Їх ділять на дві категорії:

– поточні;

– словникові.

Поточним методом кодування є кодування довжин серій (RLE - Run-Length Encoding).

RLE – простий і розповсюджений алгоритм стиску потокових даних. В RLE послідовності повторюваних символів замінюється одним символом і числом його дублювань.

Припустимо, що для зберігання кожного символу відводиться 1 байт коду. Тоді послідовність «EEEEEEE» передбачає зберігання семи байт. За RLE її потрібно замінити на «7A» - всього два байти.

RLE тим ефективніший, чим довші послідовності однакових символів. Відповідно, недолік RLE - неефективність на послідовностях без повторів символів.

Словникові методи кодування починають історію з 1977 року – тоді вийшла в світ робота двох ізраїльських вчених Лемпеля і Зіва. Так з’явився метод кодування Лемпеля-Зіва або LZ-метод.

Суть LZ-методу – фрази в тексті замінюються покажчиком на місце, де вони в тексті з'являлися раніше.

Перевага LZ-методу – він швидко адаптується до структурної будови тексту і може виявляти короткі слова, що з'являються найчастіше. Фрагменти тексту можуть утворюватися і з частин слів.

Словник фрагментів доповнюється вказівниками на джерела (місце в тексті) повторюваних символів. Сам словник LZ-методу міститься в закодованому повідомленні.

Декодування стисненого тексту LZ-методу здійснюється протилежною заміною вказівників на символи (слова та фрази тощо) зі словника, на які вказівники посилаються. LZ-метод забезпечує досить якісне стиснення і дуже швидку процедуру декодування.

На сьогодні метод Лемпеля-Зіва перетворився в повноцінне сімейство алгоритмів LZ-стиснення.

Класичний алгоритм Лемпеля-Зіва простий і може описуватися правилом – якщо в розглянутому раніше коді вже зустрічалася певна сукупність символів, її довжина і зміщення від поточної позиції коротші самої сукупностс, то до файлу записується посилання зміщення-довжина, а не ця сукупність.

За алгоритмом Лемпеля-Зіва LZ77 послідовність «EEEEEEE» потрібно замінити на «E(-1,7)».

Основу алгоритму Лемпеля-Зіва LZ77 складають 4 принципи [2, 5]:

– кожна чергова закодована сукупність символів додається до раніше закодованої послідовності символів таким чином, що разом з ними вона утворює розкладання всієї вихідної послідовності на незбіжні між собою фрази;

– спосіб розкладання зберігається в пам'яті і використовується в подальшому в якості словника;

– кодування здійснюється за допомогою покажчиків на фрази з уже сформованого словника;

– кодування - динамічна процедура, орієнтована на блоки. процес кодування може бути продемонстрований двома вікнами, що містять поточний словник фраз і вікно перегляду.

Версія алгоритму LZ78 застосовується і для стиску двійкових даних. LZ78 зберігає словник з вже переглянутих фраз. На старті алгоритму словник містить 1 порожній рядок. Алгоритм зчитує символи до накопичення послідовності, що входить цілком в одну з фраз словника. Як тільки послідовність перестане відповідати хоча б 1 фразі словника, алгоритм генерує код з індексу послідовності в словнику, яка до останнього

символу містила вхідну послідовність, і символу, який порушив збіг. Вона додається в словник. Коли словник заповнюється, з нього видаляють найменш використовувані послідовності.

Важливим є розмір словника у фразах, оскільки кожен код за LZ78 містить номер фрази в словнику. Звідси слідує, що ці коди мають постійну довжину, рівну округленому в більшу сторону бінарного логарифму розміру словника (кількість біт в байт-коді розширеного ASCII).

У 1984 р Велчем (Welch) модифікацією LZ78 був створений алгоритм LZW (алгоритм Лемпеля-Зіва-Велча - Lempel-Ziv-Welch, LZW), найбільш досконалий на той час алгоритмів LZ-сімейства.

За алгоритмом LZW символи вхідного потоку послідовно зчитуються і відбувається перевірка, чи існує в таблиці послідовностей ця послідовність. Якщо існує, зчитується наступний символ, а якщо ні, в потік заноситься код для попередньої знайденої послідовності, рядок заноситься в таблицю, а пошук починається знову.

Якщо стискають двійкові байтові дані, то послідовностей в таблиці виявиться 256. Якщо використовується 10-бітний код, то під коди рядків беруть значення від 256 до 1023.

Особливість алгоритму LZW – для декомпресії немає потреби зберігати таблицю рядків в файл для розпакування. Алгоритм побудований таким чином, що є можливість відновити початковий текст, користуючись тільки стиснутим потоком.

Високоефективним є двоступеневе кодування інформації [7], коли вихідний потік кодувальника алгоритму Лемпеля-Зіва або Лемпеля-Зіва-Велча надходить на вхід блоку арифметичного кодера (LZARI) або кодера Хаффмена (LZHUF). Подібний підхід використовується в архіваторах LHA, ARJ, PKZIP, RAR і робить їх одними з найбільш ефективних і популярних на сьогодні.

В підсумку зазначимо, що ефективні алгоритми безпосередньо не призначені для вирішення задач захисту даних. Навпаки, з певної точки зору ці алгоритми призводять до збільшення уразливості стиснутих кодів –

помилка в окремому біті коду може викривляти декілька символів закодованого тексту. В той же час, саме стиснення даних дає позитивний ефект для захисту інформаційних ресурсів комп’ютерних систем – чим менше розмір повідомлення, що передається, тим менша імовірність втрат інформації при передачі. Відхилення варіанту кодування від примітивного кодування символів також ускладнює дешифрування тексту. При цьому очікуваною є підвищення ефективності алгоритмів шифрування із застосуванням методів оптимального кодування для захисту ресурсів комп’ютерних систем.

1.1.4 Завадостійке кодування в захисті комп’ютерних систем

Завадостійке кодування передбачає те, що в потік переданих символів вводяться додаткові (надлишкові) символи для виправлення помилок, які виникають на приймальній стороні [12, 16-19]. Це вимагає збільшення швидкості передачі даних каналом, що, при обраному типі модему, еквівалентно розширенню смуги частот сигналу і зменшення енергії посилки. Тому може виникнути правомірне запитання про доцільність використання надлишкового кодування. На це питання дає відповідь теорема Шеннона про пропускну здатність безперервного каналу зв'язку, з якої випливає, що пропускна здатність безперервного каналу збільшується з розширенням його смуги, але при оптимальному кодуванні. Тому слід очікувати підвищення достовірності передачі при заданій швидкості і відношенні сигнал/шум в каналі при внесенні надмірності.

Завадостійкі коди можна розділити на чотири категорії:

- блокові;
- безперервні;
- систематичні;
- нероздільні.

У найпростішому випадку [9], групі з k символів джерела ставиться у відповідність n символів, що передаються по каналу. Такий код називається блоковим і записується умовно як (n, k) код.

Можливе також використання безперервних кодів, яке характеризуються тим, що операції кодування і декодування виробляються над безперервною послідовністю символів без розбивки на блоки.

Якщо до символів джерела додаються надлишкові символи, то код називають систематичним.

Якщо групі інформаційних символів ставиться у відповідність нова група символів, що передається по каналу, в якій інформаційних символів в явному вигляді немає, то коди класифікують як нероздільні.

Побудова коду із заданою здатністю виявляти або виявляти та виправляти помилки зводиться до забезпечення необхідної кодової відстані через введення надлишковості [9,10]. При цьому переслідується додаткова умова – кількість використовуваних перевірочних символів має бути мінімальною.

Завдання визначення мінімально-необхідної кількості використовуваних перевірочних символів для забезпечення заданої кодової відстані не є вирішеним. Є лише ряд оцінок для одержання максимального значення кодової відстані при фіксованих розрядностях кодів [18].

Найпростіший спосіб завадостійкого кодування двійкових посилок - кодування з перевіркою на парність (альтернатива - кодування з перевіркою на непарність).

Кодування з перевіркою на парність (кодування з перевіркою на непарність) - це простий метод для виявлення помилок в переданому пакеті даних [11]. За допомогою даного коду неможливо відновити дані, лише можливо виявити одиночну помилку або непарну кількість викривлень.

Кодування з перевіркою на парність у кожному пакеті даних додає один біт парності (паритетний біт). Цей біт встановлюється під час запису (або відправки) даних, а потім, під час читання (отримання) даних, розраховується повторно і порівнюється. Цей біт визначається сумою за модулем 2 всіх біт даних в пакеті. Структура кодуючого пристрою для кодування з перевіркою на парність наведена на рис.1.4.

Число одиниць в правильному пакеті завжди буде парне. Зміна контрольного розрахункового біта (з 0 на 1 чи навпаки) свідчить про наявну помилку.

Структура декодуючого пристрою для кодування з перевіркою на парність наведена на рис.1.5.

Якщо взяти початковий код 0001, то закодований варіант 00011 (0 + 0 + 0 + 1 = 1 (mod 2)). У випадку, що спотворено один розряд (3-й, для прикладу), прийнятий код буде 00111. У прийнятому коді кількість одиниць не є парною – декодуючий пристрій виявить помилку.

Як говорилося раніше, метод кодування з перевіркою на парність (кодування з перевіркою на непарність) служить тільки для визначення одиночної помилки. У разі зміни стану пари бітів виникає ситуація, коли розрахунок контрольного біта співпадає з записаним. В цьому випадку система не визначить помилку.

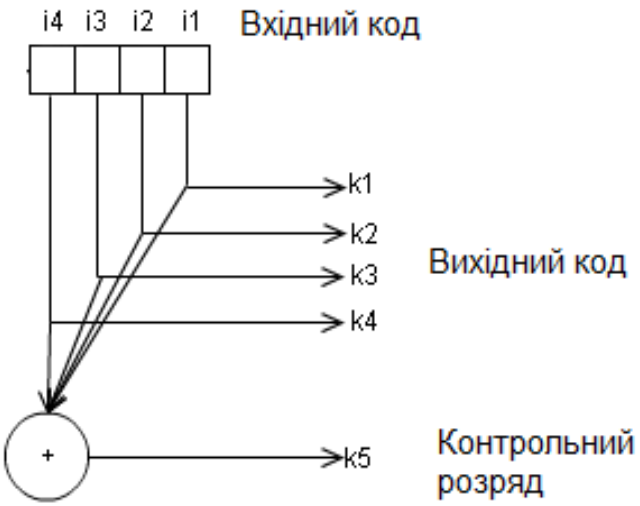


Рисунок 1.4 – Кодуючий пристрій кодування з перевіркою на парність

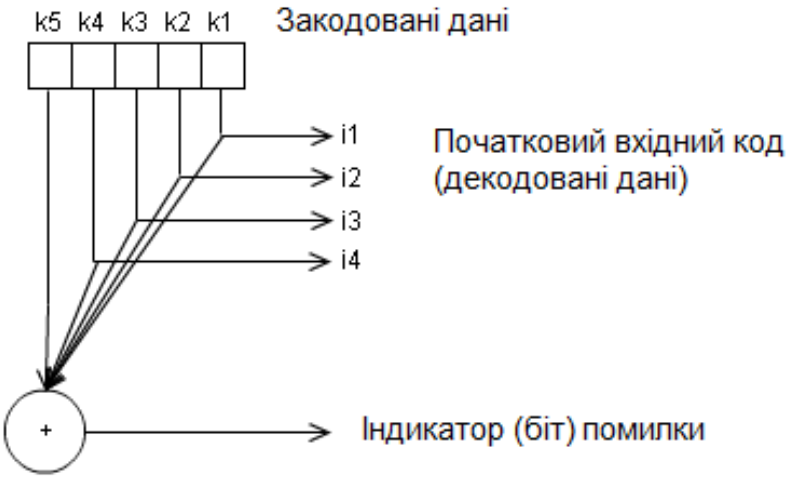


Рисунок 1.5 – Декодуючий пристрій кодування з перевіркою на парність

Якщо попередньому прикладі прийнятий код 00100 (спотворені розряди 3 і 4), у прийнятому коді кількість одиниць стає парною – декодуючий пристрій не виявить помилку.

Оскільки близько 90% всіх нерегулярних помилок відбувається саме з одиночним розрядом [12, 13, 14], перевірки парності буває досить для більшості ситуацій.

Якщо виникає потреба виявлення більшої кількості помилок, використовуються більш складні способи кодування.

Таким кодом є код Хеммінга [2, 14].

Код Річарда Хеммінга. забезпечує виявлення і виправлення одиночних помилок при мінімально можливому числі додаткових перевірочних розрядів [11, 15]

Структура кодуючого пристрою коду Хеммінга (7, 4) наведена на рис.1.6.

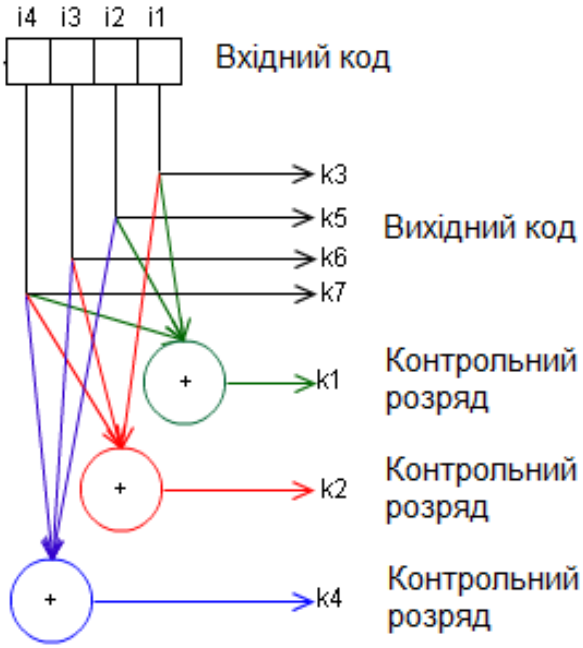


Рисунок 1.6 – Кодуючий пристрій коду Хеммінга (7, 4)

Для перевірочних символів коду Хеммінга використовується спеціальне маркування виду (k, i), де k - кількість символів в повідомленні, i - кількість інформаційних символів в повідомленні. Існують коди (7, 4), (15, 11), (31, 26).

Кожен контрольний символ коду Хеммінга є сумою за модулем 2 розрядів початкового коду, що знаходяться на певних позиціях розрядної сітки. Контрольні розряди також розміщуються на фіксованих позиціях розрядної сітки повідомлення.

На етапі декодування кожен контрольний символ коду Хеммінга додається за модулем 2 до розрядів початкового коду, що знаходяться на певних позиціях розрядної сітки і використовувались для утворення цього розряду. Отримання нульових результатів свідчить про відсутність помилок передачі.

Особливість коду Хеммінга – якщо в повідомленні виникне одна помилка, то отримуваний при декодуванні код однозначно вказує на місце її виникнення.

Код Хеммінга відноситься до систематичних (роздільних) кодів.

Широкого застосування набули циклічні коди (є нероздільні і роздільні циклічні завадостійкі коди).

Циклічні коди названі так тому, що частина їх комбінації можуть бути отримані циклічним зсувом (зліва направо) однієї або декількох комбінацій коду. Ідея побудови циклічних кодів базується на використанні незвідних у полі двійкових чисел многочленів, які називаються утворюючими поліномами. Ознакою незвідності є властивість многочлена ділитися без залишку лише на себе та на одиницю (в двійковому коді з урахуванням особливостей виконання операції ділення) [2,18,19].

Циклічний код повідомлення можна отримати множенням його значення на утворюючий поліном, але недоліком формування циклічного коду методом множення поліномів є відсутність можливості розділення в закодованому повідомленні інформаційних та контрольних розрядів. Цього недоліку позбавлений метод формування циклічного коду діленням поліномів.

Формування циклічних кодів також може бути реалізоване діленням поліномів.

Формування методом ділення поліномів циклічного коду многочлена $U(x)$ за допомогою утворюючого многочлена $K(x)$ виконується за наступним алгоритмом:

– многочлен $U(x)$ множиться на одночлен x^n, де n дорівнює степеню многочлена $K(x)$. Операція множення на одночлен призводить до додавання в молодші розряди двійкового коду многочлена $U(x)$ нулів у кількості, яка дорівнює степеню одночлена;

– виконується ділення отриманого многочлена $U(x)\cdot x^n$ на утворюючий многочлен $K(x)$ до отримання залишку $Q(x)$, степінь якого менша степені утворюючого многочлена $K(x)$;

– виконується операція додавання за модулем 2 многочлена $U(x)\cdot x^n$ та залишку $Q(x)$ (на місце доданих в результаті виконання операції $U(x)\cdot x^n$ нулів записується n розрядів залишку, які і є контрольними).

Перевірка на наявність помилки повідомлення, закодованого циклічним кодом, зводиться до повторного його ділення на утворюючий поліном з метою обчислення залишку. Якщо отримане значення залишку дорівнює нулю, то повідомлення вважається прийнятим правильно, інакше фіксується помилка [2].

У підсумку аналізу завадостійких кодів зазначимо, що, вони безпосередньо для підвищення ефективності захисту даних в каналах зв'язку комп'ютерних систем, але цей захист не збільшує криптостійкість коду. Певним виключенням є нероздільні завадостійкі коди, але загальновідомі алгоритми їх кодування та декодування не дозволяють розглядати ці коди як коди шифрування даних від зловмисного їх використання.

1.1.5 Криптографічні методи захисту інформаційних ресурсів

Одна із найбільш повних класифікацій криптографічних методів захисту наведена в [1, 6, 7, 13], тому в розгляді даного питання будемо спиратися на матеріал даного посібника.

Криптографічні методи захисту асоціюють з шифруванням даних.

Мета шифрування даних – представлення даних в такому вигляді, щоб сторонні особи, перехопивши їх, не могли розпізнати і використати їх зміст, а адресати мали змогу відновити передану інформацію повністю.

Криптографія спирається на використанні шифрів та ключів, які застосовуються в процесах шифрування і дешифрування інформаційних повідомлень.

Існує безліч криптографічних шифрів з різними принципами дії і властивостями. Розглянемо особливості деяких типових способів шифрування інформації.

Шифр Цезаря отримав свою назву на честь імператора Гая Юлія Цезаря, який застосував цей шифр для забезпечення секретності листування з генералами своєї армії.

Цей шифр реалізує перетворення відкритого тексту, при якому кожна буква відкритого тексту замінюється іншою буквою, що знаходиться в алфавіті на заданій відстані від початкової. Наприклад, кожна буква може замінюватись третьою після неї. Таким чином, в ході застосування шифр Цезаря ніби зсуває символи тексту на основі таблиці алфавіту, тому його називають шифром зсуву. Алфавіт використовується циклічно.

Шифр Цезаря надзвичайно простий як у застосуванні, але не є стійким. На його основі розроблено цілу серію вдосконалених шифрів.

Шифр Віженера є модифікованим шифром Цезаря зі змінною величиною зсуву.

Шифр Цезаря і шифр Віженера базуються на прямій заміні (моноалфавітна підстановка) - букви шифрованого тексту замінюються іншими буквами того ж самого або деякого іншого алфавіту.

Пряма заміна або моноалфавітна підстановка, коли букви шифрованого тексту замінюються іншими буквами того ж самого або деякого іншого алфавіту, є простою як в реалізації, так і для розкриття шифру.

Для підвищення стійкості шифрів використовують поліалфавітні підстановки, в яких для заміни символів початкового тексту використовуються символи декількох алфавітів (вони можуть бути отримані з одного застосуванням різних варіантів його перетворення).

Оскільки шифри зсуву є основою методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем, більш детальний аналіз цього виду шифрів проведемо окремо.

Інший клас шифрів застосовують методи шифрування з перестановкою.

Суть методів шифрування з перестановкою полягає в тому, що символи шифрованого тексту переставляються за певними правилами всередині блоку символів.

Найчастіше зустрічаються три види перестановки [1]:

– проста;

– ускладнена за таблицею;

– ускладнена за маршрутами.

Шифрування простою перестановкою здійснюється за алгоритмом:

– вибирається ключове слово з символами, що не повторюються;

– шифрований текст записується послідовними рядками під символами ключового слова;

– зашифрований текст виписується колонками в тій послідовності, в якій розташовуються в алфавіті букви ключа (або в порядку проходження цифр в натуральному ряду, якщо він цифровий).

Недолік шифрування простою перестановкою – при великій довжині відкритого тексту в зашифрованому повідомленні можуть виявитися закономірності символів ключа. Протидією цьому періодична зміна ключа. При достатньо частій зміні ключа стійкість шифрування може суттєво підвищуватися, але при цьому ускладняється шифрування і дешифрування.

Ускладнений метод перестановки за таблицями полягає в тому, що для запису символів шифрованого тексту використовується спеціальна таблиця, в яку введені деякі ускладнюючі елементи.

Таблиця є матрицею, розміри якої можуть бути вибрані довільно. В таблицю записуються знаки тексту. Ускладнення полягає в тому, що певна кількість клітинок таблиці заблокована і не використовується. Кількість і розташування заблокованих позицій є додатковим ключем шифрування. Відкритий текст блоками з ($m \times n$-S) елементів записується в таблицю ($m \times n$ – розміри таблиці, S – кількість заблокованих позицій). Далі процедура шифрування аналогічна простій перестановці.

Ускладнений метод перестановок за маршрутами типу гамільтоновських забезпечує високу стійкість шифрування. При цьому для запису символів тексту використовуються вершини деякого графа, а шифри тексту визначаються за маршрутами.

Якісний криптографічний захист шифрування за допомогою аналітичних перетворень. Для цього можна застосовувати методи алгебри матриць, наприклад, множення матриці $||a_{ij}||$ на вектор b_j за правилом:

$$C_j = \left\| a_{ij} \right\| b_j = \sum_j a_{ij} b_j. \qquad (1.4)$$

Якщо в якості ключа використовувати матрицю $||a_{ij}||$, а замість компонента вектора b_j підставити символи початкового тексту, то компоненти вектора C_j будуть символами зашифрованого тексту.

Дешифрування здійснюється також з використанням множення матриці на вектор, тільки за основу береться матриця, зворотна матриці шифрування, а за вектор-співмножник – відповідна кількість символів зашифрованого тексту.

Шифрування методом гамірования полягає в тому, що символи відкритого тексту послідовно складаються з символами деякої спеціальної послідовності, яка називається гамою. Іноді такий метод представляють як накладення на початковий текст певної гамми, тому він одержав назву "гамірования".

Якщо довжина гамми менша за довжину повідомлення, яке підлягає шифруванню, гамма використовується повторно. Властивість гамми щодо можливостей її застосування без повторів характеризують як тривалість періоду гамми.

Процедуру накладення гамми на початковий текст можна здійснити двома способами.

– заміною символів початкового тексту і гамми цифровими еквівалентами, які потім додаються за модулем *K;*

– представленням символів початкового тексту і гамми у вигляді двійкового коду, розряди якого і гамми додаються за модулем 2.

Стійкість шифрування методом гамірування визначається властивостями гамми. За хороших статистичних властивостей гамми стійкість шифрування визначається тільки довжиною її періоду.

Достатньо ефективним засобом підвищення стійкості шифрування є комбінування різних способів шифрування, тобто, послідовне (дублююче) шифрування початкового тексту за допомогою двох або більше методів.

Типовим прикладом комбінованого шифру є національний стандарт США криптографічного закриття даних (DES).

Алгоритм DES та багато інших методів з'явились завдяки розвитку інформаційних технологій і орієнтовані на використання в комп'ютерних системах. Алгоритм DES відноситься до категорії алгоритмів симетричного шифрування (алгоритми традиційної криптографії).

Симетричне шифрування - алгоритм, що отриує код відкритого тексту повідомлення *P* і ключ *K*. Поєднання коду повідомлення *P* і ключа *K* за правилами шифрування дає зашифрований код повідомлення *C*. Ключ шифрування не залежить від вмісту повідомлення. Зміна ключа *K* призводить до зміни коду повідомлення *C*.

Альтернатива алгоритмів симетричного шифрування – алгоритми асиметричного шифрування або алгоритми шифрування з відкритим ключем [1, 6].

Шифрування з відкритим ключем має суттєві відмінності від симетричного шифрування:

– у симетричному шифруванні є 1 секретний ключ, а при асиметричному шифруванні два ключі – відкритий і закритий;

– у симетричному шифруванні ключ універсальний для шифрування і дешифрування, а при асиметричному мають різне призначення – відкритий для шифрування, а закритий ключ для дешифрування.

Відкриті ключі надаються всім у вільний доступ. Закриті ключі створюються кожним учасником і є таємними.

У підсумку аналітичного дослідження криптографічних шифрів можна зробити висновки, що існує велика їх кількість (рис. 1.7.) з різними принципами дії і властивостями, що унеможливлює їх огляд в даній роботі.

Криптографічні шифри є основним інструментом захисту секретного вмісту повідомлень між двома сторонами, що є офіційними користувачами інформації, тому вони постійно вдосконалюються.

З іншої сторони, постійно вдосконалюються і методи зламу шифрів, створювані з наміром несанкціонованого заволодіння інформацією.

Через постійну конкуренцію між засобами захисту інформації і засобів зламу цього захисту задача вдосконалення методів криптографічного захисту ніколи не втратить актуальності.

За результатами дослідження методів кодування як засобу і способу захисту інформаційних ресурсів комп'ютерних систем можна зробити висновки, що, хоча розглянуті способи кодування мають різне призначення, але всі вони в певному аспекті використовуються як інструмент захисту цілісності і конфіденційності даних, з чого слідує перспективність поєднання властивостей різних видів кодування для підвищення ефективності захисту інформаційних ресурсів комп'ютерних систем.

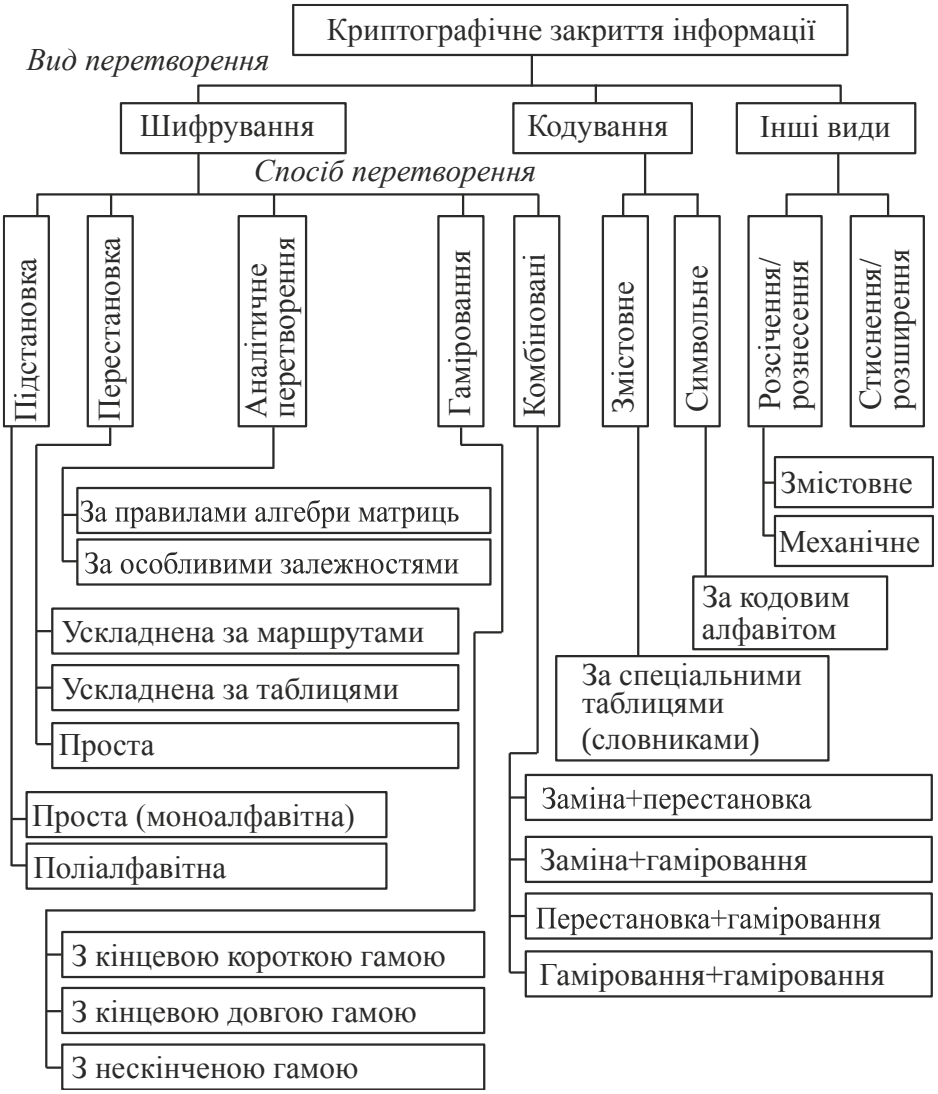


Рисунок 1.7 – Методи криптографічного захисту ресурсів комп’ютерних систем

1.2 Дослідження криптографічних властивостей шифрів зсуву

Як зазначалося в попередньому розділі, найпростішим шифром зсуву є шифр Цезаря.

Шифр Цезаря є також класифікують як шифр заміни [1, 6, 7]. Найточнішим ідентифікатором шифру Цезаря і цілого класу побудованих на його основі шифрів є їх опис як шифрів заміни зі зсувом алфавітів.

Шифр Цезаря шифр реалізує перетворення відкритого тексту, при якому кожна буква відкритого тексту замінюється іншою буквою, що знаходиться в алфавіті на заданій відстані від початкової. Таким чином, в

ході застосування шифр Цезаря ніби зсуває символи тексту на основі таблиці алфавіту, алфавіт при цьому використовується циклічно. Саме тому його називають шифром зсуву.

Секрет шифру Цезаря - напрямок і відстань зсуву алфавіту, знання яких дозволяє легко відтворювати відкритий текст зашифрованого повідомлення.

Розглянемо приклад шифру Цезаря при використанні зсуву в 5 символів. Український алфавіт в такому шифрі перетворюється в шифр заміни, відображений в таблиці 1.2.

Фраза «метод формування шифрів зсуву» після застосування даного шифру Цезаря зі зміщенням 5 розрядів перетвориться в шифровану: «січуи щухсшєдттґ алщхмє кцшєш».

Таблиця 1.2 – Таблиця шифру Цезаря зі зміщенням 5 розрядів (український алфавіт)

Вид алфавіту	Склад алфавіту
Алфавіт основний	а б в г ґ д е є ж з и і ї й к л м н о п р с т у ф х ц ч ш щ ь ю я
Алфавіт шифру	д е є ж з и і ї й к л м н о п р с т у ф х ц ч ш щ ь ю я а б в г ґ

Кидається в очі абсурдність отриманого шифрованого тексту і, на перший погляд, складність його зламу без даних таблиці шифрування.

Зрозуміло, що подібний захист міг залишатися ефективним лише за умов відсутності вмінь, знань і засобів злому шифрів. Сучасні методи криптоаналізу дозволяють з легкістю встановити закономірності шифрування зсувом і зламати шифр.

Першочергово кидається в очі наявність двох літер «тт» в закодованому слові «щухсшєдттґ». В українських словах найчастіше таке сполучення характерне для літер «нн», рідше «тт». Після літер «нн» і «тт» найчастіше використовується літера «я», рідше «і».

Таким чином, елементарний зовнішній аналіз зашифрованого тексту за наявності інформації про застосування шифру Цезаря дозволяє робити гіпотези, які призводять до зламу шифру.

Насправді, злам шифру Цезаря робиться ще простіше, якщо враховувати статистичні дані щодо частоти входження символів алфавіту в тексти певної мови.

На рис. 1.8 зображено гістограму частоти входження символів латинського алфавіту в тексти англійської мови.

За даними подібної статистики зашифрований текст відновлюється без особливих проблем за умови наявності тексту (декількох текстів) достатніх розмірів.

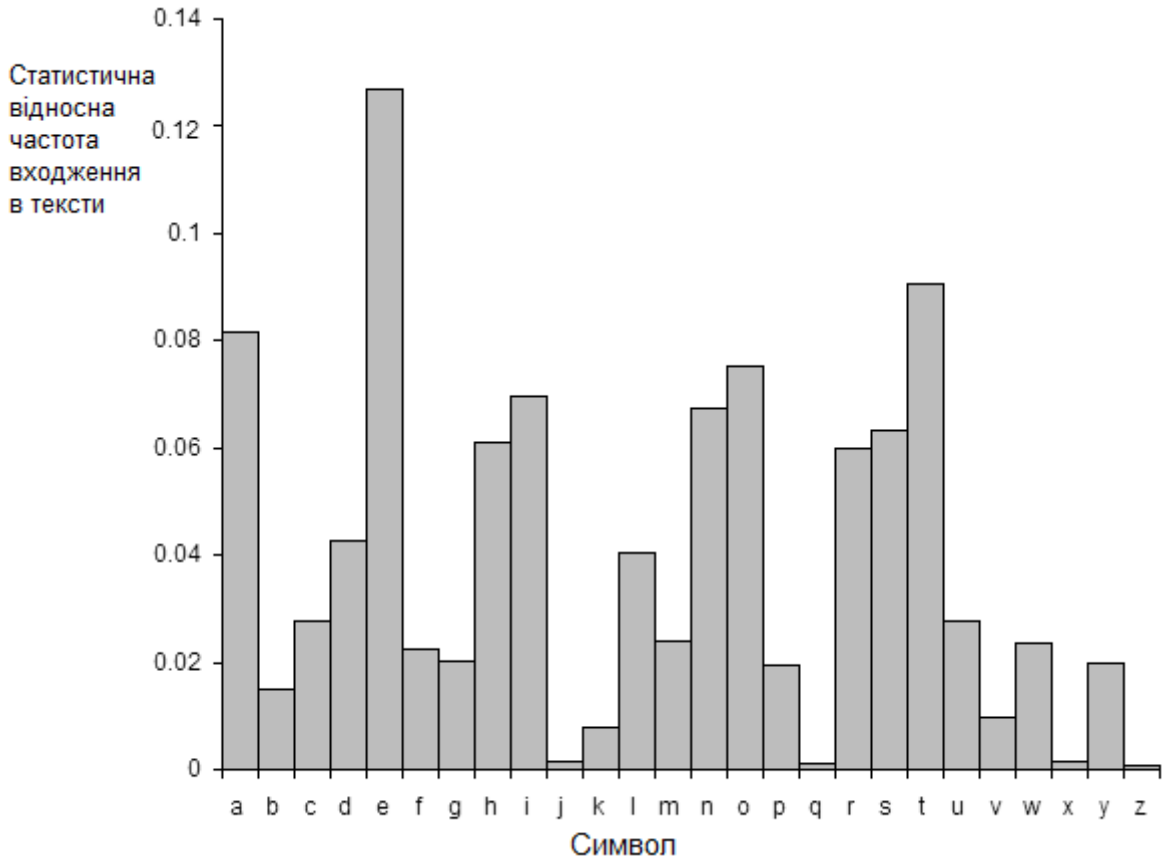


Рисунок 1.8 – Гістограму статистики входження символів латинського алфавіту в тексти англійської мови

Таким чином, статистичні залежності – основна вада шифру Цезаря, що робить його надзвичайно вразливим. З цього слідує, що запорукою

стійкості подібних шифрів є порушення статистичних залежностей частоти входження символів алфавіту в тексти певної мови.

Перший варіант порушення статистичних залежностей частоти входження символів алфавіту в тексти - зміна величини зсуву (шифр Віженера).

Щоб знати кратність кожного чергового зсуву, спосіб зсувів визначається наперед. Для цього використовується ключове слово, кожен символ якого задає величину зсуву своїм номером в алфавіті. Ключове слово повторюється циклічно, поки не буде завершено шифрування всіх букв відкритого тексту.

Шифр Віженера утворюється за таблицею Віженера [1, 6]. Таблиця Віженера є матрицею з $n \times n$ елементами, де n – кількість символів алфавіту. Кожен рядок матриці одержаний циклічним зсувом алфавіту на один символ відносно попереднього.

Для шифрування вибирається символьний ключ, відповідно до якого формується робоча матриця шифрування.

Для отримання робочої матриці з повної таблиці вибирається перший рядок і ті рядки, останні букви яких відповідають буквам ключа. Першим розміщується перший рядок, а під ним – рядки, відповідні буквам ключа в порядку проходження цих букв в ключі.

Процес шифрування здійснюється за правилами:

– під кожною буквою відкритого тексту записують букви ключа;

– кожна буква тексту замінюється за робочою матрицею буквою, що знаходиться на перетині стовпчика, в заголовку (першому рядку) якого знаходиться обрана для заміни буква, та рядка, що закінчується буквою ключа, яку зіставлено замінюваній букві тексту.

Процес дешифрування здійснюється за подібними правилами:

– під кожною буквою шифрованого тексту записують букви ключа (ключ при цьому повторюється необхідну кількість раз);

– з матриці для кожного символу шифрованого тексту обирається рядок, що закінчується буквою ключа, яку зіставлено обраному символу, в

рядку відшукується цей символ (обирається стовпчик) і з відповідного стовпчика береться буква, що знаходиться в першому рядку і використовується для заміни.

Методи підстановки (заміни) відносяться до найпростіших видів шифрування, що зумовлює їх малу стійкість проти розкриття ключа.

Шифри Цезаря і Віженера побудовані на моноалфавітній підстановці - букви тексту шифрованого замінюються іншими буквами того ж самого або іншого алфавіту.

Для підвищення стійкості шифрів використовують поліалфавітні підстановки, в яких для заміни символів початкового тексту використовуються символи декількох алфавітів (вони можуть бути отримані з одного застосуванням різних варіантів його перетворення).

При поліалфавітній одноконтурній звичайній підстановці для заміни символів початкового тексту використовується декілька алфавітів, зміна алфавітів здійснюється послідовно і циклічно (перший символ замінюється відповідним символом першого алфавіту, другий – символом другого тощо.) Для поліалфавітних підстановок статистичні характеристики початкового тексту майже не виявляються в зашифрованому тексті.

Загальна модель шифрування підстановкою може бути представлена в наступному вигляді:

$$t = z + w \mod (k-1), \tag{1.5}$$

де t – символ зашифрованого тексту; z – символ початкового тексту; w – ціле число в діапазоні від 0 до (k-1); k – кількість символів алфавіту.

Якщо значення w фіксоване, то формула описує моноалфавітну підстановку. Якщо w вибирається з послідовності $w_1, w_2, \ldots, w_n$ то виходить поліалфавітна підстановка з періодом n.

Якщо в поліалфавітній підстановці $n>m$ (де m – кількість знаків шифрованого тексту) і будь-яка послідовність $w_1, w_2, \ldots, w_n$ використовується тільки один раз, то такий шифр, теоретично, не допускає

розкриття без доступу до початкового тексту. Цей шифр називають шифром Вермана.

В сучасних умовах алгоритмічно реалізовувані шифри зсуву не можуть слугувати надійними засобами захисту інформаційних ресурсів комп'ютерних систем. З іншої сторони, шифри зсуву часто використовуються для дослідження способів підвищення криптостійкості алгоритмів шифрування. При цьому базовою стає гіпотеза, що метод, здатний підвищити криптостійкість елементарного шифрування Цезаря, апріорно може розглядатися як кандидат в методи підвищення криптостійкості більш складних алгоритмів шифрування.

Саме на останній гіпотезі базується ідеологія розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

1.3 Оптимальне кодування Хаффмена в задачах криптографічного захисту

Перед тим, як безпосередньо перейти до розгляду перспектив застосування оптимального кодування Хаффмена в задачах криптографічного захисту при розробці методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, проведемо узагальнюючий аналіз характерних особливостей цього кодування за результатами дослідження методів ефективного кодування.

Це дозволяє зробити ряд висновків:

– оптимальне кодування Хаффмена є різновидом ефективного кодування;

– оптимальне кодування Хаффмена відноситься до класу ентропійних кодів;

– коди Хаффмена є нерівномірними кодами;

– коди Хаффмена є префіксними кодами;

– головною відмінністю коду Хаффмена від коду Шеннона-Фано є те, що він завжди дає оптимальний варіант ентропійного кодування за наявними статистичними даними.

– коди Хаффмена, як і інші методи ефективного кодування, безпосередньо не призначені для вирішення задач шифрування і захисту даних;

– стиснення даних оптимальним кодуванням Хаффмена дає позитивний ефект для захисту інформаційних ресурсів комп'ютерних систем через зменшення розмірів повідомлень, що передаються (менша імовірність втрат інформації при передачі);

– відхилення варіанту кодування Хаффмена від примітивного кодування символів також ускладнює дешифрування тексту.

Для визначення перспектив застосування оптимального кодування Хаффмена в задачах криптографічного захисту дослідимо принципи цього кодування.

Побудова оптимального нерівномірного коду за методикою Хаффмена виконується за загальним алгоритмом, що включає наступні етапи:

– всі символи, що кодуються, упорядковуються в порядку зменшення ймовірностей;

– останні два символи впорядкованої множини (вони повинні мати найменші значення ймовірностей) замінюються допоміжним символом, значення ймовірності для якого визначається сумарною ймовірністю елементів, що замінюються;

– всі елементи нової множини знову упорядковуються на зменшення ймовірностей;

– виконання попередніх двох операції повторюється до отримання єдиного допоміжного символу.

Для визначення кодових комбінацій символів виконується зворотний аналіз виконаних об'єднань.

Тобто, двом останнім символам, при об’єднанні яких було отримано символ з ймовірністю 1, присвоюються значення коду 0 і 1. Після цього розглядаються символи попереднього рівня, які прийняли участь в утворенні останніх допоміжних символів. Аналогічним чином їм ставляться у відповідність значення 0 та 1, які дописуються в молодший розряд кодових комбінацій.

Завершення кодування Хаффмена відбувається після досягнення етапу, на якому кодові комбінації будуть співставлені у відповідність всім символам вихідного алфавіту.

Формування кодових комбінацій оптимального нерівномірного коду Хаффмена (другий етап) також може виконуватися на підставі кодового дерева.

Побудова кодового дерева та визначення комбінацій оптимального нерівномірного коду з його допомогою виконується за наступним алгоритмом:

– коренева вершина дерева ставиться у відповідність останньому отриманому допоміжному символу (символу, ймовірність якого дорівнює 1);

– гілки кодового дерева формуються в послідовності, зворотній проходженню процесу об’єднання символів. З кореневої вершини дерева відводяться дві гілки, кінцевим вершинам яких ставляться у відповідність символи, об’єднання яких дозволило отримати символ з сумарною ймовірністю 1. При цьому кінцевій вершині лівої гілки ставиться у відповідність символ, що знаходиться в таблиці у вищому рядку (як правило, це символ з більшою ймовірністю), а кінцевій вершині правої гілки – символ, розташований у таблиці в нижньому рядку;

– процес нарощування гілок продовжується до отримання дерева, кінцевим вершинам якого відповідають символи вихідного алфавіту;

– після закінчення побудови кодового дерева його гілкам ставляться у відповідність символи коду: гілкам, які відгалужуються з вершин вліво, –

символ 0; гілкам, які відгалужуються з вершин вправо, – символ 1 (можливе кодування навпаки);

– кодові комбінації ОНК, що відповідають символам первинного (вихідного) алфавіту, визначаються послідовністю значень, які ставилися у відповідність гілкам дерева від кореневої вершини і до відповідної кінцевої.

Отриманий за наведеним алгоритмом методикою нерівномірний код буде оптимальним кодом Хаффмена.

Для визначення перспектив застосування оптимального нерівномірного коду Хаффмена для збільшення криптостійкості алгоритмів шифрування розглянемо приклад практичного застосування методу кодування Хаффмена.

Розглянемо приклад застосування методу Хаффмена для довільного набору з 100 символів:

ADBCBADCSDCASCZDSMMMCCMMMMAWSDSASWZDSMWUSCCMMMMEEE

WWBWBUDWSDWUSCCCCMMMMCWSDSASCZDSMMMMMEWWZDSMMZDSMM

Потужність застосованого в наданому для кодування тексті алфавіту дорівнює 10, оскільки в ньому використано 10 символів: A, B, C, D, E, M, S, U, W, Z.

За формулою (1.2) визначимо рекомендовану розрядність двійкових кодів для заданого алфавіту при використанні примітивних (рівномірних) кодів: $n = \log_2 10 = 3.322$.

Відповідно, для результату 3.322 отримуємо мінімально-достатню розрядність примітивного коду 4.

Для аналізу використаємо простий варіант послідовного кодування символів двійкового алфавіту двійковими числами (табл. 1.3).

Таблиця 1.3 – Примітивний код для кодування вихідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Код	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Із застосуванням наведеного в таблиці 1.3 примітивного рівномірного коду початковий текст перетворюється в масив двійкових кодів загальною розрядністю 400 біт:

0000 0011 0001 0010 0001 0000 0011 0010 0110 0011
0010 0000 0110 0010 1001 0011 0110 0101 0101 0101
0010 0010 0101 0101 0101 0101 0000 1000 0110 0011
0110 0000 0110 1000 1001 0011 0110 0101 1000 0111
0110 0010 0010 0101 0101 0101 0101 0100 0100 0100
1000 1000 0001 1000 0001 0111 0011 1000 0110 0011
1000 0111 0110 0010 0010 0010 0010 0101 0101 0101
0101 0010 1000 0110 0011 0110 0000 0110 0010 1001
0011 0110 0101 0101 0101 0101 0101 0100 1000 1000
1001 0011 0110 0101 0101 1001 0011 0110 0101 0101

Дослідження статистичних властивостей вихідного тексту дозволяє визначити ймовірності появи в ньому символів алфавіту (табл. 1.4).

Наявність статистичних даних щодо імовірностей входження символів алфавіту до тексту, що подається на кодування, дозволяє застосувати алгоритм кодування Хаффмена.

Хід поєднань символів на основі статистичних даних за алгоритмом кодування Хаффмена наведено в таблиці 1.5.

За даними таблиці 1.5 будуємо кодове дерево Хаффмена (рис. 1.9) і формуємо кодові комбінації оптимального нерівномірного коду Хаффмена для кодування вихідного тексту (табл. 1.6).

Таблиця 1.4 – Статистичні характеристики вихідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Кількість повторів символу	6	4	14	12	4	25	16	3	11	5
Статистична імовірність	0.06	0.04	0.14	0.12	0.04	0.25	0.16	0.03	0.11	0.05

Таблиця 1.5 – Реалізація алгоритму Хаффмена

Символ	Імовірність	Символ	Імовірність	Символ	Імовірність	Символ	Імовірність	Символ	Імовірність	Символ	Імовірність
M	0.25	M	0.25	M	0.25	M	0.25	M	0.25	M	0.25
S	0.16	S	0.16	S	0.16	S	0.16	WZB	0.20	EUAD	0.25
C	0.14	C	0.14	C	0.14	C	0.14	S	0.16	WZB	0.20
D	0.12	D	0.12	D	0.12	EUA	0.13	C	0.14	S	0.16
W	0.11	W	0.11	W	0.11	D	0.12	EUA	0.13	C	0.14
A	0.06	EU	0.07	ZB	0.09	W	0.11	D	0.12		
Z	0.05	A	0.06	EU	0.07	ZB	0.09				
B	0.04	Z	0.05	A	0.06						
E	0.04	B	0.04								
U	0.03										

Таблиця 1.5 завершення

Символ	Імовірність	Символ	Імовірність	Символ	Імовірність	Символ	Імовірність
SC	0.30	EUADWZB	0.45	SCM	0.55	SCMEUADWZB	1.00
M	0.25	SC	0.30	EUADWZB	0.45		
EUAD	0.25	M	0.25				
WZB	0.20						

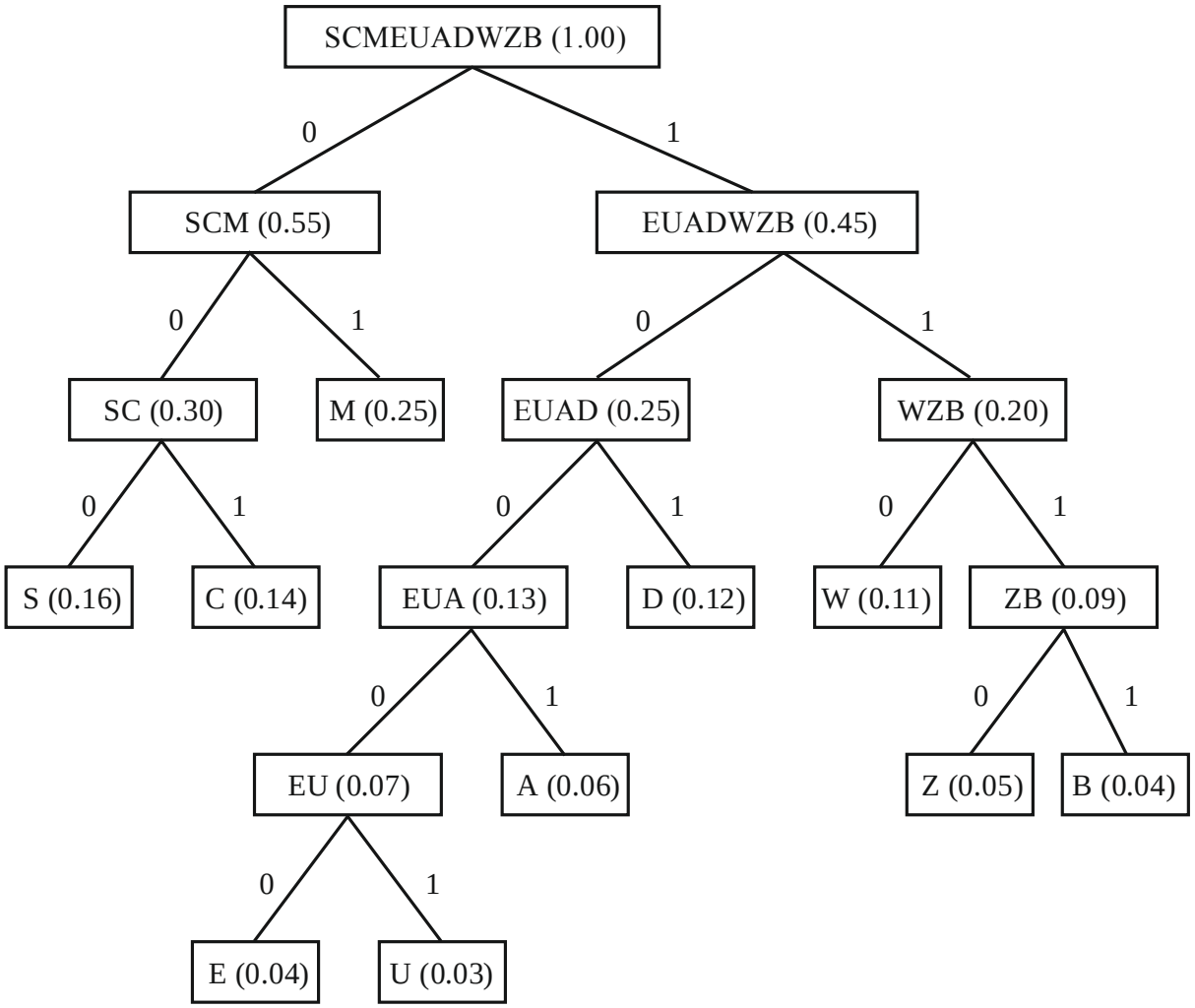


Рисунок 1.9 – Кодове дерево Хаффмена

Таблиця 1.6 – Оптимальний код Хаффмена для кодування вихідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Код Хаффмена	1001	1111	001	101	10000	01	000	10001	110	1110

Із застосуванням наведеного в таблиці 1.6 оптимального нерівномірного коду Хаффмена початковий текст перетворюється в масив двійкових кодів загальною розрядністю 304 біт:

1001 101 1111 001 1111 1001 101 001 000 101

001 1001 000 001 1110 101 000 01 01 01

001 001 01 01 01 01 1001 110 000 101

000 1001 000 110 1110 101 000 01 110 10001

000 001 001 01 01 01 01 10000 10000 10000

110 110 1111 110 1111 10001 101 110 000 101

110 10001 000 001 001 001 001 01 01 01

01 001 110 000 101 000 1001 000 001 1110

101 000 01 01 01 01 01 10000 110 110

1110 101 000 01 01 1110 101 000 01 01

Першочергово відзначимо позитивний ефект стиску даних із застосуванням оптимального нерівномірного кодування Хаффмена – замість 400 біт текст в оптимальному кодуванні займає лише 304 біти.

За формулою (1.3) визначимо коефіцієнт стиснення коду:

$$k = \frac{304}{400} * 100\% = 76\%.$$

Отриманий коефіцієнт є досить високим показником для несистематизованого вихідного тексту і дозволяє стверджувати, що ризик ушкодження оптимального коду тексту порівняно з початковим примітивним зменшується майже на чверть (24% пропорційно зменшенню розрядності коду). Це підтверджує ефективність використання оптимального кодування Хаффмена для кодування наданого тексту.

Для визначення перспектив застосування оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем з використанням шифрів зсуву перетворимо отриманий нерівномірний код початкового тексту в рівномірні кодові комбінації, потрібні нам для реалізації шифру зсуву.

Початково формуємо потокове (нерозривне послідовне) представлення тексту кодом Хаффмена. Для наочності і зручності подальшого аналізу коди Хаффмена окремих символів з непарними номерами позицій в тексті виділено жирним шрифтом (застосовано виділення жирним символів через один для можливості наочного розрізнення кодів символів в бітовій послідовності):

100**1**10**1**11**1**10**0**11**1**11**1**10**0**11**0**10**1**00**1**00**0**10**1**00**1**10**0**10**0**00**0**11**1**10**1**01
000**0**10**1**01**0**01**0**01**0**10**1**01**1**10**0**01**1**10**0**00**0**10**1**00**0**10**0**11**0**11**1**01
010**0**00**1**11**0**10**0**01**0**00**0**01**0**01**0**10**1**01**0**11**0**00**0**01**0**00**0**01**0**00**0**11**0**11**0**1
111**1**10**1**11**1**11**0**00**1**10**1**11**1**00**0**01**0**11**1**01**0**00**0**10**0**00**0**01**0**01**0**01**0**10**1**
010**1**00**1**11**0**00**0**01**0**10**0**01**0**01**0**00**0**01**1**11**0**10**1**00**0**01**0**10**1**01**0**11**0**00**0**0
110**1**10**1**11**0**10**1**00**0**01**0**11**1**10**1**01**0**00**0**01**0**1.

Розіб’ємо отримане потокове представлення тексту кодом Хаффмена на чотирьохрозрядні комбінації (тетради) у відповідності із розмірністю початкового варіанту примітивного кодування:

1001 1011 1110 0111 1110 0110 1001 0001 0100 1100
1000 0011 1101 0100 0010 1010 0100 1010 1010 1100
1110 0001 0100 0100 1000 1101 1101 0100 0011 1010
0010 0000 1001 0101 0101 1000 0100 0010 0001 1011
0111 1110 1111 1000 1101 1100 0010 1110 1000 1000
0010 0100 1001 0101 0101 0011 1000 0101 0001 0010
0000 1111 0101 0000 1010 1010 1100 0011 0110 1110
1010 0001 0111 1010 1000 0101

З прикладу можна побачити, що в отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради кодових комбінаціях наявні всі 16 можливих кодових комбінацій довжиною

4 біти, а не 10, як це було в початковому примітивному коді.

Невідповідність кількості кодових комбінацій застосовуваного коду кількості символів кодованого алфавіту та руйнування статистичних залежностей між символами є передумовою підвищення криптостійкості методів шифрування загалом і формування шифрів зсуву для підвищення ефективності захисту комп’ютерних систем зокрема.

1.4 Постановка задачі

За результатами проведеного аналізу і зроблених висновків можливо виконати постановку задачі щодо розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Головна мета дослідження – підвищення ефективності захисту інформаційних ресурсів комп’ютерних систем за рахунок зміни статистичних властивостей криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Для досягнення поставленої мети необхідно виконати ряд подальших робіт, які визначать задачі досліджень:

а) розробити математичну модель для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем;

б) визначити основні положення методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем та розробити алгоритми його реалізації;

в) обґрунтувати ефективність запропонованого методу і алгоритмів його реалізації моделюванням.

1.5 Висновки

В першому розділі кваліфікаційної роботи проведено дослідження використовуваних в комп’ютерних системах способів і методів кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів при розробці методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

На підставі проведених аналітичних досліджень і виявлених закономірностей було визначено наступне:

– всі різновиди кодування певною мірою пов’язані з вирішенням завдань захисту інформації, що робить доцільним сумісне використання різних кодів для підвищення ефективності систем захисту;

– шифри заміни зі зсувом є одним із найпростіших за реалізацією способів шифрування;

– недоліком шифрів зсуву є збереження статистичних характеристик появи символів первинного алфавіту (вихідного тексту) у вторинному алфавіті (зашифрованому тексті), що зумовлює їх низьку криптостійкість;

– методи оптимального кодування забезпечують заміну рівномірних кодів первинного алфавіту на нерівномірні коди вторинного алфавіту, за рахунок чого досягається стиснення даних. При цьому статистичні характеристики первинного алфавіту зберігаються у вторинному алфавіті;

– для зберігання і передачі інформації повідомлення з вторинного алфавіту оптимального коду розбиваються на рівномірні коди (третинний алфавіт), при чому статистичні залежності появи символів між символами первинного і третинного алфавіту втрачаються.

Це дозволило зробити наступні висновки.

В сучасних умовах алгоритмічно реалізовувані шифри зсуву не можуть слугувати надійними засобами захисту інформаційних ресурсів комп’ютерних систем. З іншої сторони, шифри зсуву часто

використовуються для дослідження способів підвищення криптостійкості алгоритмів шифрування. При цьому базовою стає гіпотеза, що метод, здатний підвищити криптостійкість елементарного шифрування Цезаря, апріорно може розглядатися як кандидат в методи підвищення криптостійкості більш складних алгоритмів шифрування. На цій гіпотезі базується ідеологія розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Оптимальне нерівномірне кодування Хаффмена забезпечує руйнування статистичних залежностей між кодовими комбінаціями, що співставляються символам вихідного тексту при примітивному кодуванні, а також здатне змінювати кількість використовуваних комбінацій рівномірного коду і характер їх формування. Невідповідність кількості кодових комбінацій застосовуваного коду кількості символів кодованого алфавіту та руйнування статистичних залежностей і ентропійних взаємозв’язків між символами є передумовою підвищення криптостійкості методів шифрування загалом і формування шифрів зсуву для підвищення ефективності захисту комп’ютерних систем зокрема.

Таким чином, в першому розділі доведено актуальність розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Отримані в першому розділі результати дозволили виконати постановку задачі дослідження.

2 МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ ФОРМУВАННЯ ШИФРІВ ЗСУВУ ІЗ ЗАСТОСУВАННЯМ ОПТИМАЛЬНОГО КОДУВАННЯ ХАФФМЕНА

2.1 Формування математичної моделі

Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем базується на використанні двох видів кодування як засобу підвищення ефективності захисту інформаційних ресурсів комп’ютерних систем:

– ефективного кодування (оптимального нерівномірного кодування Хаффмена);

– криптографічного кодування або шифрування (використання шифрів зсуву).

Відповідно, створювана математична модель методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена повинна мати інструменти адекватного відображення принципів реалізації зазначених методів кодування.

Проведемо аналіз послідовності виконання операцій, використовуваних в них даних та отримуваних результатів при реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем з визначенням необхідних для відображення цих операцій, даних та результатів складових математичної моделі.

Першочергово на систему формування шифрів зсуву із застосуванням оптимального кодування Хаффмена подається відкритий вхідний текст, що підлягає захисту.

Позначимо відкритий вхідний текст як T_1 .

Відкритий вхідний текст як T_1 складається з символів алфавіту, який в задачах кодування розглядається як первинний алфавіт вхідного тексту.

Первинний алфавіт вхідного тексту методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем утворює собою множину символів, представлених примітивними рівномірними кодами. Відповідно, для відображення первинного алфавіту вхідного тексту методу введемо до математичної моделі методу множину символів первинного алфавіту вхідного тексту:

$S_1 : \{s_{1.1}, s_{1.2}, \ldots, s_{1.i}, \ldots, s_{1.k}\}$ – множина кодів символів первинного алфавіту вхідного тексту T_1.

На основі алфавіту S_1 первинний текст T_1 можна представити як упорядковану за характером тексту множину символів алфавіту $s_{1.i} \in S_1$ з можливими повторами зазначених символів:

$T_1 : \{s_{1.k}, s_{1.f}, \ldots, s_{1.i}, \ldots, s_{1.k}, \ldots, s_{1.f}, \ldots, s_{1.f}, s_{1.j}\}$ – представлення первинного тексту як упорядкованої за характером тексту множини символів алфавіту $s_{1.i} \in S_1$ з можливими повторами зазначених символів (можливий варіант).

Як правило, для формування оптимального нерівномірного коду Хаффмена з мінімальною надлишковістю, склад символів алфавіту і їх статистичні характеристики визначаються на підставі аналізу самого вхідного тексту.

Таким чином, як було також визначено при дослідженні характерних особливостей оптимального нерівномірного кодування Хаффмена в першому розділі даної роботи, основою для реалізації цього методу кодування є відкритий вхідний текст як T_1, на основі якого формуються два описи статистичних характеристик:

– алфавіт S_1 символів $s_{1.i}$ відкритого вхідного тексту T_1;

– статистичні характеристики частоти входження символів $s_{1.i}$ алфавіту S_1 в текст T_1.

Оскільки символи певного формованого з тексту алфавіту S_x можуть повторюватися в аналізованому тексті T_x по декілька разів, для формування цільового алфавіту S_3, відобразимо в математичній моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем функцію алгоритмічного усунення дублювання кодів символів F_{RD}, що формує алфавіт використаних в тексті T_x символів S_x на підставі обробки самого тексту T_x:

$$S_x = F_{\text{RD}}\,(T_x). \qquad (2.1)$$

Для накопичення даних статистичних характеристик частоти входження символів $s_{1.i}$ алфавіту S_1 в текст T_1 введемо до складу математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем множину P_1:

P_1 : $\{p(s_{1.1}),\ p(s_{1.2}),\ \ldots,\ p(s_{1.i}),\ \ldots,\ p(s_{1.k})\}$ – статистичні частоти входження символів $s_{1.i}$ первинного алфавіту S_1 в текст T_1 (ймовірності появи символів алфавіту $s_{1.i} \in S_1$ в тексті T_1).

Для відображення в математичній моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних $_{1.i}$) систем введемо до її складу операцію визначення статистичних характеристик ймовірності $p(s_{x.i})$ появи символів алфавіту $s_{x.i} \in S_x$ в тексті T_x:

$$P_x = F_{\text{P}}\,(S_x,\, T_x) \qquad (2.2)$$

Дані характеристик первинного вхідного тексту T_1, відображених множинами кодів символів первинного алфавіту S_1 і ймовірностей появи символів алфавіту $s_{1.i} \in S_1$ в цьому тексті P_1 є достатніми для реалізації оптимального нерівномірного кодування Хаффмена.

Оптимальне нерівномірне кодування Хаффмена базується на застосуванні алгоритмічних процедур кодування символів первинного алфавіту S_1 на основі даних ймовірностей $s_{1.i} \in S_1$ появи символів $p(s_{1.i}) \in P_1$ в тексті T_1. Результатом процедури оптимального нерівномірного кодування Хаффмена стає отримання вторинного алфавіту для кодування первинного вхідного тексту T_1 , що дозволяє отримати вторинне представлення $T_{\text{ОНК}}$ вхідного тексту з аналогічною кількістю елементів (тобто, потужності множин T_1 і $T_{\text{ОНК}}$ залишаються однаковими: $|T_1|=|T_{\text{ОНК}}|$), але з властивостями нерівномірності коду елементів множини, що дозволяє досягти оптимізації коду вхідного тексту за рахунок зменшення його загальної довжини в бітах (стиску бітової послідовності).

Для відображення вторинного алфавіту і можливості опису з його застосуванням тексту $T_{\text{ОНК}}$ введемо до математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем множину символів вторинного алфавіту вхідного тексту:

$S_2 : \{s_{2.1}, s_{1.2}, \ldots, s_{2.i}, \ldots, s_{2.k}\}$ – множина кодів символів вторинного алфавіту (алфавіту оптимального коду).

Особливістю оптимального нерівномірного кодування Хаффмена є те, що для визначення оптимального нерівномірного коду $s_{2.i} \in S_2$ для будь-якого символу первинного алфавіту $s_{1.i} \in S_1$ це кодування потребує наявності статистичних даних (як базису оцінки ентропії) первинного алфавіту S_1 в цілому, а не окремих його символів. Тобто, операцію визначення складу алфавіту оптимального нерівномірного коду Хаффмена $S_{\text{НС}}$ (де НС – скорочення від Huffman Coding) для символу первинного алфавіту S_x можна

в узагальненому вигляді записати алгоритмічною функцією кодування Хаффмена, що враховує властивості символів $s_{x.i} \in S_x$ у взаємозв'язку статистичних даних множини P_x всіх елементів множини S_x:

$$S_{\text{НС}} = F_{\text{НС}}(S_x, P_x) \tag{2.3}$$

На основі алфавіту S_2 первинний текст T_1 перетворюється в оптимізований вторинний текст $T_{\text{НС}}$.

За аналогією з попереднім описом, оптимізований вторинний текст $T_{\text{НС}}$ можна представити як упорядковану за характером тексту множину символів алфавіту $s_{2.i} \in S_2$ з можливими повторами зазначених символів:

$T_{\text{НС}}$: $\{s_{2.\text{k}}, s_{2.\text{f}}, \ldots, s_{2.i}, \ldots, s_{2.\text{k}}, \ldots, s_{2.\text{f}}, \ldots, s_{2.\text{f}}, s_{2.j}\}$ – представлення вторинного тексту $T_{\text{НС}}$ як упорядкованої за характером тексту множини символів алфавіту $s_{2.i} \in S_2$ з можливими повторами зазначених символів (можливий варіант).

Операцію заміни символу $s_{1.i} \in S_1$ в тексті T_1 на оптимальний код $s_{2.i} \in S_2$, що дозволить отримати оптимальний код всього тексту $T_{\text{НС}}$ опишемо наступним чином:

$$\forall (s_{1.j} \in T_1, s_{1.i} \in S_1, s_{2.j} \in T_{\text{НС}}) \exists\, s_{1.i} = s_{1.j} : s_{2.j} \in T_{\text{НС}} = s_{1.i} \in S_1. \tag{2.4}$$

Фактично, це можна описати як узагальнену операцію заміни в закодованому тексті T_x, кодів символів одного алфавіту S_x, що використовувався для кодувався тексту T_x, на символи іншого алфавіту S_y. Тобто, мова йде про реалізацію перекодування тексту T_x із заміною кодів символів одного алфавіту S_x на коди символів одного алфавіту S_y, що дає нам нове представлення тексту T_y. Операцію перекодування позначимо як алгоритмічно реалізовувану функцію $F_{\text{ТС}}$ (де ТС – скорочення від TransCoding):

$$T_y = F_{\text{ТС}}\,(T_x, S_x, S_y) \qquad (2.5)$$

Наступним кроком кодових перетворень при реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем постає переведення оптимального нерівномірного коду тексту $T_{\text{НС}}$ в рівномірні кодові комбінації для застосування шифрів зсуву.

Переведення оптимального нерівномірного коду тексту $T_{\text{НС}}$ в рівномірні кодові комбінації передбачає виконання двох операцій з бітовими кодами:

– перетворення оптимального нерівномірного коду Хаффмена тексту $T_{\text{НС}}$ в нерозривне послідовне повідомлення (потокове представлення коду Хаффмена тексту $T_{\text{НС}}$);

– дроблення потокового представлення тексту $T_{\text{НС}}$ на рівномірні кодові комбінації у відповідності до потреб алгоритму (засобів) реалізації шифрів зсуву.

Визначимо спосіб математичного представлення зазначених операцій в математичній моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

Функцію послідовного поєднання кодів $s_{j.i} \in S_j$ і $s_{j.f} \in S_j$ ідентифікуємо як F_{SC} (де SC – Sequential Combination)

$$F_{\text{SC}}\,(s_{j.i}\,,\, s_{j.f}) = s_{j.i} \cup s_{j.f} \qquad (2.6)$$

Застосування функції (2.6) для всіх елементів $s_{2.i} \in T_2$ дає можливість виконати перетворення оптимального нерівномірного коду Хаффмена тексту $T_{\text{НС}}$ в нерозривне послідовне повідомлення $T'_{\text{НС}}$ (потокове представлення коду Хаффмена тексту $T_{\text{НС}}$).

$$T'_{НС} = F_{SC}(T_{НС}). \quad (2.7)$$

Пропорційне розбиття (дроблення) потокового представлення тексту $T'_{НС}$ на рівномірні кодові комбінації у відповідності до потреб алгоритму (засобів) реалізації шифрів зсуву представимо як реалізацію функції F_{PD} (де PD – скорочення від Proportional Division) розбиття бітової послідовності B (від Binary) на рівномірні кодові комбінації розрядності r з утворенням розподіленого рівномірного коду T_{UC} (де UC – скорочення від Uniform Code):

$$T_{UC} = F_{PD}(B, r). \quad (2.8)$$

Відзначимо оборотність функцій (2.8) і (2.7) стосовно двійкового коду B:

$$F_{SC}(F_{PD}(B, r)) = B. \quad (2.9)$$

Результатом застосування функції (2.8) до послідовного двійкового коду тексту $T'_{НС}$ є нове представлення $T''_{НС}$ тексту $T'_{НС}$ як упорядкованої за порядком входження в текст $T'_{НС}$ множини символів нового шуканого алфавіту з порушеними статистичними властивостями і оновленим складом набору символів.

Оскільки символи нового шуканого алфавіту (алфавіту з порушеними статистичними властивостями і оновленим складом символів) можуть повторюватися по декілька разів в тексті $T''_{НС}$, для формування цільового алфавіту S_3 застосуємо функцію алгоритмічного усунення дублювання кодів символів (2.1).

Для систематизованого опису третинного алфавіту S_3 і виконання операцій з його елементами введемо до математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування

Хаффмена для підвищення ефективності захисту комп'ютерних систем множину символів третинного алфавіту вхідного тексту:

$S_3 : \{s_{3.1}, s_{3.2}, \ldots, s_{3.i}, \ldots, s_{3.n}\}$ – множина рівномірних кодів третинного алфавіту оптимального коду вхідного тексту.

За аналогією з попередніми описами, оптимізований текст $T''_{НС}$ можна представити як упорядковану за порядком входження в текст $T'_{НС}$ множини символів нового шуканого алфавіту тексту множину символів алфавіту $s_{3.i} \in S_3$ з можливими повторами зазначених символів:

$T''_{НС} : \{s_{3.h}, s_{3.u}, \ldots, s_{3.j}, \ldots, s_{3.l}, \ldots, s_{3.f}, \ldots, s_{3.j}, \ldots, s_{3.x}\}$ – представлення тексту $T''_{НС}$ як упорядкованої множини символів алфавіту $s_{3.j} \in S_3$ з можливими повторами зазначених символів (можливий варіант).

Одразу відзначимо можливість зворотного перетворення тексту $T''_{НС}$ з нового рівномірного алфавіту в нерозривне послідовне повідомлення $T'_{НС}$ із застосуванням функцій (2.7) та властивості (2.9), що є необхідним при зворотному перетворенні зашифрованого тексту в текст T_1:

$$T'_{НС} = F_{SC}(T''_{НС}). \qquad (2.10)$$

Третинний алфавіт створено спеціально для забезпечення можливості застосування шифрів зсуву щодо оптимального коду $T_{НС}$ вхідного тексту T_1.

Таким чином, на даному етапі відбувається перехід від оптимального кодування Хаффмена до застосування шифрів зсуву.

Для застосування шифрів зсуву при реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем уточимо формулу загальної модель шифрування підстановкою зі зсувом (1.5) з урахуванням вже визначених властивостей математичної моделі методу:

$$F_{Ew}(s_{j.i}) = e_i = s_{j.(i+w) \bmod |Sj|}; \quad (2.11)$$

де $s_{j.i}$ – символ тексту, що входить в шифрування (традиційно, символ початкового тексту); $F_{Ew}(s_{j.i})$ – оператор шифрування символів тексту $s_{j.i}$ заміною зі зсувом на розмір ключа шифрування; e_i – символ зашифрованого тексту (замінник символу тексту $s_{j.i}$); w – ключ шифрування (ціле число, яке визначає кратність зсуву алфавітів, $0 \le w < \text{mod} | S_j |$); mod $| S_j |$ – кількість символів або потужність алфавіту S_j.

Для систематизованого опису алфавіту шифрування і виконання операцій з його елементами введемо до математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем множину символів алфавіту шифрування оптимізованого тексту:

$E:\{e_1, e_2,\ldots, e_i, \ldots, e_m\}$ – множина кодів символів алфавіту шифрування.

Множини E і S_3 складаються з однакових елементів, але з циклічно зміщеним на w позицій порядком слідування кодів символів відповідно до формули (2.11).

З формули (2.11) і загальних принципів реалізації шифрів заміни слідує однозначна відповідність між елементами алфавітів S_3 і E, яка дозволяє сформувати функцію зворотного перетворення – дешифрування зашифрованого тексту T_E:

$$F_{Dw}(e_i) = s_{j.i} = e_{(i-w) \bmod |Sj|}. \quad (2.12)$$

За формулою (2.12) з зашифрованого тексту T_E відбувається відтворення оптимізованого тексту $T''_{НС}$, який є базовим для виконання зворотних перетворень і відтворення відкритого вхідного тексту як T_1 (із

застосуванням класичного варіанту реалізації алгоритму декодування оптимального коду Хаффмена).

Таким чином, на підставі проведеного аналітичного синтезу можна сформувати узагальнене представлення математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена:

$$M=<T_1, T_{HC}, T'_{HC}, T''_{HC}, T_{Ew}, P_1, S_1, S_2, S_3, E, F_{RD}, F_P, F_{HC}, F_{TC}, F_{SC}, F_{PD}, F_E, F_D>$$

2.2 Апробація математичної моделі

Перед застосуванням запропонованої математичної моделі для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем потрібно провести апробацію адекватності математичної моделі потребам методу.

Апробацію адекватності математичної моделі потребам методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем проведемо на основі даних прикладу, розглянутого в першому розділі при дослідженні можливостей застосування оптимального кодування Хаффмена в задачах криптографічного захисту.

На підставі аналізу використаного в прикладі вхідного тексту з 100 символів було визначено, що потужність застосованого в наданому для кодування тексті алфавіту дорівнює 10, оскільки в ньому використано 10 символів: A, B, C, D, E, M, S, U, W, Z.

Це дає нам узагальнене представлення множини S_1 кодів символів первинного алфавіту вхідного тексту T_1 у вигляді множини з 10 елементів: $S_1 : \{s_{1.1}, s_{1.2}, s_{1.3}, s_{1.4}, s_{1.5}, s_{1.6}, s_{1.7}, s_{1.8}, s_{1.9}, s_{1.10}\}$.

Формування множини кодів алфавіту S_1 на підставі наявного вхідного тексту у вигляді множини рівномірних двійкових кодів T_1 можна відобразити як застосування функції алгоритмічного усунення дублювання кодів символів (2.1) до коду вихідного тексту T_1:

$$S_1 = F_{RD}\,(T_1). \qquad (2.13)$$

Результатом операції (2.13) є узагальнене представлення множини кодів символів первинного алфавіту вхідного тексту T_1, відповідного даним таблиці 1.3: S_1 : {0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001}.

Оскільки вхідний текст у вигляді множини рівномірних двійкових кодів T_1 зі 100 елементів занадто громіздкий для наочного відображення апробації моделі, обмежимося демонстрацією властивостей математичної моделі для перших 10 елементів множини T_1 з узагальненим відображенням інших елементів:T_1 : {0000, 0011, 0011, 0001, 0010, 0001, 0000, 0011, 0010, 0110, 0011, .., 0101}.

Дані складу множин T_1 і S_1 є базовими для застосування функції визначення статистичних характеристик ймовірності $p(s_{1.i})$ появи символів алфавіту $s_{1.i} \in S_1$ в тексті T_1 (2.2), що дає нам можливість визначити статистичні частоти входження символів $s_{1.i}$ первинного алфавіту S_1 в текст T_1 (ймовірності появи символів алфавіту $s_{1.i} \in S_1$ в тексті T_1): P_1 : {0.06, 0.04, 0.14, 0.12, 0.04, 0.25, 0.16, 0.03, 0.11, 0.05}.

Наявні на даному етапі дані множин S_1 і P_1 є достатніми для застосування методу оптимального кодування Хаффмена з метою отримання оптимальних нерівномірних кодів $s_{2.i}$ символів $s_{1.i} \in S_1$ і утворення з них множини кодів символів вторинного алфавіту S_2 (алфавіту оптимального коду). Операція визначення оптимальних нерівномірних кодів $s_{2.i}$ символів $s_{1.i} \in S_1$ в математичній моделі визначена функцією (2.3), яку можна записати для множини символів $s_{1.i} \in S_1$ узагальнено:

$$\forall\, s_{1.i} \in S_1 : s_{2.i} = F_{\text{HC}}\,(s_{1.i}, P_1). \qquad (2.14)$$

Результатом застосування функції (2.14) до S_1 і P_1 є множина символів вторинного алфавіту: S_2 : {1001, 1111, 001, 101, 10000, 01, 000, 10001, 110, 1110}.

На підставі формул (2.4)-(2.5) реалізується операція заміни символів $s_{1.i} \in S_1$ в тексті T_1 на оптимальний код $s_{2.i} \in S_2$, що дозволяє отримати оптимальний код всього тексту: $T_{\text{НС}}$: {1001, 101, 1111, 001, 1111, 1001, 101, 001, 000, 101, ..., 01}.

Застосування функцій (2.6)-(2.7) для всіх елементів $s_{2.i} \in T_{\text{НС}}$ дає можливість виконати перетворення оптимального нерівномірного коду Хаффмена тексту $T_{\text{НС}}$ в нерозривне послідовне повідомлення $T'_{\text{НС}}$ (потокове представлення коду Хаффмена тексту $T_{\text{НС}}$): $T'_{\text{НС}}$: {1001101111100111111001101001000101...01}.

Розбиття потокового представлення тексту кодом Хаффмена $T'_{\text{НС}}$ на чотирьохрозрядні комбінації (тетради) у відповідності із розмірністю початкового варіанту примітивного кодування можна описати на основі (2.8) функцією:

$$T''_{\text{НС}} = F_{\text{PD}}(T'_{\text{НС}}, 4). \qquad (2.15)$$

Це дає нам оновлене представлення початкового тексту, закодованого оптимальним нерівномірним кодом, у вигляді чотирьохрозрядних комбінацій рівномірного коду: $T''_{\text{НС}}$: {1001, 1011, 1110, 0111, 1110, 0110, 1001, 0001, 0100, ..., 0101}.

Формування множини кодів нового рівномірного алфавіту S_3 на підставі наявного оптимізованого тексту у вигляді множини рівномірних двійкових кодів $T''_{\text{НС}}$ можна відобразити як застосування функції алгоритмічного усунення дублювання кодів символів (2.1) до коду вихідного тексту T_1:

$$S_3 = F_{RD}\,(T''_{\text{НС}}). \qquad (2.16)$$

Результатом операції (2.16) є узагальнене представлення множини кодів символів третинного алфавіту тексту $T'_{НС}$: S_3 : {0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111}.

Дані складу множин $T''_{НС}$ і S_3 є базовими для застосування функції визначення статистичних характеристик ймовірності $p(s_{3.i})$ появи символів алфавіту $s_{3.i} \in S_3$ в тексті $T''_{НС}$ (2.2), що дає нам можливість визначити статистичні частоти входження символів $s_{3.i}$ первинного алфавіту S_3 в текст $T''_{НС}$ (ймовірності появи символів алфавіту $s_{3.i} \in S_3$ в тексті $T''_{НС}$), наведені в таблиці 1.7: P_3 : {3.947368, 6.578947, 7.894737, 5.263158, 10.52632, 9.210526, 2.631579, 3.947368, 10.52632, 5.263158, 10.52632, 2.631579, 5.263158, 5.263158, 7.894737, 2.631579}.

Дані множини P_3 не є необхідними для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, але можуть бути необхідними для обгрунтування його ефективності.

Множина кодів символів третинного алфавіту S_3 є основою для формування шифру заміни зі зсувом на основі оператора шифрування (2.11).

Для прикладу приймемо w=5, що дасть на основі оператора шифрування (2.11) зсунутий алфавіт таблиці шифрування: E : {1011, 1100, 1101, 1110, 1111, 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010}.

Безпосередньо операцію шифрування тексту зі зсувом $T''_{НС}$ із застосуванням зміщеного алфавіту заміни E на основі оператора (2.5) можна описати функцією:

$$\forall (s_{3.j} \in T''_{НС}, e_i \in E, e_j \in T_E) \exists\, e_i = s_{1.j} : e_j \in T_E = e_i \in E \qquad (2.17)$$

Результатом застосування функції заміни (2.17) відносно $T''_{НС}$ на основі алфавіту заміни E дасть зашифрований шифрами зсуву текст T_E: T_E : {0100**,** 0110**,** 1001, 0010**,** 1001, 0001**,** 0100, 1100, 1111, ..., 0000}.

Реалізація функції заміни (2.17) відносно $T''_{НС}$ на основі алфавіту заміни E дозволяє отримати зашифрований шифрами зсуву текст T_E із статистичними властивостями, які не відповідають статистичним властивостям вхідного тексту T_1 і вхідного алфавіту S_1.

2.3 Висновки

В даному розділі описано математичну модель, розроблену для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

В ході розробки моделі досліджено і визначено:

- елементи опису початкового і перехідних варіантів представлення способів кодування вхідного тексту;
- елементи опису використовуваних на різних етапах алфавітів кодування;
- функції формування і заміни алфавітів кодування;
- функції перетворення форм представлення кодованого тексту;
- властивості математичної моделі.

Також доведено відповідність і достатність інструментарію математичної моделі задачам методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

Наявний інструментарій математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем дозволяє перейти до визначення основних положень методу і засобів його реалізації.

3 МЕТОД ФОРМУВАННЯ ШИФРІВ ЗСУВУ ІЗ ЗАСТОСУВАННЯМ ОПТИМАЛЬНОГО КОДУВАННЯ ХАФФМЕНА ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЗАХИСТУ КОМП'ЮТЕРНИХ СИСТЕМ

3.1 Визначення базових положень методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем

Згідно з сформульованими в постановці задачі дослідження положеннями, наступним етапом після розробки математичної моделі для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем постає визначення основних положень методу та розробка алгоритмів його реалізації.

Для визначення основних положень методу необхідно провести узагальнення результатів раніше зроблених аналітичних досліджень, висновків, зроблених при розробці і апробації математичної моделі, та задач, що переслідуються при розробці методу.

Першочергово, метод орієнтований на підвищення ефективності захисту інформаційних ресурсів комп'ютерних систем за рахунок зміни статистичних властивостей криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Як було констатовано на етапі розробки математичної моделі методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, створюваний метод базується на використанні двох видів кодування як засобу підвищення ефективності захисту інформаційних ресурсів комп'ютерних систем:

– оптимального нерівномірного кодування Хаффмена;
– шифрування заміною на основі зсунутого алфавіту.

Шифрування зі зсувом алфавіту, або, як частіше кажуть, шифрування зсувом відрізняється низькою криптостійкістю, тому не може бути на сьогодні ефективним способом криптографічного захисту. Шифри зсуву в методі формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем застосовуються як елементарний варіант криптографічного захисту даних з мінімальною криптостійкістю для дослідження можливостей підвищення ефективності алгоритмів криптографічного шифрування застосуванням методів оптимального кодування. З цієї точки зору шифрування і дешифрування при реалізації методі формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем доцільно виконувати за найпростішим класичним алгоритмом реалізації шифру зсуву (шифр Цезаря).

Формування оптимального нерівномірного коду вхідного тексту реалізується за традиційним алгоритмом Хаффмена на основі статистичних властивостей цього тексту.

Позитивний ефект від застосування двох традиційних алгоритмів кодування (шифрів зсуву і оптимального нерівномірного кодування Хаффмена) досягається за рахунок нетрадиційного використання результатів кодування Хаффмена як засобу зміни статистичних властивостей криптографічних шифрів зсуву.

Зміна статистичних властивостей криптографічних шифрів зсуву в пропонованому методі досягається застосуванням оптимального кодування Хаффмена для утворення нового способу рівномірного кодування оптимізованого тексту. При цьому склад нового алфавіту кодування і статистичні властивості частот входження його символів до тексту докорінно відрізняються від початкового алфавіту рівномірного кодування вихідного тексту, що і є підставою для підвищення криптостійкості шифрів зсуву при реалізації методу формування шифрів зсуву із застосуванням

оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Позитивний ефект стосовно застосування шифрів зсуву і загального підвищення ефективності систем захисту від запропонованого способу застосування оптимального нерівномірного кодування Хаффмена для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем є багатоаспектним.

По перше, досягається зменшення розміру бінарного коду вхідного тексту, що підлягає шифруванню, через застосування оптимального кодування як такого.

По друге, змінюється потужність (кількість елементів) алфавіту шифрування.

По третє, змінюється спосіб рівномірного кодування символів алфавіту шифрування.

По четверте, руйнується залежність між символами початкового тексту і комбінаціями кодів рівномірного кодування символів алфавіту шифрування.

І найважливіше п’яте – усувається основна вразливість шифрів зсуву, а саме змінюються статистичні характеристики частоти входження символів алфавіту шифрування в закодований текст порівняно з відповідними характеристиками початкового тексту.

Зазначені позитивні ефекти від оптимального нерівномірного кодування Хаффмена досягаються завдяки тому, що оптимальний нерівномірний код вхідного тексту представляється послідовним кодом, до якого застосовується операція розбиття на кодові комбінації рівної довжини (фіксованої розрядності), які і є вхідними в алгоритм шифрування.

У підсумку наведеного аналізу сформулюємо базові положення методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем, на які будемо спиратися в подальшій його алгоритмічні реалізації:

1. Метод орієнтований на підвищення ефективності захисту інформаційних ресурсів комп'ютерних систем за рахунок зміни статистичних властивостей криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

2. Метод базується на послідовному застосування до вихідного тексту повідомлень оптимального нерівномірного кодування Хаффмена і шифру зсуву.

3. Шифри зсуву застосовуються як елементарний варіант криптографічного захисту даних з мінімальною криптостійкістю для дослідження можливостей підвищення ефективності алгоритмів криптографічного шифрування застосуванням методів оптимального кодування. Шифрування і дешифрування виконується за найпростішим класичним алгоритмом реалізації шифру зсуву (шифр Цезаря).

4. Оптимальний нерівномірний код Хаффмена для вхідного тексту утворюється за класичним алгоритмом, представляється послідовним кодом, до якого застосовується операція розбиття на кодові комбінації рівної довжини (фіксованої розрядності), які є вхідними в алгоритм шифрування.

5. Застосування оптимального нерівномірного кодування Хаффмена з переходом до рівномірних кодів забезпечує:

- зменшення розміру бінарного коду вхідного тексту, що підлягає шифруванню (традиційний результат);
- зміну потужності (кількість елементів) алфавіту шифрування;
- змінюваність способу кодування алфавіту шифрування для різних текстів;
- руйнування відповідності між символами початкового тексту і комбінаціями кодів алфавіту шифрування;
- зміну статистичних характеристики частоти входження символів алфавіту шифрування в закодований текст порівняно з відповідними характеристиками початкового тексту.

3.2 Алгоритмічна реалізація методу

В загальному алгоритмі реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем можна виділити наступні основні етапи:

– формування множини кодів алфавіту S_1 на підставі наявного вхідного тексту у вигляді множини рівномірних двійкових кодів T_1;

– визначення статистичних характеристик ймовірності появи символів алфавіту $s_{1.i} \in S_1$ в тексті T_1;

– застосування методу оптимального нерівномірного кодування Хаффмена з метою отримання оптимальних нерівномірних кодів $s_{2.i}$ символів $s_{1.i} \in S_1$ і утворення з них множини кодів символів вторинного алфавіту S_2 (алфавіту оптимального коду);

– заміна символів $s_{1.i} \in S_1$ в тексті T_1 на оптимальні коди $s_{2.i} \in S_2$, що дає оптимальний код тексту;

– перетворення оптимального нерівномірного коду тексту $T_{\text{НС}}$ в нерозривне послідовне повідомлення $T'_{\text{НС}}$ (потокове представлення коду Хаффмена тексту $T_{\text{НС}}$);

– розбиття потокового представлення тексту кодом Хаффмена $T'_{\text{НС}}$ на рівномірні кодові комбінації заданої розрядності w;

– формування множини кодів нового рівномірного алфавіту S_3 на підставі наявного оптимізованого тексту у вигляді множини рівномірних двійкових кодів;

– формування шифру заміни зі зсувом на основі третинного алфавіту S_3 і оператора шифрування;

– шифрування тексту із застосуванням зміщеного алфавіту заміни.

Алгоритм дій протилежного перетворення (дешифрування) тексту, закритого шифром зсуву згідно методу формування шифрів зсуву із

застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, можна описати наступною послідовністю операцій:

– дешифрування тексту із застосуванням зміщеного алфавіту заміни;

– перетворення декодованого тексту в нерозривне послідовне повідомлення $T'_{НС}$ (потокове представлення коду Хаффмена тексту $T_{НС}$);

– виділення в потоковому представленні коду Хаффмена символів тексту за таблицею кодування алфавіту S_2;

– заміна кодів символів алфавіту S_2 кодами символів з алфавіту S_1.

Для реалізації алгоритму декодування таблиця відповідності між кодами ОНК і кодами первинного повідомлення вводиться до складу тексту $T_{НС}$ і зазнає перетворень та передається разом з ним.

Розглянемо алгоритми реалізації базових процедур методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

Для зменшення кількості процедур сумістимо деякі з них.

Першим розглянемо алгоритм формування множини кодів алфавіту S_1 і визначення статистичних характеристик ймовірності їх появи на підставі наявного вхідного тексту $T_{1.}$

Алгоритм 3.1. Алгоритм формування множин S_1 і P_1 на підставі обробки вхідного тексту як множини кодів символів T_1.

3.1.1 Прийняти вхідний текст і представити як множину кодів символів T_1.

3.1.2 Прийняти $S_1=\varnothing$, $P_1=\varnothing$, $j=1$, $k=1$.

3.1.3 Обрати з T_1 код символу $s_{1.j}\in T_1$.

3.1.4 Перевірити наявність в множині S_1 коду символу $s_{1.i}\in S_1$ аналогічного $s_{1.j}\in T_{1.}$

3.1.5 Якщо код $s_{1.i}\in S_1$ аналогічний $s_{1.j}\in T_1$ в множині S_1 знайдено, перейти до п. 3.1.9.

3.1.6 Додати елемент $s_{1.j}\in T_1$ в множину S_1 як $s_{1.k}\in S_1$.

3.1.7 Додати в множину P_1 елемент $p(s_{1.k}) = 1/|T_1|$.

3.1.8 Прийняти $k = k+1$.

3.1.9 Визначити значення i для знайденого елемента $s_{1.i} \in S_1$ аналогічного $s_{1.j} \in T_1$.

3.1.10 Прийняти $p(s_{1.i}) = p(s_{1.i}) + 1/|T_1|$.

3.1.11 Прийняти $j = j+1$.

3.1.12 Якщо $j \leq |T_1|$, перейти до п. 3.1.3.

3.1.13 Кінець алгоритму.

Одержані за алгоритмом 3.1 дані множин S_1 і P_1 є достатніми для застосування методу оптимального кодування Хаффмена з метою отримання оптимальних нерівномірних кодів $s_{2.i}$ символів $s_{1.i} \in S_1$ і утворення з них множини кодів символів вторинного алфавіту S_2 (алфавіту оптимального коду).

Алгоритм оптимального кодування Хаффмена в реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем використовується традиційний і не представляє новизни. В даному розділі власна реалізація алгоритму оптимального кодування Хаффмена наводиться як один із можливих варіантів без претендування на наукову новизну для повноти опису алгоритмічної реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

Алгоритм 3.2. Алгоритм ефективного кодування алфавіту S_1 (формування алфавіту S_2).

3.2.1 Утворити одномірний масив P розмірністю $|P|=|S_1|$

3.2.2 Ввести дані множини P_1 до масиву P.

3.2.3 Утворити масив Q (модель множини S_2) розмірністю $|Q|=|S_1|$ з пустими початковими значеннями елементів $q_i \in Q$.

3.2.4 Утворити допоміжну множину C розмірністю $|C|=|S_1|$ (для супроводу змін множини P_1 в масиві P).

3.2.5 Прийняти $c_i=i$ для всіх $c_i \in C$.

3.2.6 Утворити допоміжну множину D розмірністю $|D|=|S_2|$ (для супроводу змін множини S_2 в масиві Q).

3.2.7 Прийняти $d_i=i$ для всіх $d_i \in D$.

3.2.8 Обрати елементи $p_i \in P$ і $p_j \in P$ з найменшими значеннями (визначається два елементи алфавіту, що мають найменші значення ймовірностей).

3.2.9 Зафіксувати значення i і j.

3.2.10 Прийняти $x=c_i$, $y=c_i$ ($c_i \in C$).

3.2.11 Прийняти $z=1$.

3.2.12 Обрати елемент $d_z \in D$.

3.2.13 Якщо $d_z=x$, перейти до п.3.2.17.

3.2.14 Прийняти $z = z + 1$.

3.2.15 Якщо $z \le |C|$, перейти до п.3.2.12.

3.2.16 Перейти до п.3.2.19.

3.2.17 Додати в молодший розряд коду елемента $q_z \in Q$ значення 0.

3.2.18 Перейти до п.3.2.14.

3.2.19 Прийняти $z=1$.

3.2.20 Обрати елемент $d_z \in D$.

3.2.21 Якщо $d_z=y$, перейти до п.3.2.25.

3.2.22 Прийняти $z = z + 1$.

3.2.23 Якщо $z \le |C|$, перейти до п.3.2.23.

3.2.24 Перейти до п.3.2.19.

3.2.25 Додати в молодший розряд коду елемента $q_z \in Q$ значення 1.

3.2.26 Прийняти $d_z=x$.

3.2.27 Перейти до п.3.2.22.

3.2.28 Прийняти $p_i = p_i + p_j$.

3.2.29 Видалити $p_j \in P$ з масиву P.

3.2.30 Видалити $c_j \in C$ з масиву C.

3.2.31 Якщо $|C|>1$, перейти до п.3.2.8.

3.2.32 Кінець алгоритму.

Виходом алгоритму ефективного кодування алфавіту S_1 (алгоритм 3.2.) є масив значень Q, що містить оптимальні нерівномірні коди символів алфавіту S_1 і є аналогом алфавіту S_2.

Наступними кроками реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем є елементарні процедури за типовими алгоритмами, які можна представити як узагальнений алгоритм реалізації методу на основі наявних кодів S_2 оптимального кодування Хаффмена для символів алфавіту S_1.

Алгоритм 3.3. Узагальнений алгоритм реалізації методу на основі наявних кодів S_2 оптимального кодування Хаффмена для символів алфавіту S_1.

3.3.1 Замінити коди символів $s_{1.i} \in S_1$ в тексті T_1 на оптимальні коди $s_{2.i} \in S_2$.

3.3.2 Перетворити оптимальний варіант нерівномірного коду тексту $T_{НС}$ в нерозривне послідовне повідомлення $T'_{НС}$ (потокове представлення коду Хаффмена тексту $T_{НС}$).

3.3.3 Прийняти $S_3 = \varnothing$, $j=1$.

3.3.4 Задати розрядність w коду цільового алфавіту S_3.

3.3.5 Відокремити w старших розрядів двійкового коду від шифру $T'_{НС}$.

3.3.6 Прийняти відокремлений код як $s_{3.j} \in T''_{НС}$.

3.3.7 Прийняти $j = j+1$.

3.3.8 Якщо $|T'_{НС}| > 0$, перейти до п.3.3.5.

3.3.9 Сформувати алфавіт S_3 усуненням дублювань кодів в $T''_{НС}$.

3.3.10 Задати кратність зсуву r (оператор шифрування) для формування шифру заміни.

3.3.11 Формування шифру заміни E зі зсувом r позицій на основі третинного алфавіту S_3.

3.3.12 Шифрування тексту із застосуванням шифру заміни *E.* (зміщеного третинного алфавіту S_3).

3.3.13 Кінець алгоритму.

Процедуру усунення дублювань (пункт 3.3.9 алгоритму 3.3) при формуванні алфавіту з тексту з одночасним накопиченням статистичних даних про властивості тексту було розглянуто в алгоритмі 3.1, тому повторно її розглядати в спрощеному варіанті (без накопичення статистичних даних про властивості тексту) не будемо, оскільки вилучити статистичні операції з алгоритмі 3.1, за потреби, не викликає ускладнень.

Пункти 3.3.6-3.3.9 алгоритму 3.3 відповідають процедурі розбиття потокового представлення тексту кодом Хаффмена $T'_{\text{НС}}$ на рівномірні кодові комбінації заданої розрядності w з формуванням множини кодів нового рівномірного алфавіту S_3 на підставі наявного оптимізованого тексту у вигляді множини рівномірних двійкових кодів.

Пункти 3.3.10-3.3.11 алгоритму 3.3 відповідають процедурі утворення шифру заміни.

Пункт 3.3.12 алгоритму 3.3 відповідають процедурі шифрування тексту із застосуванням шифру заміни.

Всі зазначені процедури є простими, реалізуються за відомими алгоритмами і не потребують деталізації в науковій роботі.

Рисунок 3.1 – Узагальнений алгоритм реалізації

На наведеній на рис. 3.1 блок-схемі узагальненого алгоритму реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем вершини можна розділити на групи:

– вершини 1-3 відповідають введенню вхідних даних, необхідних для реалізації методу;

– вершини 4-5 відповідають процедурі формування множини кодів алфавіту S_1 і визначення статистичних характеристик ймовірності їх появи на підставі наявного вхідного тексту T_1 (цю процедуру описує алгоритм 3.1)

– вершина 6 відповідає процедурі ефективного кодування алфавіту S_1 методом Хаффмена з формуванням алфавіту S_2 (цю процедуру описує алгоритм 3.2);

– вершина 7 відповідає операції перекодування вхідного тексту оптимальним кодом;

– вершини 8-10 відображують особливості процедури переходу від нерівномірного способу кодування тексту кодом Хаффмена до рівномірних кодів, необхідних для реалізації шифрування зсувом (саме ця частина алгоритму забезпечує зміну статистичних властивостей тексту для реалізації криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена;

– вершини 11-17 відображують класичні операції реалізації шифрів зсуву.

3.3 Висновки

В третьому розділі кваліфікаційної роботи магістра отримано наступні основні результати:

– надано опис основних положень методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем;

– запропоновано узагальнений алгоритм реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем;

– надано деталізовані алгоритми базових процедур, застосованих в узагальненому алгоритмі реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Таким чином, в даному розділі повністю вирішена чергова складова із задач дослідження: визначити основні положення методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем та розробити алгоритми його реалізації.

4 АПРОБАЦІЯ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО МЕТОДУ І АЛГОРИТМІВ ЙОГО РЕАЛІЗАЦІЇ

4.1 Апробація роботи методу при шифруванні експериментального тексту з набору символів латинського алфавіту

Для підтвердження дієвості методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем проведемо поглиблений аналіз отримуваних з його допомогою результатів стосовно прикладу кодів Хаффмена для довільного набору з 100 символів латинських, дані якого наведено в таблицях 1.3-1.6.

Першочергово наведемо базовий варіант примітивного кодування тексту T1 (у вигляді масиву двійкових кодів загальною розрядністю 400 біт), необхідний для порівняльного аналізу результатів застосування методу:

0000 0011 0001 0010 0001 0000 0011 0010 0110 0011
0010 0000 0110 0010 1001 0011 0110 0101 0101 0101
0010 0010 0101 0101 0101 0101 0000 1000 0110 0011
0110 0000 0110 1000 1001 0011 0110 0101 1000 0111
0110 0010 0010 0101 0101 0101 0101 0100 0100 0100
1000 1000 0001 1000 0001 0111 0011 1000 0110 0011
1000 0111 0110 0010 0010 0010 0010 0101 0101 0101
0101 0010 1000 0110 0011 0110 0000 0110 0010 1001
0011 0110 0101 0101 0101 0101 0101 0100 1000 1000
1001 0011 0110 0101 0101 1001 0011 0110 0101 0101

Як зазначалося при апробації математичної моделі, на основі зазначених даних узагальнене представлення множини кодів символів первинного алфавіту вхідного тексту T_1 відповідного даним таблиці 1.3: S_1 : {0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001}. Дані складу множин T_1 і S_1 є базовими для застосування функції визначення статистичних характеристик ймовірності $p(s_{1.i})$ появи символів алфавіту $s_{1.i} \in S_1$ в тексті T_1: P_1 : {0.06, 0.04, 0.14, 0.12, 0.04, 0.25, 0.16, 0.03, 0.11, 0.05}.

В результаті застосування алгоритму кодування Хаффмена (алгоритм 3.2) до S_1 і P_1 отримана множина символів вторинного алфавіту: S_2 : {1001, 1111, 001, 101, 10000, 01, 000, 10001, 110, 1110}. Це дозволяє отримати оптимальний код всього тексту $T_{НС}$ (масив двійкових кодів загальною розрядністю 304 біт). Для наочності і зручності подальшого аналізу коди Хаффмена окремих символів з непарними номерами позицій в тексті виділено жирним шрифтом (застосовано виділення жирним символів через один для можливості наочного розрізнення кодів символів в бітовій послідовності):

1001 101 **1111** 001 **1111** 1001 **101** 001 **000** 101
001 1001 **000** 001 **1110** 101 **000** 01 **01** 01
001 001 **01** 01 **01** 01 **1001** 110 **000** 101
000 1001 **000** 110 **1110** 101 **000** 01 **110** 10001
000 001 **001** 01 **01** 01 **01** 10000 **10000** 10000
110 110 **1111** 110 **1111** 10001 **101** 110 **000** 101
110 10001 **000** 001 **001** 001 **001** 01 **01** 01
01 001 **110** 000 **101** 000 **1001** 000 **001** 1110
101 000 **01** 01 **01** 01 **01** 10000 **110** 110
1110 101 **000** 01 **01** 1110 **101** 000 **01** 01

З тексту $T_{НС}$ початково сформовано потокове (нерозривне послідовне) представлення тексту кодом Хаффмена $T'_{НС}$.

1001101**1111**001**1111**1001**101**001**101**001**1001**000**001**1110**101**000**01**01**0
01001**01**101**01**101**1001**110**000**101**000**1001**000**110**1110**101**000**01**110**1000**1**000**00
1**001**01**01**01**01**10000**10000**10000**110**110**1111**110**1111**1001**101**110**000**101**110**
10001**000**001**001**1001**001**01**01**01**01**001**110**000**101**000**100**1000**001**1110**101**000**
101**01**01**01**10000**110**110**1110**101**000**01**01**1110**101**0000**01**01.

Розбиття потокового представлення тексту кодом Хаффмена $T'_{НС}$ на чотирьохрозрядні комбінації (тетради) у відповідності із розмірністю початкового варіанту примітивного кодування дало варіант тексту, представлений в математичній моделі як $T''_{НС}$:

1001 1011 **111**0 01**11** **11**10 01**10** **1**001 **000**1 01**00** **1**100

1**000** 0011 **110**1 01**00 0**01**0 1**01**0 0**1**00 1**01**0 1**01**0 1**100**

1110 **000**1 01**00 0**100 1**000** 110**1 110**1 01**00 0**011 **10**10

001**0 00**00 1**001** 01**01** 01**01** 1000 0**100 00**10 000**1 10**11

0**111 1**110 **1111** 1000 1**101** 110**0 00**10 1**110** 1000 1**000**

001**0 0**100 1**001** 01**01** 01**01** 0011 **10**00 0**101** 000**1 001**0

00**00 1**111 0**101** 000**0 1**01**0 1**01**0 1**100 00**11 0**110 **1110**

101**0 00**01 **01**11 10**10 1**000 **0**101

Аналіз отриманого розбиття дозволяє побачити певні особливості утворення рівномірного коду з нерівномірного:

– лише невелика кількість отриманих кодових комбінацій отриманого рівномірного коду відповідає кодовим комбінаціям символів у реалізації оптимального нерівномірного коду Хаффмена (однократно зустрічаються коди 1001, 1110 і 1111);

– визначені в попередньому аналізі кодові комбінації оптимального нерівномірного коду Хаффмена 1001, 1110 і 1111 в рівномірному коді наявні не лише як коди символів A, B і Z відповідно – аналогічні кодові комбінації утворюються з фрагментів кодів інших символів;

– в більшості випадків кодові комбінації оптимального нерівномірного коду Хаффмена при переході від нерівномірного кодування до рівномірного дробленням бітової послідовності зазнають дроблення на частини між декількома кодовими комбінаціями рівномірного коду (кожна з комбінацій 101**1, 111**0, 01**11**, **11**10**,** 01**10**, як і більшість інших, утворені ділять між собою фрагменти двох кодових комбінацій оптимального нерівномірного коду Хаффмена);

– наслідком попередньої властивості є те, що більшість кодових комбінацій рівномірного коду, отриманого дробленням бітової послідовності оптимального нерівномірного коду Хаффмена, містять фрагменти декількох кодових комбінацій коду Хаффмена (кожна з комбінацій 101**1, 111**0, 01**11, 11**10, 01**10,** як і більшість інших, утворені з фрагментів двох кодових комбінацій оптимального нерівномірного коду Хаффмена, а комбінації **0**011 і **0**011**,** як приклад, утворені з фрагментів трьох

кодових комбінацій);

– серед кодових комбінацій рівномірного коду, отриманого дробленням бітової послідовності оптимального нерівномірного коду Хаффмена, зустрічаються кодові комбінації, які є фрагментами кодових комбінацій коду Хаффмена більшої розрядності (кодова комбінація 1000 в одному місці отримана урізанням з комбінації 10000, а в іншому - з комбінації 10001). При цьому можливе виокремлення частини розрядів як зі сторони молодших розрядів, так і зі сторони старших або з середньої частини довгої кодової комбінації коду Хаффмена

Більш детальний аналіз рівномірних кодів, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради, дозволило визначити збільшення кількості наявних кодових комбінацій порівняно з початковим примітивним кодуванням.

В наведеному нижче представленні підкреслено різні види кодових комбінацій, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради:

1001 1011 **111**0 0**11**1 **11**10 0**11**0 1001 **000**1 0**10**0 **1**100

1000 0011 **110**1 0**10**0 **0**01**0** **1**01**0** **01**00 1**01**0 1**01**0 1**100**

1110 **000**1 0**10**0 **0**100 1**000** 11**01** **11**01 0**10**0 **0**01**1** **10**10

001**0** **00**00 1**001** 0**101** 0**101** 1000 0**100** **00**10 000**1** **10**11

0**111** **1**110 **1111** 1000 1**101** 110**0** **00**10 1**110** 1000 1**000**

001**0** **01**00 1**001** 0**101** 0**101** 001**1** **10**00 0**101** 000**1** **001**0

00**00** **1**111 0**101** 000**0** **1**01**0** **1**01**0** **1**100 00**11** **0**110 **1110**

101**0** **00**01 **01**11 10**10** **1**000 **01**01

З прикладу чітко видно, що в отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради кодових комбінаціях наявні всі 16 можливих кодових комбінацій довжиною 4 біти, а не 10, як це було в початковому примітивному коді.

В таблиці 4.1 наведено статистичні дані щодо частоти появи різних кодових комбінацій рівномірного коду, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради.

Таблиця 4.1 – Статистичні характеристики рівномірного коду S_3

№ з/п	Код	Кількість повторів коду	Статистична імовірність
1.	0000	3	0,039474
2.	0001	5	0,065789
3.	0010	6	0,078947
4.	0011	4	0,052632
5.	0100	8	0,105263
6.	0101	7	0,092105
7.	0110	2	0,026316
8.	0111	3	0,039474
9.	1000	8	0,105263
10.	1001	4	0,052632
11.	1010	8	0,105263
12.	1011	2	0,026316
13.	1100	4	0,052632
14.	1101	4	0,052632
15.	1110	6	0,078947
16.	1111	2	0,026316

Порівняння статистичних даних таблиць 1.4 і 4.7 свідчить про руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями, що співставляються символам вихідного тексту при примітивному кодуванні, а також про здатність відповідної процедури змінювати кількість використовуваних комбінацій рівномірного коду і характер їх формування.

Для наочності за даними таблиць 1.4 і 4.7 побудовано гістограми імовірностей появи символів обох кодів.

Гістограма імовірності появи символів початкового алфавіту зображена на рис. 4.1.

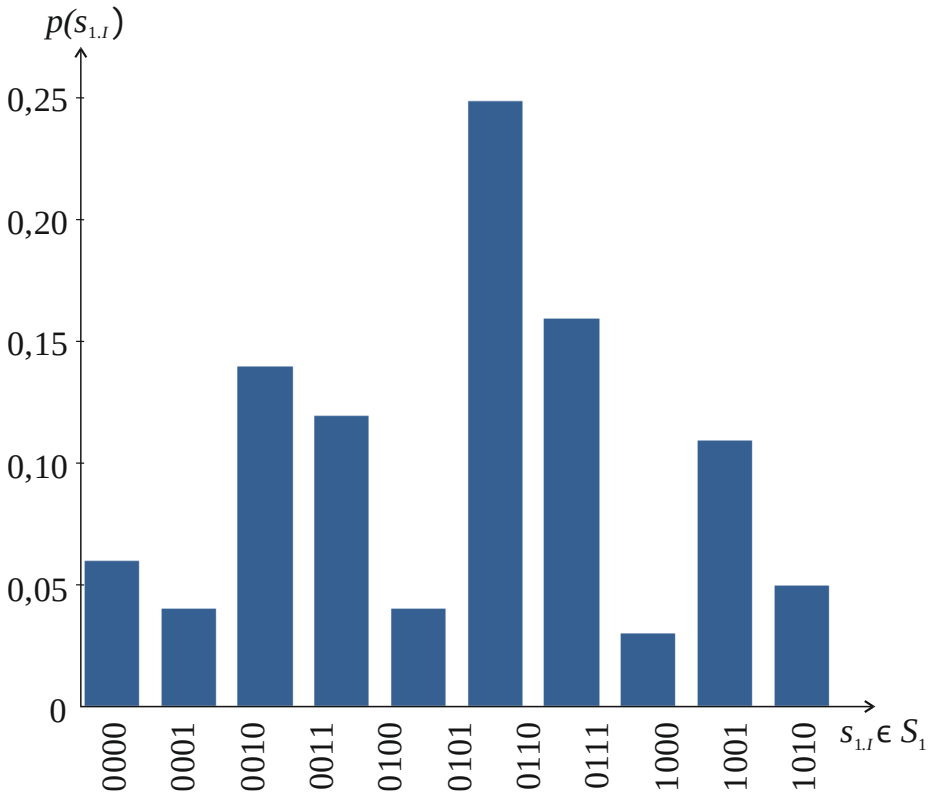


Рисунок 4.1 – Гістограма імовірності появи символів початкового алфавіту S_1

Гістограма імовірності появи символів алфавіту шифрування зображена на рис. 4.2.

Порівняння статистичних гістограм також наочно свідчить про руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями алфавітів S_1 і S_3.

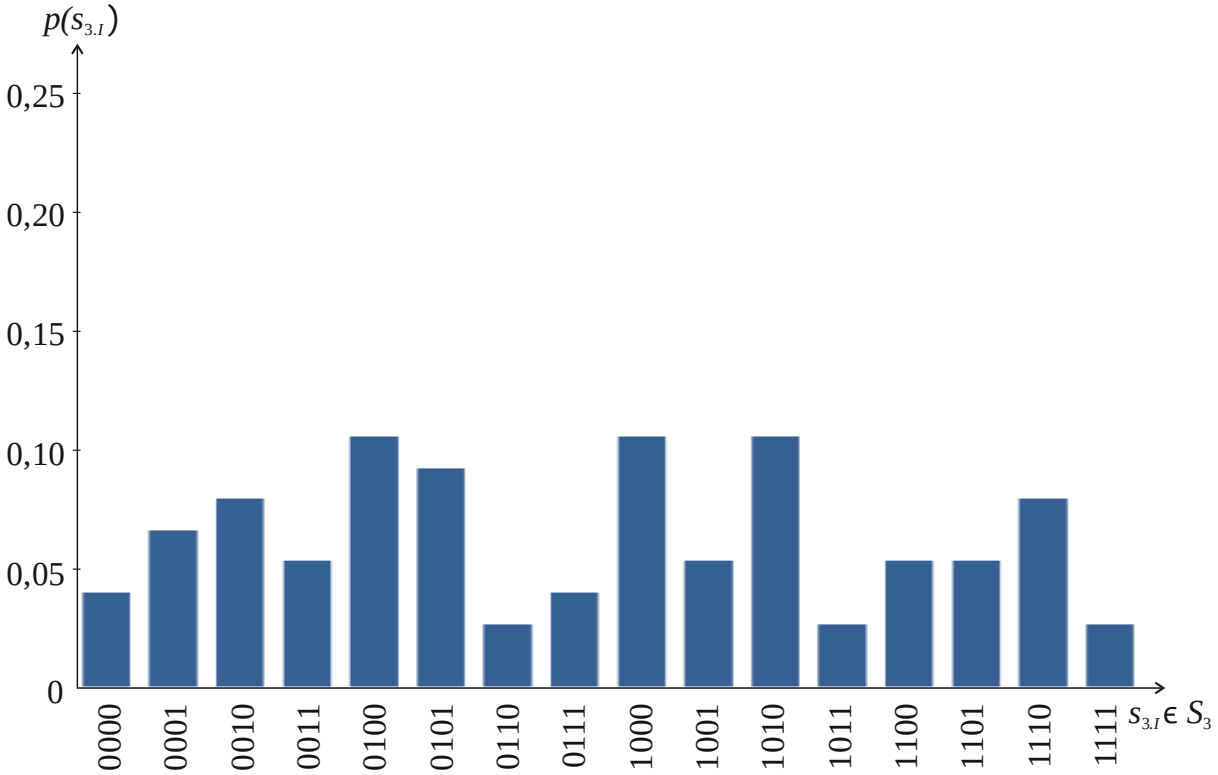


Рисунок 4.2 – Гістограма імовірності появи символів алфавіту шифрів зсуву S_3

Множина кодів символів третинного алфавіту S_3 є основою для формування шифру заміни зі зсувом на основі оператора шифрування (2.11).

Для прикладу з w=5 зсунутий алфавіт таблиці шифрування має вигляд і повністю наслідує статистичні властивості алфавіту S_3: E : {1011, 1100, 1101, 1110, 1111, 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010}.

При апробації роботи методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем моделюванні було проведене моделювання перетворень двійково-кодованих алфавітів і текстів згідно представленої на рис.2.1 граф-моделі, що дало можливість відобразити роботу методу за граф-моделлю на розглянутих двійкових даних (рис 4.3).

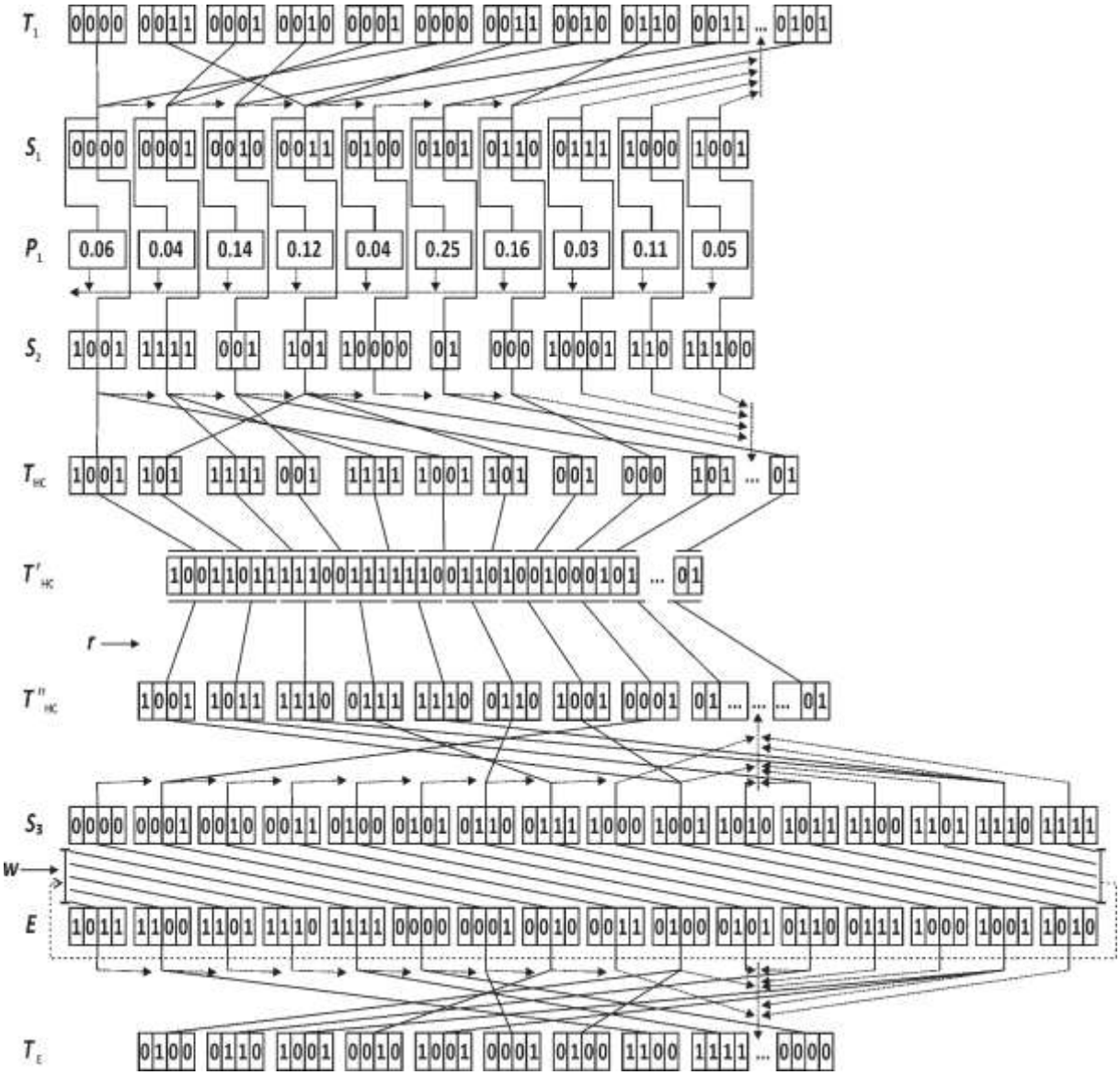


Рисунок 4.3 – Модель роботи методу на двійкових даних

4.2 Апробація роботи методу при шифруванні україномовного тексту

Для підтвердження адаптивності методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем щодо різних варіантів і мов подання текстів наступним експериментом з апробації методу було моделювання його роботи при шифруванні україномовного тексту.

Для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем обрано фрагмент тексту вступу даної роботи, в якому

описуються основні характеристики магістерського дослідження (починаючи від мети роботи і завершуючи практичною цінністю результатів). При кодуванні були дещо оптимізовано текст на зменшення кількості різновидів знаків, а також при кодуванні не враховувався регістр літер тексту.

Обраний для кодування текст наведено в додатку В.

Загальна статистична оцінка тексту наведена на рис. 4.4.

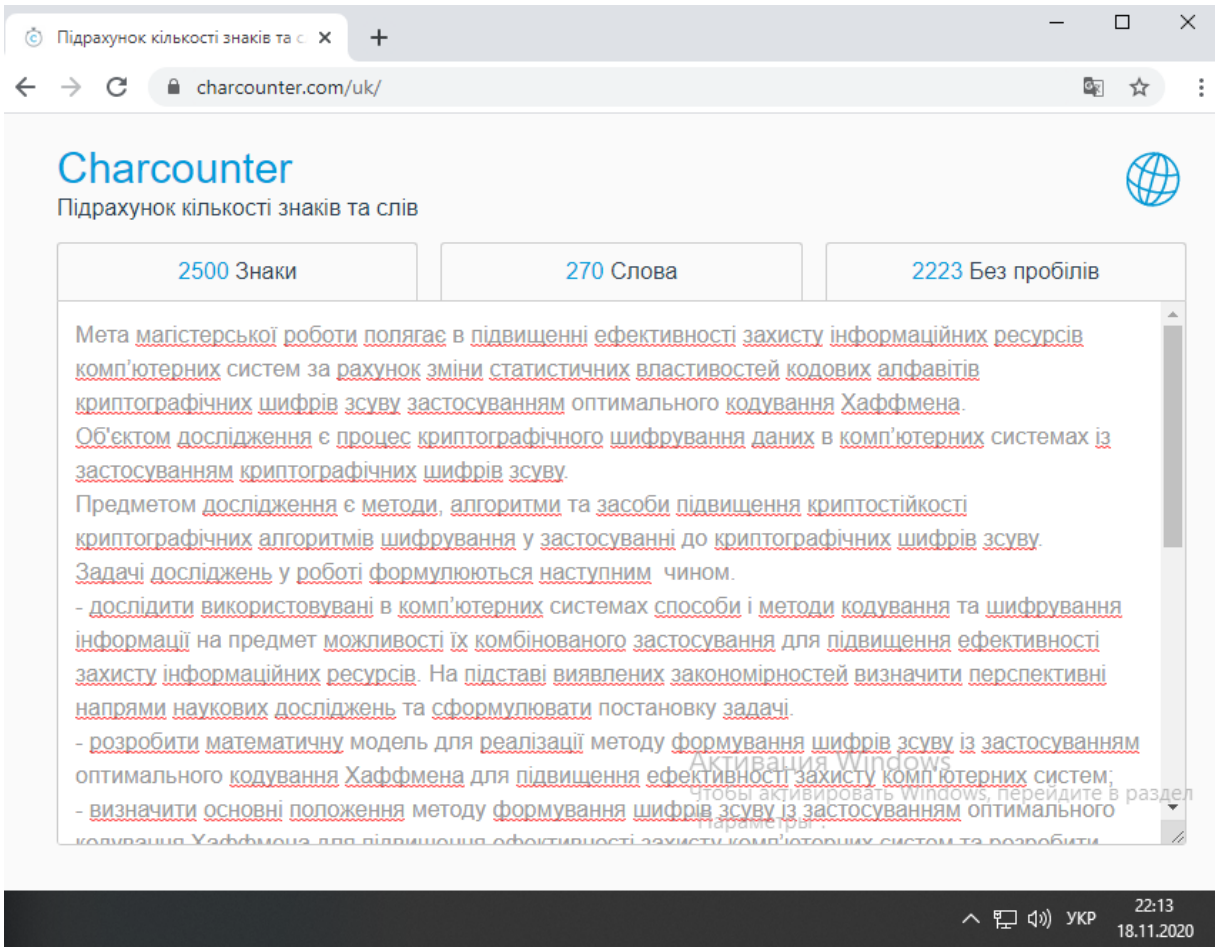


Рисунок 4.4 – Загальна статистична оцінка розміру обраного для експерименту тексту

На рис. 4.5 наведено гістограму статистичних властивостей символів $s_{1,i} \in S_1$ відповідно до результатів дослідження складу тексту.

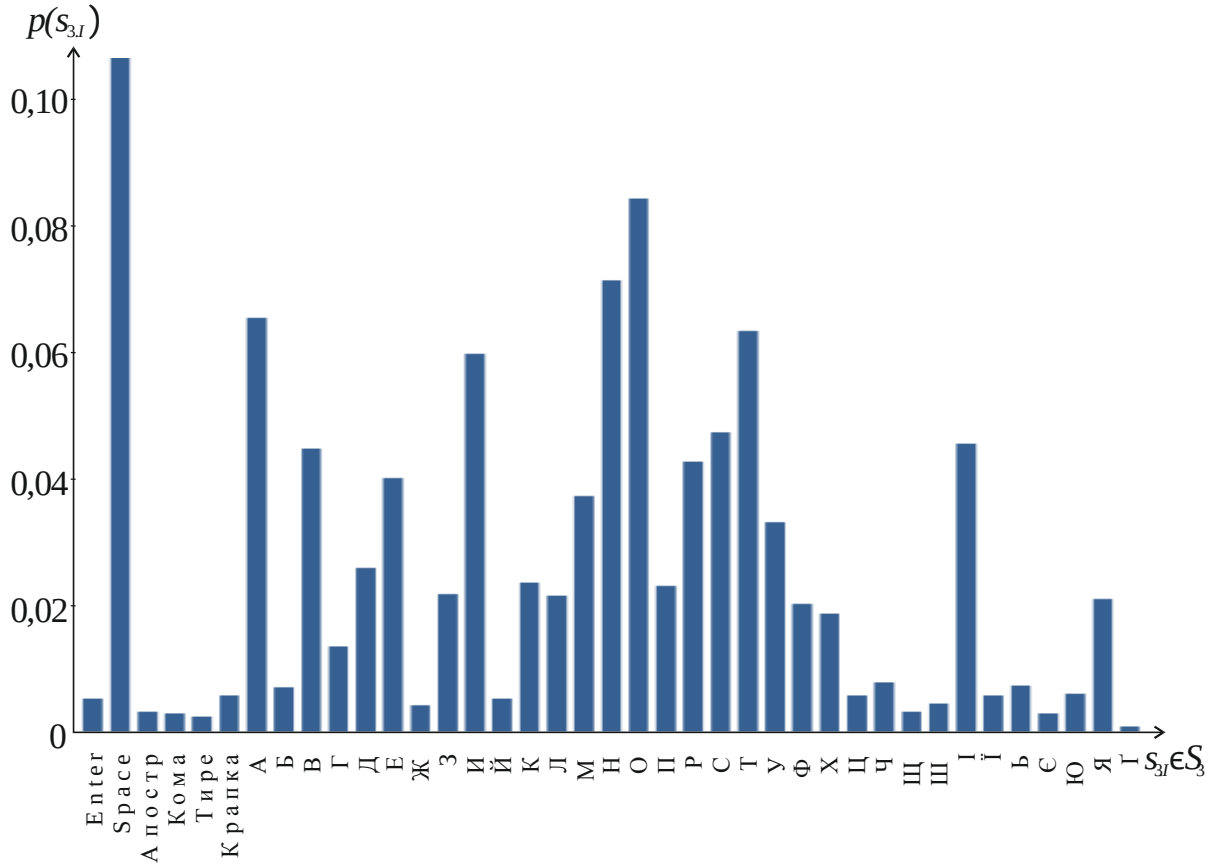


Рисунок 4.5 – Гістограма імовірності появи символів початкового алфавіту S_1 україномовного тексту

Слід відзначити наближеність статистичних характеристик, одержаних при дослідженні обраного для експерименту фрагменту тексту, до середньостатистичних характеристик імовірності появи символів українського алфавіту в довільних текстах, що зумовлює апріорно низьку криптостійкість шифротексту при безпосередньому застосуванні шифрів зсуву для його шифрування.

Основні дані статистичного аналізу складу тексту і результати реалізації алгоритму кодування Хаффмена систематизоно наведені в таблиці 4.2.

В таблиці 4.2 наведені:

– спосіб кодування символів текстового алфавіту кодами ASCII (алфавіт S_1)

– деталізовані статистичні характеристики тексту (кількість символів $s_{1.i} \in S_1$ та розрахункові статистичні імовірності появи символів

– $s_{1.i} \in S_1$ в тексті T_1;

– результати застосування алгоритму оптимального кодування Хаффмена (оптимальний алфавіт S_2);

– маски кодів Хаффмена;

– опис дерева Хаффмена (у вигляді масок кодів Хаффмена і нумерації гілок дерева).

Таблиця 4.2 – Вхідні і вихідні дані алгоритму оптимального кодування

i	Сим-вол	Код ASCII ($s_{1.i} \in S_1$)	Кількість символів $s_{1.i} \in S_1$	Імовір-ність $p(s_{1.i}) \in P_1$	Коди Хаффмена ($s_{2.i} \in S_2$)	Маска коду Хаффмена	Номер гілки дерева
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1.	Enter	00001010	13	0,0052	11011001	00000001	33
2.	Space	00100000	264	0,1056	010	001	13
3.	`	00100110	8	0,0032	001011100	000000001	8
4.	,	00101100	7	0,0028	110110000	000000001	31
5.	-	00101110	6	0,0024	0010111011	0000000001	10
6.	.	00101111	14	0,0056	0010100	0000001	5
7.	А	10000000	162	0,0648	1001	0001	20
8.	Б	10000001	17	0,0068	0110110	0000001	16
9.	В	10000010	111	0,0444	11111	00001	39
10.	Г	10000011	33	0,0132	011010	000001	15
11.	Д	10000100	64	0,0256	01100	00001	14
12.	Е	10000101	99	0,0396	11001	00001	30
13.	Ж	10000110	10	0,004	10101010	00000001	23
14.	З	10000111	54	0,0216	00000	00001	0
15.	И	10001000	148	0,0592	0111	0001	18
16.	Й	10001001	13	0,0052	11011010	00000001	34
17.	К	10001010	58	0,0232	00100	00001	4

Таблиця 4.2 – Завершення

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
18.	Л	10001011	53	0,0212	110111	000001	36

19.	М	10001100	92	0,0368	11000	00001	27
20.	Н	10001101	177	0,0708	1011	0001	26
21.	О	10001110	209	0,0836	1110	0001	37
22.	П	10001111	57	0,0228	00001	00001	1
23.	Р	10010000	106	0,0424	11110	00001	38
24.	С	10010001	117	0,0468	0011	0001	12
25.	Т	10010010	157	0,0628	1000	0001	19
26.	У	10010011	82	0,0328	10100	00001	21
27.	Ф	10010100	50	0,02	110100	000001	28
28.	Х	10010101	46	0,0184	101011	000001	25
29.	Ц	10010110	14	0,0056	11011011	00000001	35
30.	Ч	10010111	19	0,0076	1010100	0000001	22
31.	Щ	10011000	8	0,0032	00101111	00000001	11
32.	Ш	10011001	11	0,0044	10101011	00000001	24
33.	І	10011010	113	0,0452	0001	0001	2
34.	Ї	10011011	14	0,0056	0010101	0000001	6
35.	Ь	10011100	18	0,0072	0110111	0000001	17
36.	Є	10011101	7	0,0028	110110001	000000001	32
37.	Ю	10011110	15	0,0060	0010110	0000001	7
38.	Я	10011111	52	0,0208	110101	000001	29
39.	Ґ	10100000	2	0,0008	0010111010	0000000001	9

В додатку В наведено різні форми представлення вхідного тексту, що отримуються за даними таблиці 4.2 у відповідності до алгоритму реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем:

– T_1 - первинного тексту;

– $T_{НС}$ - закодованого кодом Хаффмена тексту;

– $T'_{НС}$ - потокового представлення закодованого кодом Хаффмена тексту.

Особливістю поданого в додатку В варіанту закодованого кодом Хаффмена тексту $T_{НС}$ є доповнення його компонентами, необхідними для декодування тексту (таблицею кодування). Таблиця кодування складається з трьох шифрів, наведених в таблиці 4.2:

– маски кодів Хаффмена;

– кодів Хаффмена ($s_{2.i} \in S_2$);

– кодів ASCII ($s_{1.i} \in S_1$) .

Співставлення зазначених складових дозволяє відновити таблицю кодування символів тексту кодами Хаффмена з потокового бінарного коду і виконати декодування коду Хаффмена в коди ASCII або інші варіації початкового коду, тобто, відновити вхідний в кодування Хаффмена текст.

На підставі потокового представлення закодованого кодом Хаффмена тексту $T'_{НС}$ було проведено три експериментальних розбиття бінарного коду $T'_{НС}$ для різних варіантів значень розрядності коду шифрування r.

Перший експериментальний варіант розбиття бінарного коду $T'_{НС}$ було зроблено для розрядності коду шифрування r=4. Оптимальний нерівномірний код Хаффмена було розподілено на $|T''_{НС4}|$=2887 кодових комбінацій символів тексту $T''_{НС}$. Основні результати і статистичні дані представлені в таблиці 4.3, а деталізовані дані і результати експерименту (проміжні та фінальні коди і шифри) – в додатку Г.

На рис. 4.6 відображена гістограма імовірності появи символів алфавіту S_3 при шифруванні обраного україномовного тексту, побудована за даними таблиці 4.3

Таблиця 4.3 – Характеристики рівномірного коду шифрування для r=4

i	Код символу ($s_{3.i} \in S_3$)	Кількість символів $s_{3.i} \in S_3$	Імовір-ність $p(s_{3.i}) \in P_3$	i	Код символу ($s_{3.i} \in S_3$)	Кількість символів $s_{3.i} \in S_3$	Імовір-ність $p(s_{3.i}) \in P_3$
1	0000	128	0,044337	9	1000	165	0,057153
2	0001	137	0,047454	10	1001	187	0,064773

3	0010	137	0,047454	11	1010	176	0,060963
4	0011	202	0,069969	12	1011	187	0,064773
5	0100	137	0,047454	13	1100	183	0,063388
6	0101	191	0,066159	14	1101	203	0,070315
7	0110	175	0,060617	15	1110	247	0,085556
8	0111	225	0,077936	16	1111	207	0,071701

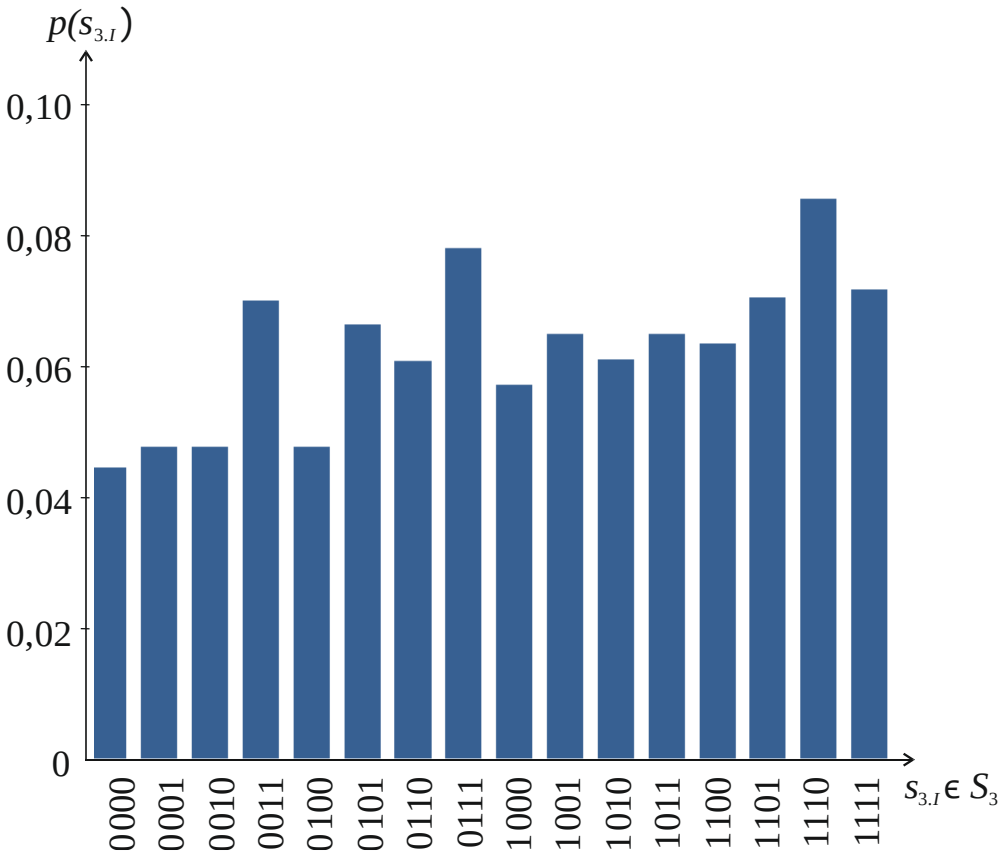


Рисунок 4.6 – Гістограма імовірності появи символів кодів алфавіту шифрування україномовного тексту T_1 для розрядності коду шифру r=4

Дані таблиці 4.3 і гістограми підтверджують здатність руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями алфавітів S_1 і S_3 для розрядності r=4 двійкового коду символів алфавіту S_3.

Другий експериментальний варіант розбиття бінарного коду $T'_{НС}$ було зроблено для розрядності коду шифрування r=5. Основні результати і

статистичні дані представлені в таблиці 4.4, а деталізовані дані і результати експерименту (проміжні та фінальні коди і шифри) – в додатку Д.

Таблиця 4.4 – Характеристики рівномірного коду шифрування для r=5

i	Код символу ($s_{3.i} \in S_3$)	Кількість символів $s_{3.i} \in S_3$	Імовір-ність $p(s_{3.i}) \in P_3$	i	Код символу ($s_{3.i} \in S_3$)	Кількість символів $s_{3.i} \in S_3$	Імовір-ність $p(s_{3.i}) \in P_3$
1	00000	60	0,019497	17	10000	65	0,02383
2	00001	66	0,025563	18	10001	77	0,031196
3	00010	83	0,027730	19	10010	46	0,015165
4	00011	57	0,023830	20	10011	105	0,042461
5	00100	60	0,021231	21	10100	71	0,025563
6	00101	57	0,022964	22	10101	96	0,037262
7	00110	70	0,027296	23	10110	66	0,029029
8	00111	122	0,041594	24	10111	92	0,042028
9	01000	53	0,023397	25	11000	88	0,036395
10	01001	60	0,027296	26	11001	79	0,035095
11	01010	74	0,034229	27	11010	64	0,02773
12	01011	74	0,02383	28	11011	102	0,044194
13	01100	73	0,034229	29	11100	86	0,031196
14	01101	72	0,029896	30	11101	97	0,043761
15	01110	99	0,041594	31	11110	95	0,041594
16	01111	90	0,038128	32	11111	71	0,031196

Оптимальний нерівномірний код Хаффмена було розподілено на $|T''_{НС5}|$=2470 кодових комбінацій символів тексту $T''_{НС}$.

На рис. 4.7 відображена гістограма імовірності появи символів алфавіту S_3 при шифруванні обраного україномовного тексту, побудована за даними таблиці 4.4

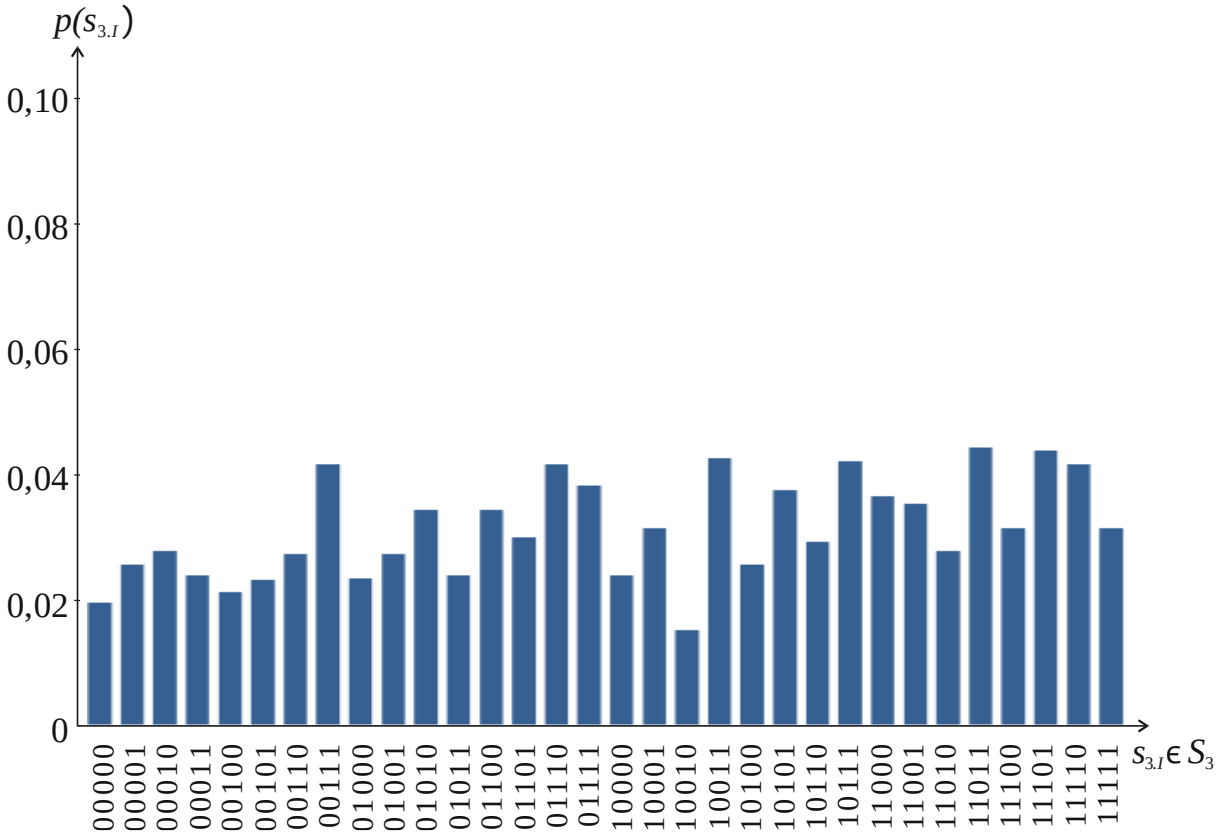


Рисунок 4.7 – Гістограма імовірності появи символів кодів алфавіту шифрування україномовного тексту T_1 для розрядності коду шифру r=5

Дані таблиці 4.4 і гістограми рис. 4.7 підтверджують здатність руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями алфавітів S_1 і S_3 для розрядності r=5 двійкового коду символів алфавіту S_3.

Третій експериментальний варіант розбиття бінарного коду $T'_{НС}$ було зроблено для розрядності коду шифрування r=6 (основні дані і результати – таблиця 4.3, а деталізовані дані і результати експерименту – в додатку Г.

Таблиця 4.5 – Характеристики рівномірного коду шифрування для r=6

i	Код символу ($s_{3.i} \in S_3$)	Кількість символів $s_{3.i} \in S_3$	Імовір-ність $p(s_{3.i}) \in P_3$	i	Код символу ($s_{3.i} \in S_3$)	Кількість символів $s_{3.i} \in S_3$	Імовір-ність $p(s_{3.i}) \in P_3$
1	000000	19	0,00987	33	100000	20	0,01039
2	000001	24	0,012468	34	100001	18	0,009351
3	000010	17	0,008831	35	100010	32	0,016623

4	000011	26	0,013506	36	100011	26	0,013506
5	000100	31	0,016104	37	100100	14	0,007273
6	000101	18	0,009351	38	100101	24	0,012468
7	000110	10	0,005195	39	100110	46	0,023896
8	000111	32	0,016623	40	100111	52	0,027013
9	001000	12	0,006234	41	101000	27	0,014026
10	001001	34	0,017662	42	101001	31	0,016104
11	001010	28	0,014545	43	101010	32	0,016623
12	001011	17	0,008831	44	101011	32	0,016623
13	001100	28	0,014545	45	101100	18	0,009351
14	001101	30	0,015584	46	101101	30	0,015584
15	001110	37	0,019221	47	101110	40	0,020779
16	001111	23	0,011948	48	101111	51	0,026494
17	010000	18	0,009351	49	110000	36	0,018701
18	010001	19	0,00987	50	110001	33	0,017143
19	010010	15	0,007792	51	110010	19	0,00987
20	010011	31	0,016104	52	110011	43	0,022338
21	010100	27	0,014026	53	110100	24	0,012468
22	010101	47	0,024416	54	110101	34	0,017662
23	010110	23	0,011948	55	110110	32	0,016623
24	010111	22	0,011429	56	110111	42	0,021818
25	011000	33	0,017143	57	111000	56	0,029091
26	011001	26	0,013506	58	111001	46	0,023896
27	011010	16	0,008312	59	111010	40	0,020779
28	011011	36	0,018701	60	111011	48	0,024935
29	011100	47	0,024416	61	111100	26	0,013506
30	011101	42	0,021818	62	111101	48	0,024935
31	011110	22	0,011429	63	111110	34	0,017662
32	011111	40	0,020779	64	111111	21	0,010909

Оптимальний нерівномірний код Хаффмена для розрядності коду шифрування r=6 було розподілено на $|T''_{НС6}|$=1925 кодових комбінацій символів тексту $T''_{НС}$. На рис. 4.8 відображена гістограма імовірності появи символів алфавіту S_3 при шифруванні обраного україномовного тексту, побудована за даними таблиці 4.5

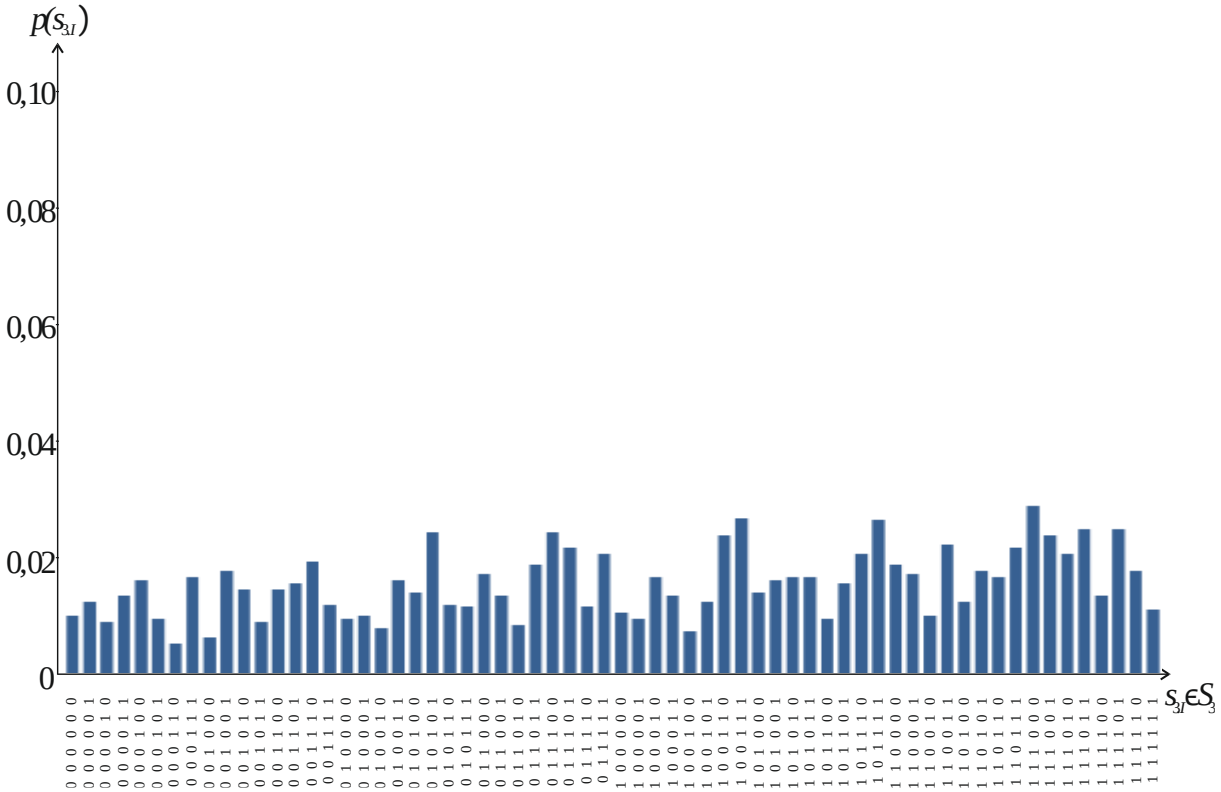


Рисунок 4.8 – Гістограма імовірності появи символів кодів алфавіту шифрування україномовного тексту T_1 для розрядності коду шифру r=6

Дані таблиці 4.5 і гістограми рис. 4.8 підтверджують здатність руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями алфавітів S_1 і S_3 для розрядності r=6 двійкового коду символів алфавіту S_3.

4.3 Висновки

В четвертому розділі кваліфікаційної роботи проведена апробація методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Апробація проведена моделюванням роботи методу та алгоритмів його реалізації на текстах з латинських і українських алфавітів.

В ході апробації отримано наступні основні результати.

Відзначити наближеність статистичних характеристик, одержаних при дослідженні обраного для експерименту тексту, до середньостатистичних характеристик імовірності появи символів алфавітів в довільних текстах, що зумовлює апріорно низьку криптостійкість шифротексту при безпосередньому застосуванні шифрів зсуву для його шифрування.

Визначено можливість зміни складу алфавіту шифрування процедурою оптимального нерівномірного кодування Хаффмена, при чому в усіх проведених експериментах отримано повноформатні алфавіти з 2^r символів розрядністю r біт незалежно від кількості символів і розрядності вхідного алфавіту.

Доведено здатність руйнування процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями алфавітів тексту і шифрування для різних розрядностей шифрів зсуву.

Проведено апробацію роботи методу для різної розрядності шифрів зсуву.

Таким чином, в четвертому розділі доведено дієвість методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем та алгоритмів його реалізації.

ВИСНОВКИ

В роботі комплексно розв’язано задачу розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем, а саме:

– досліджено використовувані в комп’ютерних системах способи і методи кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів. На підставі виявлених закономірностей визначено перспективні напрями наукових досліджень і сформулювано постановку задачі;

– розроблено математичну і структурно-логічну модель методу;

– визначено основні положення методу та розроблено алгоритми його реалізації;

– обґрунтовано ефективність запропонованого методу і алгоритмів його реалізації моделюванням.

Застосування оптимального кодування для попередньої підготовки даних до шифрування згідно із запропонованим методом дозволяє підвищити криптостійкість алгоритмів шифрування за рахунок зміни потужності і статистичних властивостей кодового алфавіту шифрування, а також забезпечує підвищену гнучкість щодо адаптації розрядності шифрокодів до потреб алгоритму шифрування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Методи і алгоритми захисту інформаційних ресурсів комп'ютерних систем : навч. посібник / В.М. Джулій, Ю.П. Кльоц, І.В. Муляр, В.М. Чешун. – Хмельницький : ХмНУ, 2020. – 196 с.

2 Мережні інформаційні технології: навчальний посібник / О. А. Мясіщев, В. М. Джулій, С. Р. Красильников, В. М. Чешун. – Хмельницький : ХНУ, 2012. – 422 с.

3 Курко А. М. Введення в теорію інформації: посібник до вивчення дисципліни / А. М. Курко, В. Я. Решетник – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017. – 108 с.

4 Беркман Л.Н. Основні поняття та теореми теорії інформації: навчальний посібник для самостійної роботи студентів ВНЗ / Беркман Л.Н., Комарова Л.О., Чумак О.І. – Київ: ДУТ НННТІ, 2015. – 91 с.

5 Класифікація основних методів кодування і кодів [Електронний ресурс] / Портал «stud.com.ua». – Режим доступу: https://stud.com.ua/171429/tehnika/klasifikatsiya_osnovnih_metodiv_koduvannya_kodiv (дата звернення 30.10.2020). – Назва з екрана.

6 Захист інформації в комп’ютерних системах та мережах: навчальний посібник / С.Г.Семенов, А.О.Подорожняк, О.І.Баленко, С.Ю.Гавриленко. – Х.: НТУ «ХПІ», 2014. – 251 с.

7 Кодирование текстовой информации [Електронний ресурс] / Режим доступу: http://school497.ru/download/u/02/les10/les.html (дата звернення 28.10.2019). – Назва з екрана.

8 Тарнавський, Ю. А. Технології захисту інформації: підручник / Ю. А. Тарнавський ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с.

9 Помехоустойчивое кодирование и декодирование в дискретных КПС [Електронний ресурс] / Портал «wiki». – Режим доступу: https://ru.bmstu.wiki/Помехоустойчивое_кодирование_и_декодирование_в_дискретных_КПС (дата звернення 30.10.2020). – Назва з екрана.

10 Романюк М.І. Основи теорії інформації та кодування: лабораторний практикум: навчальний посібник / М.І. Романюк, Г.Г. Власюк; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 81 с.

11 Помехоустойчивое кодирование с использованием различных кодов [Електронний ресурс] / Портал «wiki». – Режим доступу: https://uk.wikipedia.org/wiki/Центральний_процесор (дата звернення 30.10.2019). – Назва з екрана. https://habr.com/ru/post/111336/

12 Остапов С. Е. Технології захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

13 Бойко Ю. М. Теоретичні аспекти підвищення завадостійкості й ефективності обробки сигналів в радіотехнічних пристроях та засобах телекомунікаційних систем за наявності завад : монографія / Ю. М. Бойко, В. А. Дружинінін, С. В. Толюпа. - Київ : Логос, 2018. - 227 с.

14 Прикладна криптологія : системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.

15 Шинкарук О.М. Основи функціонування багатоканальних систем передачі інформації: навч. посіб./ О.М. Шинкарук, Ю.М. Бойко, І.І. Чесановський. – Хмельницький : ХНУ, 2011. – 245с.

16 Кодування джерел інформації та каналів зв’язку: навчальний посібник / [Беркман Л.Н., Бондарчук А.П., Гайдур Г.І., Чумак Н.С.]. – Київ: ННІТІ ДУТ, 2018. – 91 с.

17 Майданюк В. П. Кодування та захист інформації. навчальний посібник / В. П. Майданюк. – Вінниця:ВНТУ, 2009. – 164 с.

18 Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications / [Martin Tomlinson, Cen Jung Tjhai, Marcel A. Ambroze, Mohammed Ahmed, Mubarak Jibril.]. – Cham, Switzerland : Springer, 2017. – 527 p.

19 Alencar & Marcelo S Information theory / Alencar & Marcelo S. – NEWYORK : Momentum Press , LLC, 2015. – 178 p.

20 Письмак Д. О. Перешкодостійке кодування повідомлення електронного підпису ./ Д. О. Письмак, В. П. Малайчук, С. В. Клименко. //

Актуальні проблеми автоматизації та інформаційних технологій. – 2017. – Том 21. – С. 123-131.

21 Здробилко А. В. Завадостійке кодування в цифрових системах зв'язку / А. В. Здробилко, С. В. Денбновецкий // Цифрові технології. – 2015. – Вип. 17. – С. 30-34.

22 Погребняк Л.М. Оцінка ефективності завадостійкого та просторового кодувань в нестаціонарних частотно-селективних каналах систем військового радіозв'язку / Л .М. Погребняк, М.І. Науменко. // Збірник наукових праць ВІТІ. – 2017. – № 4 –С. 103-110.

23 Гребенюк О. П. Застосування завадостійкого кодування в системах зв'язку і передачі даних комплексів радіомоніторингу для забезпечення достовірності інформаційного обміну / О. П. Гребенюк, В. Д. Меленський, В. І. Коріненко // Проблеми створення, випробування, застосування та експлуатації складних інформаційних систем. – 2015. – Вип. 11. - С. 44-50.

24 Galić I. Image compression with B-tree coding algorithm enhanced by data modelling with Burrows-Wheeler transformation / Irena Galić, Časlav Livada, Branka Zovko-Cihlar. //Journal for Control, Measurement, Electronics, Computing and Communications. – 2017. – Volume 57, Issue 1. – P. 76-88.

25 Гончар Р.М. Оптимальне кодування як засіб підвищення захищеності передачі шифрованих даних / Р.М. Гончар, В.С. Орленко, В.Ю. Тітова, В.М. Чешун // «Інтелектуальний потенціал – 2020» - збірник наукових праць молодих науковців і студентів /Колектив авторів – Хмельницький: ПВНЗ УЕП, 2020. – Ч.2. – С.18-29.

26 Оптимальне нерівномірне кодування в підвищенні криптостійкості шифрів / Р.М. Гончар, В.С. Орленко, В.Ю. Тітова, В.М. Чешун // Тези доповідей XVI Міжнародної НПК "Військоваосвіта і наука: сьогодення та майбутнє" / за заг. редакцією Ігоря Толока.– К. : ВІКНУ, 2020. – С.123-124.

ДОДАТОК А

(обов'язковий)

Копії публікацій

к.т.н., доц. Чешун В.М. (ХмНУ)
к.т.н., доц. Орленко В.С. (ХмНУ)
к.т.н., доц. Тітова В.Ю. (ХмНУ)
Гончар Р.М. (ХмНУ)

ОПТИМАЛЬНЕ НЕРІВНОМІРНЕ КОДУВАННЯ В ПІДВИЩЕННІ КРИПТОСТІЙКОСТІ ШИФРІВ

В умовах стрімкого розвитку інформаційних технологій, постійного збільшення обсягів інформації в кіберпросторі і зростання її цінності, а також через появу нових загроз щодо її цілісності і конфіденційності надзвичайної актуальності набувають заходи кібербезпеки. Одним із базових заходів є криптографічний захист даних, про що свідчить поява і масштабне використання великої кількості методів та алгоритмів симетричного й асиметричного шифрування з різними функціональними можливостями і принципами дії (алгоритми DES-базовий, подвійний і потрійний DES, IDEA, ГОСТ 28147, Діффі-Хелмана, RSA тощо [1]) та спроби їх постійного вдосконалення.

Підвищення криптостійкості алгоритмів шифрування можна досягти попередньою підготовкою вхідних даних, в ході якого забезпечується порушення статистичних даних повторюваності символів вхідного тексту, тобто, збільшення характеристик його ентропії. Одним із варіантів такої підготовки вхідного тексту може бути застування методів оптимального нерівномірного кодування - ОНК (кодування Шеннона-Фано, Хафмана [2]).

В узагальненому алгоритмі підготовки даних до криптографічного шифрування із застосуванням ОНК можна виділити три базових операції:

1. Заміна кодових комбінацій K_i символів вхідного тексту, що мають однакову розрядність, кодовими комбінаціями K_i' різної розрядності із урахуванням статистичних характеристик появи зазначених символів в тексті.
2. Формування двійкового представлення вихідного тексту у вигляді послідовності кодових комбінацій замінки K_i' різної розрядності.
3. Розподіл одержаної послідовності на кодові комбінації K_i'' однакової розрядності згідно з вимогами застосовуваного криптографічного алгоритму.

Особливістю реалізації етапу 3 є розподіл видозміненої стиснутої двійкової послідовності на кодові комбінації фіксованої розрядності K_i'' без урахування характеру входження в неї початкових символів повідомлення K_i, що зумовлює можливість виникнення нетипових для звичайного кодування рівномірними кодами ситуацій:

- декілька коротких комбінацій K_i' можуть стати частинами однієї комбінації K_i'';
- кодові комбінації K_i' можуть розбиватися на частини між декількома комбінаціями K_i''.

Оскільки характер розподілу нерівномірних кодових комбінацій K_i' між рівномірними кодовими комбінаціями K_i'' без урахування особливостей реалізації алгоритму ОНК апріорно є непередбачуваним зловмиснику, такий підхід, окрім зменшення розмірів призначеного для передачі двійкового коду вихідного тексту, збільшує ентропійні властивості зашифрованого тексту і його стійкість до зламу незалежно від застосовуваного методу криптографічного шифрування.

Список використаних джерел:

1. Методи і алгоритми захисту інформаційних ресурсів комп'ютерних систем : навч. посібник / В.М. Джулій, Ю.П. Кльоц, І.В. Муляр, В.М. Чешун. – Хмельницький : ХмНУ, 2020. – 196 с.
2. Кодування джерел інформації та каналів зв'язку : навч. посібник / Л.Н. Беркман, А.П. Бондарчук, Г.І. Гайдур, Н.С. Чумак. – Київ: НТІТ ДУТ, 2018. – 91с.

Перелік посилань

1.Абазина Е.С. Цифровая стеганография: состояние и перспективы / Е.С. Абазина, А.А. Ерунов // Системы управления, связи и безопасности. – 2016. – № 2. – С. 182–201.

2.Колесов В.В. Применение дискретных хаотических алгоритмов в широкополосных телекоммуникационных системах / В.В. Колесов, А.И. Полубехин, Е.П. Чигин, А.Д. Юрин // Вестник СибГУТИ. – 2016. – № 3. – С. 77–92.

3.Барабаш О. В. Методи пошуку оптимальних маршрутів графа структури розгалуженої інформаційної мережі за заданим критерієм оптимальності при різних обмеженнях / О. В. Барабаш, І. П. Саланда, А. П. Мусієнко // Наукові записки Українського науково-дослідного інституту зв'язку. -К.: УНДІЗ, 2016. - №2 (42). - С 99-106.

Оптимальне кодування як засіб підвищення захищеності передачі шифрованих даних

Гончар Р. М., Орленко В.С., Чешун В.М.
Хмельницький національний університет

Постійне збільшення обсягів інформації в кіберпросторі і зростання її цінності зумовлює зацікавленість конкуруючих сторін і зловмисників у незаконному заволодінні нею, що створює постійну появу нових загроз щодо цілісності і конфіденційності інформації і актуальність заходів її захисту. Одним із основних способів захисту даних є шифрування, про що свідчить поява великої кількості методів та алгоритмів шифрування з різними функціональними можливостями і принципами дії (алгоритми DES-базовий, подвійний і потрійний DES, IDEA, ГОСТ 28147, Діффі-Хелмана, RSA тощо [1]) та тенденція до їх постійного вдосконалення.

Підвищення криптостійкості алгоритмів шифрування досягається як розробкою нових їх реалізацій, так і модернізацією-вдосконаленням існуючих або їх комбінуванням.

Проведені дослідження показують, що підвищення криптостійкості алгоритмів шифрування можна досягти попередньою підготовкою вхідних даних, в ході якого забезпечується порушення статистичних даних повторюваності символів вхідного тексту, тобто, збільшення характеристик його ентропії. Одним із варіантів такої підготовки вхідного тексту може бути застування методів оптимального нерівномірного кодування.

Для демонстрації можливості збільшення криптостійкості алгоритмів шифрування попередньою підготовкою вхідних даних оптимальним кодуванням обрано два класичних методи:

– кодування Хаффмена як класичний метод оптимального

ентропійного нерівномірного кодування даних, що дає стабільний оптимальний код на виході у відповідності до статистичних властивостей алфавіту вхідного тексту;

– шифри зсуву (заміни) як найпростіший варіант шифрування даних, що дозволяє наочно спостерігати вплив попередньої підготовки даних на криптостійкість алгоритму шифрування.

Недоліком шифрів зсуву є збереження статистичних характеристик появи символів первинного алфавіту (вхідного тексту) у вторинному алфавіті (зашифрованому тексті), що зумовлює їх низьку криптостійкість

Таким чином, криптостійкість шифрів зсуву може бути підвищена зміною властивостей алфавіту шифрування, для чого застосовується оптимальне кодування Хаффмена.

Стосовно кодування Хаффмена можна сказати наступне:

– оптимальне кодування Хаффмена є різновидом ефективного кодування;

– оптимальне кодування Хаффмена відноситься до класу ентропійних кодів;

– коди Хаффмена є нерівномірними кодами;

– коди Хаффмена є префіксними кодами;

– головною відмінністю коду Хаффмена від коду Шеннона-Фано є те, що він завжди дає оптимальний варіант ентропійного кодування за наявними статистичними даними;

– коди Хаффмена, як і інші методи ефективного кодування, безпосередньо не призначені для вирішення задач шифрування і захисту даних;

– стиснення даних оптимальним кодуванням Хаффмена дає позитивний ефект для захисту даних через зменшення розмірів повідомлень, що передаються (менша імовірність втрат інформації при передачі);

– відхилення варіанту кодування Хаффмена від примітивного кодування символів також ускладнює дешифрування тексту.

Для визначення перспектив застосування оптимального кодування Хаффмена в задачах криптографічного захисту дослідимо принципи цього кодування.

Побудова оптимального нерівномірного коду за методикою Хаффмена виконується за загальним алгоритмом, що включає наступні етапи:

– всі символи, що кодуються, упорядковуються в порядку зменшення ймовірностей;

– останні два символи впорядкованої множини (вони повинні мати найменші значення ймовірностей) замінюються допоміжним символом, значення ймовірності для якого визначається сумарною ймовірністю елементів, що замінюються;

– всі елементи нової множини знову упорядковуються на зменшення

ймовірностей;

– виконання попередніх двох операці повторюється до отримання єдиного допоміжного символу.

Для визначення кодових комбінацій символів виконується зворотний аналіз виконаних об'єднань.

Тобто, двом останнім символам, при об'єднанні яких було отримано символ з ймовірністю 1, присвоюються значення коду 0 і 1. Після цього розглядаються символи попереднього рівня, які прийняли участь в утворенні останніх допоміжних символів. Аналогічним чином їм ставляться у відповідність значення 0 та 1, які дописуються в молодший розряд кодових комбінацій.

Завершення кодування Хаффмена відбувається після досягнення етапу, на якому кодові комбінації будуть співставлені у відповідність всім символам вхідного алфавіту.

Отриманий за наведеним алгоритмом методикою нерівномірний код є оптимальним кодом Хаффмена для використаного в тексті алфавіту.

Для визначення перспектив застосування оптимального нерівномірного коду Хаффмена для збільшення криптостійкості алгоритмів шифрування розглянемо приклад практичного застосування методу кодування Хаффмена.

В якості прикладу дослідимо застосування методу Хаффмена для довільного набору з 100 латинських символів:

ADBCBADCSDCASCZDSMMMCCMMM
MAWSDSASWZDSMWUSCCMMMMEEE
WWBWBUDWSDWUSCCCCMMMMCWSD
SASCZDSMMMMMMEWWZDSMMZDSMM

Потужність застосованого в наданому для кодування тексті алфавіту дорівнює 10, оскільки в ньому використано 10 символів: A, B, C, D, E, M, S, U, W, Z.

Спочатку визначимо рекомендовану розрядність двійкових кодів для заданого алфавіту при використанні примітивних (рівномірних) кодів:

$$n = \log_2 10 = 3.322.$$

Відповідно, для результату 3.322 отримуємо мінімально-достатню розрядність примітивного коду 4.

Для аналізу використаємо простий варіант послідовного кодування символів двійкового алфавіту двійковими числами (табл. 1.1).

Таблиця 1.1 – Примітивний код для кодування вхідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Код	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Із застосуванням наведеного в таблиці 1.1 примітивного рівномірного коду початковий текст перетворюється в масив двійкових кодів загальною розрядністю 400 біт:

0000 0011 0001 0010 0001 0000 0011 0010 0110 0011
0010 0000 0110 0010 1001 0011 0110 0101 0101 0101
0010 0010 0101 0101 0101 0101 0000 1000 0110 0011
0110 0000 0110 1000 1001 0011 0110 0101 1000 0111
0110 0010 0010 0101 0101 0101 0101 0100 0100 0100
1000 1000 0001 1000 0001 0111 0011 1000 0110 0011
1000 0111 0110 0010 0010 0010 0010 0101 0101 0101
0101 0010 1000 0110 0011 0110 0000 0110 0010 1001
0011 0110 0101 0101 0101 0101 0101 0100 1000 1000
1001 0011 0110 0101 0101 1001 0011 0110 0101 0101

Дослідження статистичних властивостей вхідного тексту дозволяє визначити ймовірності появи в ньому символів алфавіту (табл. 1.2).

Таблиця 1.2 – Статистичні характеристики алфавіту вхідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Кількість повторів символу	6	4	14	12	4	25	16	3	11	5
Статистична імовірність	0.06	0.04	0.14	0.12	0.04	0.25	0.16	0.03	0.11	0.05

Наявність статистичних даних щодо імовірностей входження символів алфавіту до тексту, що подається на кодування, дозволяє застосувати алгоритм кодування Хаффмена. На рис. 1.1 зображене кодове дерево Хаффмена, побудоване за даними таблиці 1.2.

На основі кодового дерева Хаффмена формуємо кодові комбінації оптимального нерівномірного коду Хаффмена для кодування тексту (табл. 1.3).

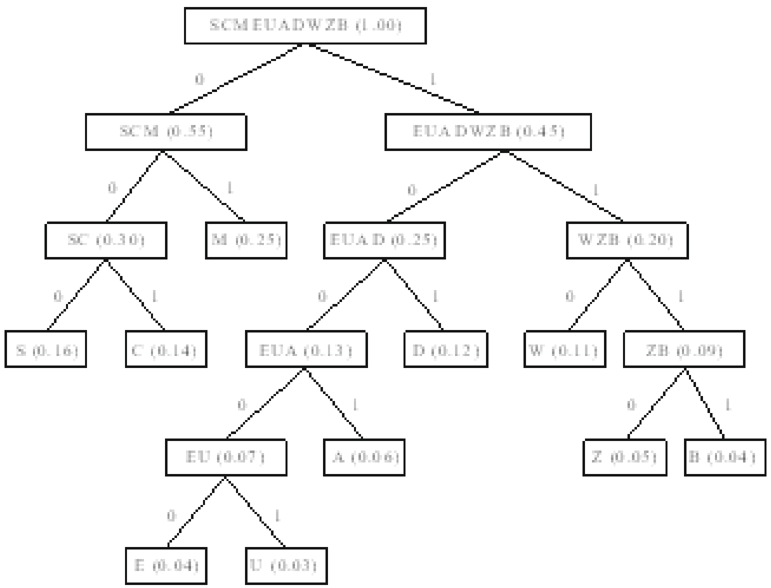


Рисунок 1.10 – Кодове дерево Хаффмена

Таблиця 1.3 – Оптимальний код Хаффмена для кодування алфавіту тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Код Хафмена	1001	1111	001	101	10000	01	000	10001	110	1110

Із застосуванням наведеного в таблиці 1.3 оптимального нерівномірного коду Хаффмена початковий текст перетворюється в масив двійкових кодів загальною розрядністю 304 біт:

1001 101 1111 001 1111 1001 101 001 000 101
001 1001 000 001 1110 101 000 01 01 01
001 001 01 01 01 01 1001 110 000 101
000 1001 000 110 1110 101 000 01 110 10001
000 001 001 01 01 01 01 10000 10000 10000
110 110 1111 110 1111 10001 101 110 000 101
110 10001 000 001 001 001 001 01 01 01
01 001 110 000 101 000 1001 000 001 1110
101 000 01 01 01 01 01 10000 110 110
1110 101 000 01 01 1110 101 000 01 01

Першочергово відзначимо позитивний ефект стиску даних із застосуванням оптимального нерівномірного кодування Хаффмена – замість 400 біт даних сам текст в оптимальному кодуванні займає лише 304 біти.

Визначимо коефіцієнт стиснення коду:

$$k = \frac{304}{400} * 100\% = 76\%.$$

Отриманий коефіцієнт є досить високим показником для несистематизованого тексту і дозволяє стверджувати, що ризик ушкодження оптимального коду тексту порівняно з початковим примітивним зменшується майже на чверть (24% пропорційно зменшенню розрядності коду). Це підтверджує ефективність використання оптимального кодування Хаффмена за призначенням для кодування тексту даного прикладу.

Для визначення перспектив застосування оптимального кодування Хаффмена для підвищення ефективності захисту даних з використанням шифрів зсуву перетворимо отриманий нерівномірний код початкового тексту в рівномірні кодові комбінації, потрібні нам для реалізації шифру зсуву.

Початково формуємо потокове (нерозривне послідовне) представлення тексту кодом Хаффмена. Для наочності і зручності подальшого аналізу коди Хаффена окремих символів з непарними номерами позицій в тексті виділено жирним шрифтом (застосовано виділення жирним символів через один для можливості наочного розрізнення кодів символів в бітовій послідовності):
100110111110011111100110100100010100110010000011110101000010101001
001010101011001110000101000100100011011101010000111010001000001001
01010101100001000010000110110111111101111110001101110000101110100010
000010010010010101010100111000010100010010000011110101000010101010
1100001101101110101000010111101010000101.

Розіб'ємо отримане потокове представлення тексту кодом Хаффмена на чотирьохрозрядні комбінації (тетради) у відповідності із розмірністю початкового варіанту примітивного кодування:

1001 1011 1110 0111 1110 0110 1001 0001 0100 1100
1000 0011 1101 0100 0010 1010 0100 1010 1010 1100
1110 0001 0100 0100 1000 1101 1101 0100 0011 1010
0010 0000 1001 0101 0101 1000 0100 0010 0001 1011
0111 1110 1111 1000 1101 1100 0010 1110 1000 1000
0010 0100 1001 0101 0101 0011 1000 0101 0001 0010
0000 1111 0101 0000 1010 1010 1100 0011 0110 1110
1010 0001 0111 1010 1000 0101

Аналіз отриманого розбиття дозволяє побачити певні особливості утворення рівномірного коду з нерівномірного:

– лише невелика кількість отриманих кодових комбінацій отриманого рівномірного коду відповідає кодовим комбінаціям символів у

реалізації коду Хаффмена (однократно зустрічаються коди 1001, 1110 і 1111);

– визначені в попередньому аналізі кодові комбінації коду Хаффмена 1001, 1110 і 1111 в рівномірному коді наявні не лише як коди символів A, B і Z відповідно – аналогічні кодові комбінації утворюються з фрагментів кодів інших символів;

– в більшості випадків кодові комбінації коду Хаффмена при переході від нерівномірного кодування до рівномірного дробленням бітової послідовності зазнають дроблення на частини між декількома кодовими комбінаціями рівномірного коду (кожна з комбінацій 1011, 1110, 0111, 1110, 0110, як і більшість інших, утворені ділять між собою фрагменти двох кодових комбінацій коду Хаффмена);

– наслідком попередньої властивості є те, що більшість кодових комбінацій рівномірного коду, отриманого дробленням бітової послідовності коду Хаффмена, містять фрагменти декількох кодових комбінацій коду Хаффмена (кожна з комбінацій 1011, 1110, 0111, 1110, 0110, як і більшість інших, утворені з фрагментів двох кодових комбінацій коду Хаффмена, а комбінації 0011 і 0011, як приклад, утворені з фрагментів трьох кодових комбінацій);

– серед кодових комбінацій рівномірного коду, отриманого дробленням бітової послідовності коду Хаффмена, зустрічаються кодові комбінації, які є фрагментами кодових комбінацій коду Хаффмена більшої розрядності (кодова комбінація 1000 в одному місці отримана урізанням з комбінації 10000, а в іншому - з комбінації 10001). При цьому можливе виокремлення частини розрядів як зі сторони молодших розрядів, так і зі сторони старших або з середньої частини довгої кодової комбінації коду Хаффмена.

Більш детальний аналіз рівномірних кодів, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради, дозволяє визначити збільшення кількості наявних кодових комбінацій порівняно з початковим примітивним кодуванням.

В наведеному нижче представленні підкреслено різні види кодових комбінацій, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради:

1001 1011 1110 0111 1110 **0110** 1001 **0001 0100 1100**
1000 0011 1101 0100 **0010 1010** 0100 1010 1010 1100
1110 0001 0100 0100 1000 1101 1101 0100 0011 1010
0010 **0000** 1001 **0101** 0101 1000 0100 0010 0001 1011
0111 1110 **1111** 1000 1101 1100 0010 1110 1000 1000
0010 0100 1001 0101 0101 0011 1000 0101 0001 0010
0000 1111 0101 0000 1010 1010 1100 0011 0110 1110
1010 0001 0111 1010 1000 0101

З прикладу чітко видно, що в отриманих дробленням бітової

послідовності оптимального нерівномірного коду Хаффмена на тетради кодових комбінаціях наявні всі 16 можливих кодових комбінацій довжиною 4 біти, а не 10, як це було в початковому примітивному коді.

В таблиці 1.4 наведено статистичні дані щодо частоти появи різних кодових комбінацій рівномірного коду, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради.

Таблиця 1.4 – Статистичні характеристики фінального рівномірного коду

№ з/п	Код	Кількість повторів коду	Статистична імовірність
1.	0000	3	0,039474
2.	0001	5	0,065789
3.	0010	6	0,078947
4.	0011	4	0,052632
5.	0100	8	0,105263
6.	0101	7	0,092105
7.	0110	2	0,026316
8.	0111	3	0,039474
9.	1000	8	0,105263
10.	1001	4	0,052632
11.	1010	8	0,105263
12.	1011	2	0,026316
13.	1100	4	0,052632
14.	1101	4	0,052632
15.	1110	6	0,078947
16.	1111	2	0,026316

Порівняння статистичних даних таблиць 1.2 і 1.4 свідчить про руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями, що співставляються символам вхідного тексту при примітивному кодуванні, а також про здатність відповідної процедури змінювати кількість використовуваних комбінацій рівномірного коду і характер їх формування.

Невідповідність кількості кодових комбінацій застосовуваного коду кількості символів кодованого алфавіту та руйнування статистичних залежностей і ентропійних взаємозв'язків між символами є передумовою підвищення криптостійкості шифрування зсувом і може бути позитивно застосоване для криптографічних методів шифрування загалом.

Перелік посилань

1 Мережні інформаційні технології: навчальний посібник / О. А. Мясіщев, В. М. Джулій, С. Р. Красильников, В. М. Чешун. – Хмельницький : ХНУ, 2012. – 422 с.

2 Курко А. М. Введення в теорію інформації: посібник до вивчення дисципліни / А. М. Курко, В. Я. Решетник – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017. – 108 с.

3 Беркман Л.Н. Основні поняття та теореми теорії інформації: навчальний посібник для самостійної роботи студентів ВНЗ / Беркман Л.Н., Комарова Л.О., Чумак О.І. – Київ: ДУТ ННІТІ, 2015. – 91 с.

4 Класифікація основних методів кодування і кодів [Електронний ресурс] / Портал «stud.com.ua». – Режим доступу: https://stud.com.ua/171429/tehnika/klasifikatsiya osnovnih metodiv koduvannya _kodiv (дата звернення 30.10.2020). – Назва з екрана.

5 Захист інформації в комп'ютерних системах та мережах: навчальний посібник / С.Г.Семенов, А.О.Подорожняк, О.І.Баленко, С.Ю.Гавриленко. – Х.: НТУ «ХПІ», 2014. – 251 с.

6 Тарнавський Ю. А. Технології захисту інформації: підручник / Ю. А. Тарнавський; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с.

7 Помехоустойчивое кодирование и декодирование в дискретных КПС [Електронний ресурс] / Портал «wiki». – Режим доступу: https://ru.bmstu.wiki/Помехоустойчивое кодирование и декодирование в ди скретных_КПС (дата звернення 30.10.2020). – Назва з екрана.

8 Остапов С. Е. Технології захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

9 Майданюк В. П. Кодування та захист інформації. навчальний посібник / В. П. Майданюк. – Вінниця:ВНТУ, 2009. – 164 с.

10 Бойко Ю. М. Теоретичні аспекти підвищення завадостійкості й ефективності обробки сигналів в радіотехнічних пристроях та засобах телекомунікаційних систем за наявності завад : монографія / Ю. М. Бойко, В. А. Дружинінін, С. В. Толюпа. - Київ : Логос, 2018. - 227 с.

11 Прикладна криптологія : системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.

12 Кодування джерел інформації та каналів зв'язку: навчальний посібник / [Беркман Л.Н., Бондарчук А.П., Гайдур Г.І., Чумак Н.С.]. – Київ: ННІТІ ДУТ, 2018. – 91 с.

13 Alencar & Marcelo S Information theory / Alencar & Marcelo S. – NEWYORK : Momentum Press , LLC, 2015. – 178 p.

14 Galić I. Image compression with B-tree coding algorithm enhanced by data modelling with Burrows-Wheeler transformation / Irena Galić, Časlav Livada, Branka Zovko-Cihlar. //Journal for Control, Measurement, Electronics, Computing and Communications. – 2017. – Volume 57, Issue 1. – P. 76-88.

ДОДАТОК Б

(обов’язковий)

Презентація роботи

Хмельницький національний університет
Факультет програмування та комп’ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп’ютерних систем і мереж

Гончар Ростислав Михайлович

Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем

спеціальність 123 – Комп’ютерна інженерія

Науковий керівник: к.т.н., доцент **Чешун Віктор Миколайович**

ЗАГАЛЬНА ХАРАКТЕРИСТИКА МАГІСТЕРСЬКОЇ РОБОТИ

Мета магістерської роботи полягає в підвищенні ефективності захисту інформаційних ресурсів комп'ютерних систем за рахунок зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Об'єктом дослідження є процес криптографічного шифрування даних в комп'ютерних системах із застосуванням криптографічних шифрів зсуву.

Предметом дослідження є методи, алгоритми і засоби підвищення криптостійкості криптографічних алгоритмів шифрування у застосуванні до криптографічних шифрів зсуву.

Задачі досліджень у роботі формулюються наступним чином:

а) дослідити використовувані в комп'ютерних системах способи і методи кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів. На підставі виявлених закономірностей визначити перспективні напрями наукових досліджень і сформулювати постановку задачі;

б) розробити математичну модель для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем;

в) визначити основні положення методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем та розробити алгоритми його реалізації;

г) обгрунтувати ефективність запропонованого методу і алгоритмів його реалізації моделюванням.

Методи досліджень базуються на основних положеннях теорії ймовірності та математичної статистики, теорії інформації та кодування, криптографії та прикладної криптології.

Наукова новизна отриманих результатів:

1. Розроблено нову математичну модель, яка враховує особливості сумісного використання методів оптимального кодування та шифрів зсуву для підвищення криптостійкості систем захисту інформаційних ресурсів комп'ютерних систем;

2. Запропоновано спосіб застосування методу оптимального кодування Хаффмена для зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву, який став основою для розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем.

Практична цінність розробленого методу полягає у обґрунтуванні можливості підвищення ефективності криптографічного захисту інформаційних ресурсів комп'ютерних систем застосуванням методів оптимального кодування перед реалізацією криптографічних методів захисту, а також у розробці алгоритмів реалізації методу.

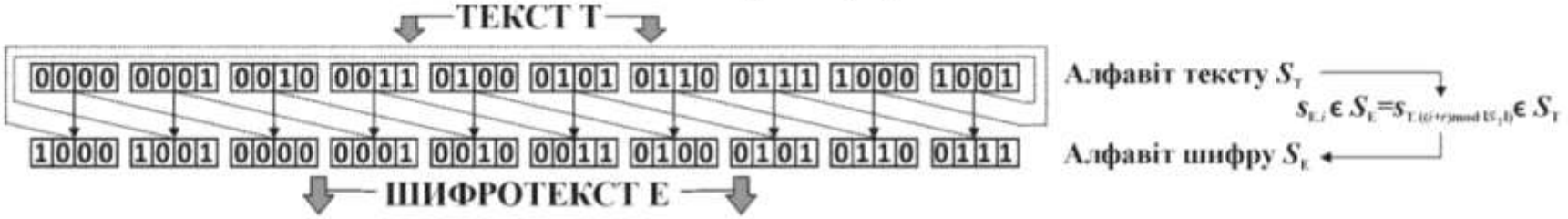
Апробація роботи. Наукові результати і основні положення дипломної роботи магістра доповідались і обговорювались на всеукраїнській і міжнародній науково-практичних конференціях.

Публікації. За темою магістерської роботи опубліковано 1 теза доповідей всеукраїнської науково-практичної конференції і 1 стаття у збірнику наукових праць молодих науковців і студентів.

ДОСЛІДЖЕННЯ СПОСОБІВ І ЗАДАЧ КОДУВАННЯ

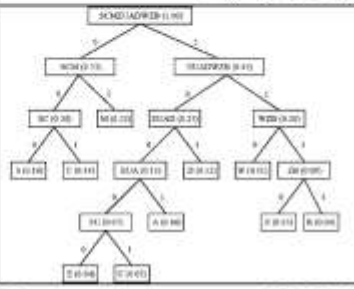
ДОСЛІДЖЕННЯ ШИФРІВ ЗСУВУ І ОПТИМАЛЬНИХ КОДІВ ХАФФМЕНА

Шифри зсуву



$|S_E|=|S_T|$; $p(s_{E,i} \in S_E)=(s_{T,(i+r) \bmod |S_T|} \in S_T)$; $H(S_E)=H(S_T)$ → легкий злам шифру

Оптимальне кодування Хаффмена

Рівномірний код тексту →	Кодування →	Оптимальний код →	Нерозривний код →	Рівномірний код виходу ↓
0000 0011 0001 0010 0001 0000 0011 0010 0110 0011 0010 0000 0110 0010 1001 0011 0110 0101 0101 0101 0010 0010 0101 0101 0101 0101 0000 1000 0110 0011 0110 0000 0110 1000 1001 0011 0110 0101 1000 0111 0110 0010 0010 0101 0101 0101 0101 0100 0100 0100 1000 1000 0001 1000 0001 0111 0011 1000 0110 0011 1000 0111 0110 0010 0010 0010 0010 0101 0101 0101 0101 0010 1000 0110 0011 0110 0000 0110 0010 1001 0011 0110 0101 0101 0101 0101 0101 0100 1000 1000 1001 0011 0110 0101 0101 1001 0011 0110 0101 0101		1001 101 1111 001 1111 1001 101 001 000 101 001 1001 000 001 1110 101 000 01 01 01 001 001 01 01 01 01 1001 110 000 101 000 1001 000 110 1110 101 000 01 110 10001 000 001 001 01 01 01 01 10000 10000 10000 110 110 1111 110 1111 10001 101 110 000 101 110 10001 000 001 001 001 001 01 01 01 01 001 110 000 101 000 1001 000 001 1110 101 000 01 01 01 01 01 10000 110 110 1110 101 000 01 01 1110 101 000 01 01	**1001**101**1111**001**1111**1001**101**001**000**101 **001**1001**000**001**1110**101**000**01**01**01001**00** **101**01**01**1001**110**000**101**000**1001**000**11 **0111**01**0**1000**01110**1000**1000**0**1001**01** **0101**10000**10000**10000**110**1101**1111**1011 **11**10001**101**1100**001**1110**1000**1000**0010 **0100**1001**01**01**01**01001**1100**001**01000**100 **1000**001**1110**1010**0001**01**01**01**1000**011 **0110**1110**101**0**00**01**0111**101**0**0010	**1001** 1011 **1110** 0111 **1110** 0110 1001 **0001** 0100 **1**100 **1000** 0011 **1101** 0100 **0010** 1010 **0100** 1010 1010 **1100** 1110 **0001** 0100 **0**100 **1000** 1101 **1101** 0100 **0011** **10**10 0010 **0**000 **1001** 0101 0101 1000 0**100** **00**10 000**1** **1011** 0**111** 1110 **1111** 1000 **1101** 1100 **0010** 1**110** 1000 1**000** 0010 **0100** **1001** 0101 0101 0011 **1000** 0**101** 000**1** **0010** 0000 1111 **0101** 0000 **1010** **1010** **1100** 0011 0**110** **1110** 1010 **0**001 **0111** 10**10** 1000 **0**101

Вплив кодування Хаффмена на алфавіти рівномірного коду тексту і виходу

	Коди символів тексту (алфавіту тексту)										Рівномірні коди виходу (утворені з коду Хаффмена тексту)															
Символ алфавіту №	S_{T1}	S_{T2}	S_{T3}	S_{T4}	S_{T5}	S_{T6}	S_{T7}	S_{T8}	S_{T9}	S_{T10}	S_{H1}	S_{H2}	S_{H3}	S_{H4}	S_{H5}	S_{H6}	S_{H7}	S_{H8}	S_{H9}	S_{H10}	S_{H11}	S_{H12}	S_{H13}	S_{H14}	S_{H15}	S_{H16}
Рівномірний код	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	0010	0011	0100	0101	0110	0111
Статистична імовірність	0.06	0.04	0.14	0.12	0.04	0.25	0.16	0.03	0.11	0.05	0,04	0,07	0,08	0,05	0,11	0,09	0,03	0,04	0,11	0,05	0,11	0,03	0,05	0,05	0,08	0,03
Код Хаффмена	1001	1111	001	101	10000	01	000	10001	110	1110																

Коди виходу відрізняються від кодів алфавіту тексту як за кількістю, так і за статистичними характеристиками

Перший науковий результат **МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ**

$$M = < T_1 , T_{HC}, T'_{HC}, T''_{HC}, T_{Ew}, P_1, S_1, S_2, S_3, E, F_{RD}, F_P, F_{HC}, F_{TC}, F_{SC}, F_{PD}, F_E, F_D >$$

де: T_1 – відкритий вхідний текст;

Оптимальне кодування тексту кодом Хаффмена:

T_{HC} – первинне нерівномірне кодування; T'_{HC} – нерозривний бінарний код; T''_{HC} – рівномірне розбиття T'_{HC};

Шифротекст: T_{Ew} – шифрований шифром зсуву текст T''_{HC};

$S_1 : \{s_{1.1}, s_{1.2}, \ldots, s_{1.i}, \ldots, s_{1.k}\}$ – множина кодів символів первинного рівномірного алфавіту вхідного тексту;

$P_1 : \{p_{1.1}, p_{1.2}, \ldots, p_{1.i}, \ldots, p_{1.k}\}$ – статистичні ймовірності появи символів первинного алфавіту;

$S_2 : \{s_{2.1}, s_{1.2}, \ldots, s_{2.i}, \ldots, s_{2.k}\}$ – множина кодів символів вторинного нерівномірного алфавіту оптимального коду;

$S_3 : \{s_{3.1}, s_{3.2}, \ldots, s_{3.i}, \ldots, s_{3.n}\}$ – множина кодів символів третинного рівномірного алфавіту оптимального коду;

$E: \{e_1, e_2, \ldots, e_i, \ldots, e_m\}$ – множина кодів символів алфавіту шифрування (шифру зсуву);

$F_{RD}(T_x) \rightarrow S_x$ – алгоритмічна функція формування алфавіту з тексту

$F_P(T_x) \rightarrow P_x$ – алгоритмічна функція визначення статистичних даних тексту

$F_{HC}(S_x, P_x) \rightarrow S_{HC}$ – алгоритмічна функція оптимального кодування алфавіту

$F_{TC}(T_x, S_x, S_y) \rightarrow T_y$ – алгоритмічна функція заміни кодів алфавіту в коді тексту з S_x на S_y

$F_{SC}(T_x) \rightarrow T_{SCx}$ – алгоритмічна функція утворення нерозривного бінарного коду тексту

$F_{PD}(T_{SCx}, r) \rightarrow T_{UC}$ – алгоритмічна функція розбиття бінарного коду на рівномірні комбінації розрядності r

$F_E(T_x, S_x, w) \rightarrow T_E$ – функція шифрування

$F_D(T_E, E, w) = F_D(F_E(T_x, S_x, w), w) \rightarrow T_x$ – функція дешифрування

Другий науковий результат

ОСНОВНІ ПОЛОЖЕННЯ МЕТОДУ

Метод орієнтований на підвищення ефективності захисту інформаційних ресурсів комп'ютерних систем за рахунок зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Метод базується на послідовному застосування до вихідного тексту повідомлень оптимального нерівномірного кодування Хаффмена і шифру зсуву.

Шифри зсуву застосовуються як елементарний варіант криптографічного захисту даних з мінімальною криптостійкістю для дослідження можливостей підвищення ефективності алгоритмів криптографічного шифрування застосуванням методів оптимального кодування. Шифрування і дешифрування виконується за найпростішим класичним алгоритмом реалізації шифру зсуву (шифр Цезаря).

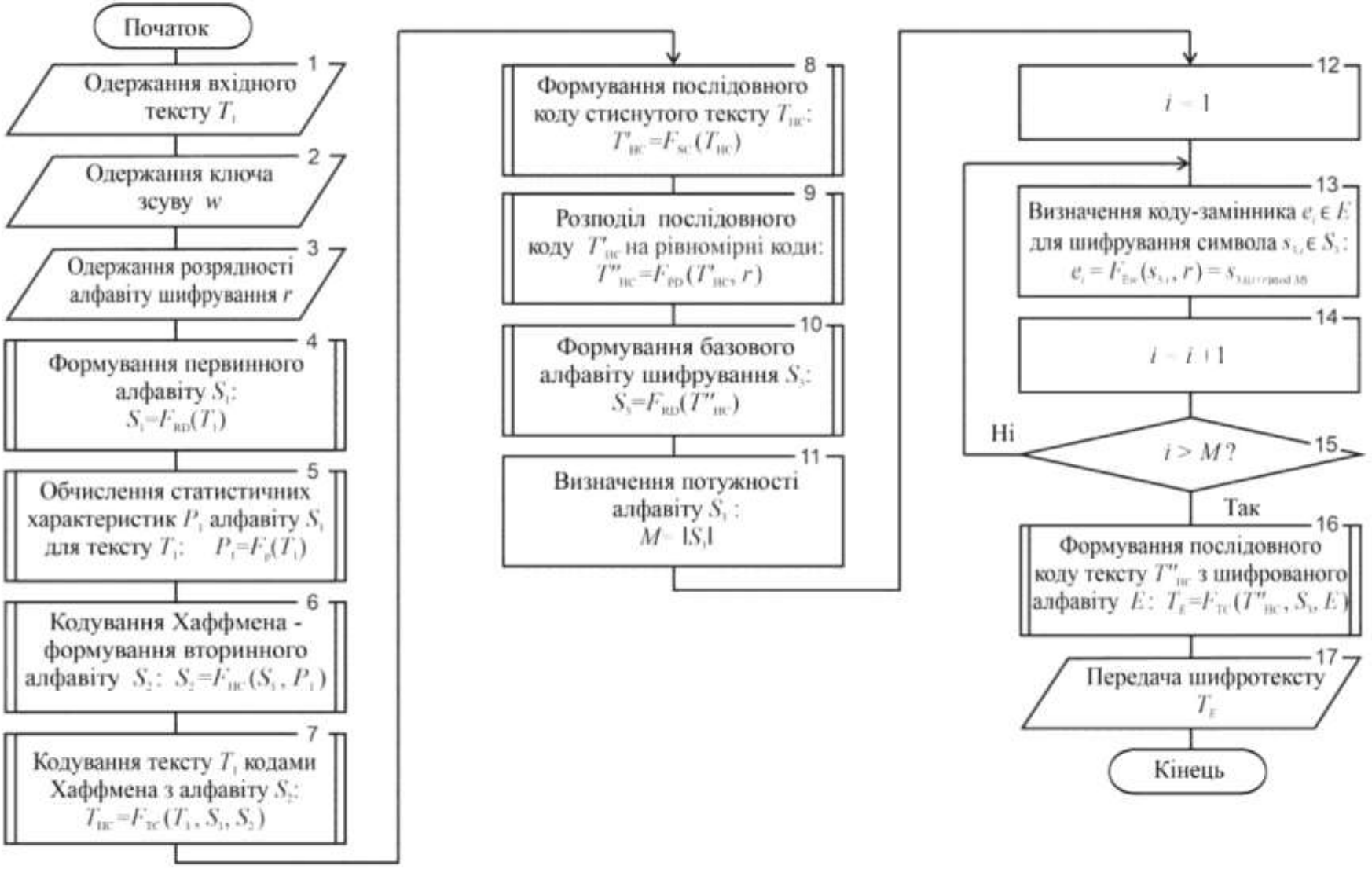
Застосування оптимального нерівномірного кодування Хаффмена забезпечує:

- зменшення розміру бінарного коду вхідного тексту, що підлягає шифруванню;
- зміну потужності (кількості елементів) алфавіту шифрування;
- можливість варіювання розрядністю рівномірних кодів алфавіту шифрування;
- зміну статистичних ймовірностей появи символів вхідного в шифрування алфавіту.

Оптимальний код вхідного тексту представляється послідовним кодом, до якого застосовується операція розбиття на кодові комбінації рівної довжини (фіксованої розрядності), які є вхідними в алгоритм шифрування.

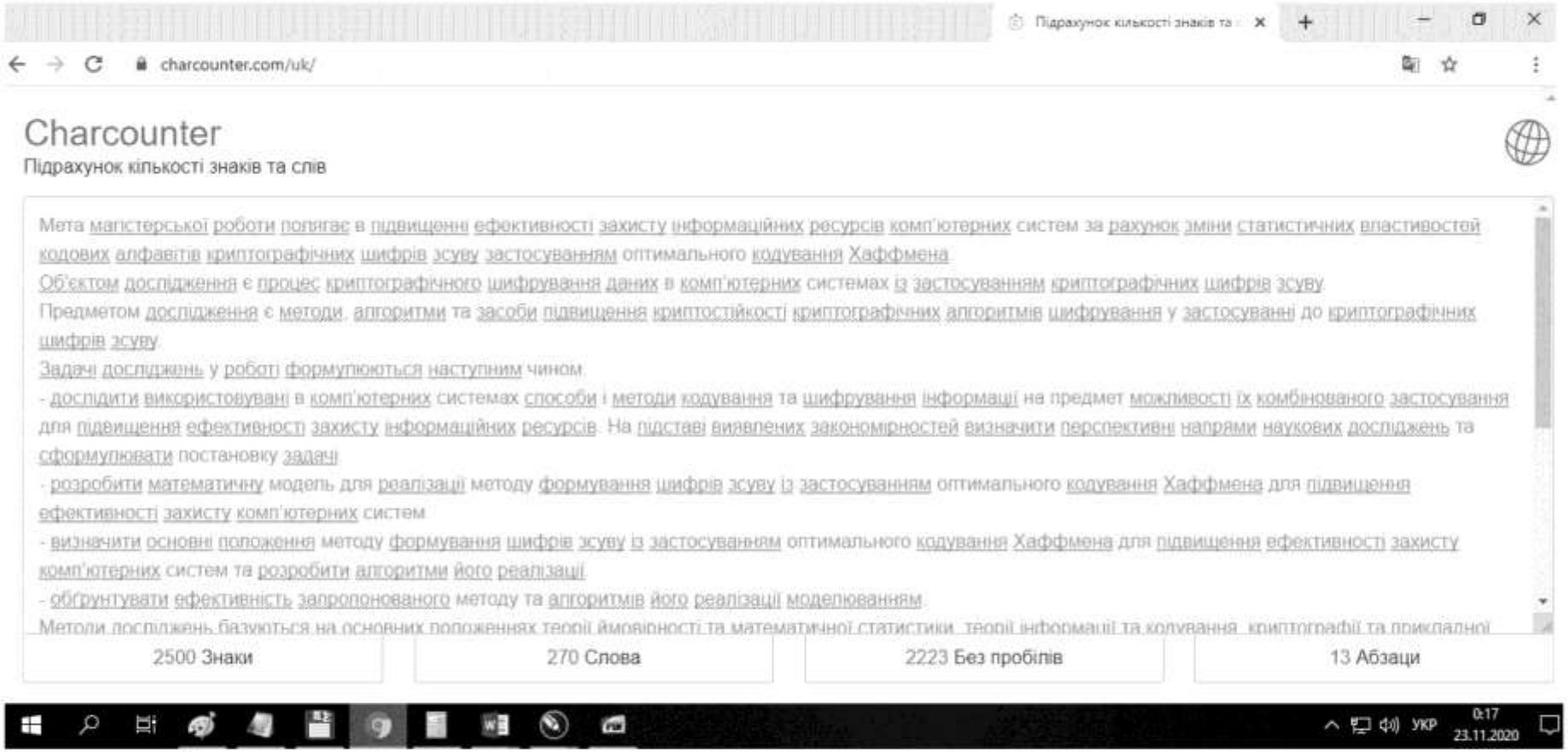
Другий науковий результат

АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ МЕТОДУ



АПРОБАЦІЯ МЕТОДУ – ВХІДНІ ДАНІ

Загальна оцінка властивостей обраного для експерименту вхідного тексту



Мова – українська

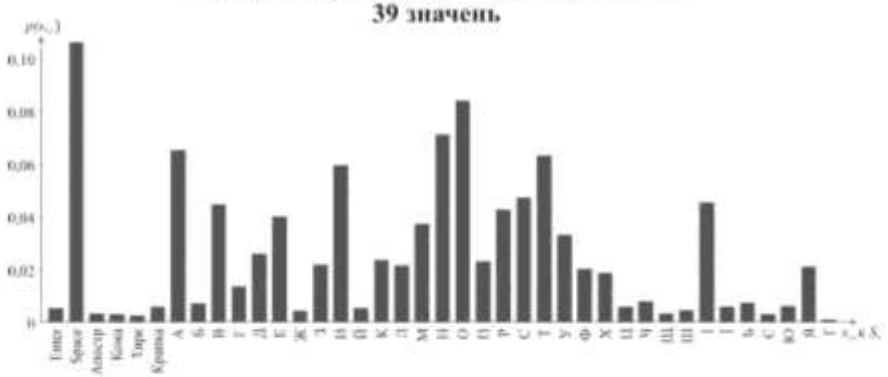
Походження – основні визначення дипломної роботи магістра (зі вступу)

Загальний розмір тексту – 2500 знаків із спеціальними включно

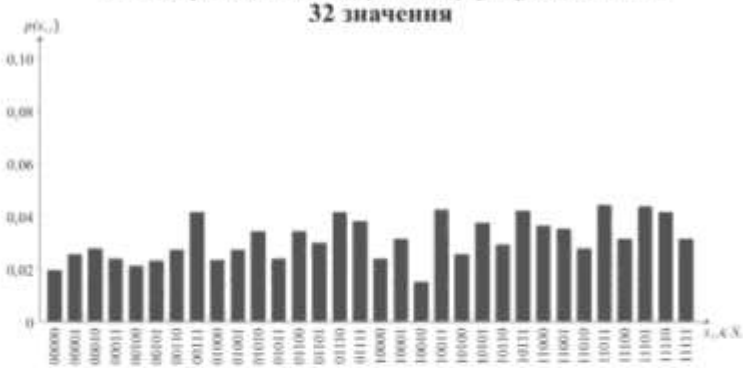
Кількість символів алфавіту – 39

АПРОБАЦІЯ МЕТОДУ - РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

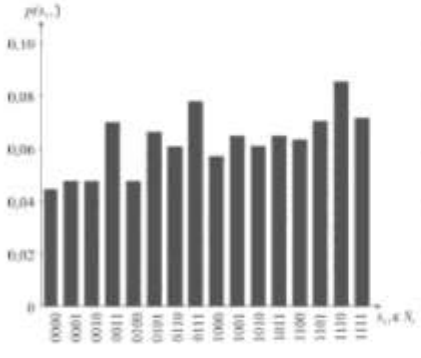
Кодовий алфавіт вхідного тексту (ASCII)
39 значень



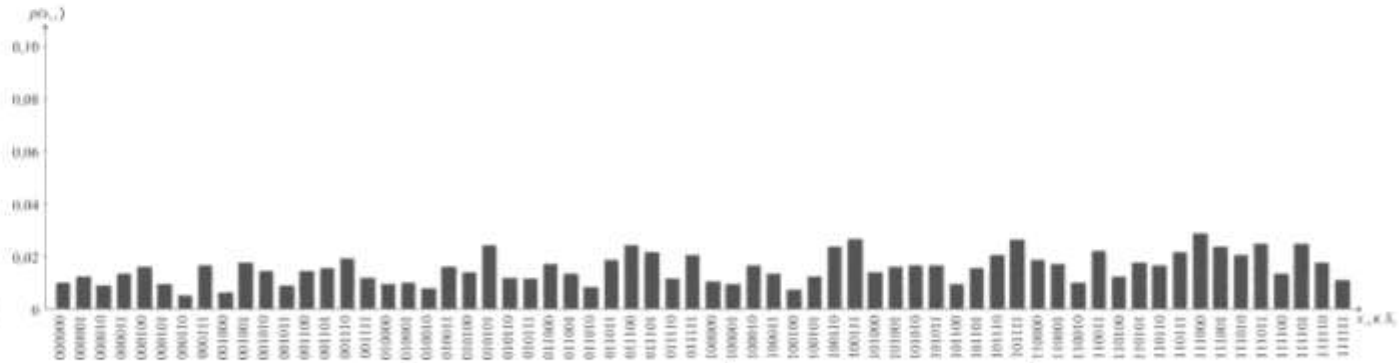
Рівномірні коди виходу методу розрядності *r*=5
32 значення



Рівномірні коди виходу методу розрядності *r*=4 - 16 значень



Рівномірні коди виходу методу розрядності *r*=6
64 значення



Застосування оптимального кодування для попередньої підготовки даних до шифрування дає наступні результати:

- змінюється склад кодового алфавіту шифрування (кількість кодів);
- змінюються статистичні властивості кодового алфавіту;
- забезпечується можливість довільної зміни розрядності коду алфавіту у відповідності до потреб алгоритму шифрування;
- незалежно від потужності вхідного алфавіту, потужність алфавіту виходу дорівнює або максимально наближена до 2^r.

ВИСНОВКИ

<u>У результаті виконаного дослідження можна зробити висновки:</u>

В роботі комплексно розв'язано задачу розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, а саме:

- досліджено використовувані в комп'ютерних системах способи і методи кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів. На підставі виявлених закономірностей визначено перспективні напрями наукових досліджень і сформулювано постановку задачі;
- розроблено математичну і структурно-логічну модель методу;
- визначено основні положення методу та розроблено алгоритми його реалізації;
- обґрунтовано ефективність запропонованого методу і алгоритмів його реалізації моделюванням.

Застосування оптимального кодування для попередньої підготовки даних до шифрування згідно із запропонованим методом дозволяє підвищити криптостійкість алгоритмів шифрування за рахунок зміни потужності і статистичних властивостей кодового алфавіту шифрування, а також забезпечує підвищену гнучкість щодо адаптації розрядності шифрокодів до потреб алгоритму шифрування.

ДОДАТОК В

(обов’язковий)

Дані і результати застосування алгоритму оптимального кодування Хаффмена

Вхідний текст (оптимізований фрагмент вступу кваліфікаційної роботи):

Мета магістерської роботи полягає в підвищенні ефективності захисту інформаційних ресурсів комп’ютерних систем за рахунок зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву застосуванням оптимального кодування Хаффмена.

Об'єктом дослідження є процес криптографічного шифрування даних в комп’ютерних системах із застосуванням криптографічних шифрів зсуву.

Предметом дослідження є методи, алгоритми та засоби підвищення криптостійкості криптографічних алгоритмів шифрування у застосуванні до криптографічних шифрів зсуву.

Задачі досліджень у роботі формулюються наступним чином.

- дослідити використовувані в комп’ютерних системах способи і методи кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів. На підставі виявлених закономірностей визначити перспективні напрями наукових досліджень та сформулювати постановку задачі.
- розробити математичну модель для реалізації методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.
- визначити основні положення методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем та розробити алгоритми його реалізації.
- обґрунтувати ефективність запропонованого методу та алгоритмів його реалізації моделюванням.

Методи досліджень базуються на основних положеннях теорії ймовірності та математичної статистики, теорії інформації та кодування, криптографії та прикладної криптології.

Наукова новизна отриманих результатів.

- Розроблено нову математичну модель, яка враховує особливості сумісного використання методів оптимального кодування та шифрів зсуву для підвищення криптостійкості систем захисту інформаційних ресурсів комп’ютерних систем.
- Запропоновано спосіб застосування методу оптимального кодування Хаффмена для зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву, який став основою для розробки методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп’ютерних систем.

Практична цінність розробленого методу полягає у обґрунтуванні можливості підвищення ефективності криптографічного захисту інформаційних ресурсів комп’ютерних систем застосуванням методів оптимального кодування перед реалізацією криптографічних методів захисту, а також у розробці алгоритмів реалізації методу.

ASCII-кодування вхідного тексту (текст *T*1):

10001100 10000101 10010010 10000000 00100000 10001100 10000000 10000011 10011010 10010001 10010010 10000101
10010000 10010001 10011100 10001010 10001110 10011011 00100000 10010000 10001110 10000001 10001110 10010010
10001000 00100000 10001111 10001110 10001011 10011111 10000011 10000000 10011101 00100000 10000010 00100000
10001111 10011010 10000100 10000010 10001000 10011000 10000101 10001101 10001101 10011010 00100000 10000101
10010100 10000101 10001010 10010010 10001000 10000010 10001101 10001110 10010001 10010010 10011010 00100000
10000111 10000000 10010101 10001000 10010001 10010010 10010011 00100000 10011010 10001101 10010100 10001110
10010000 10001100 10000000 10010110 10011010 10001001 10001101 10001000 10010101 00100000 10010000 10000101
10010001 10010011 10010000 10010001 10011010 10000010 00100000 10001010 10001110 10001100 10001111 00100110
10011110 10010010 10000101 10010000 10001101 10001000 10010101 00100000 10010001 10001000 10010001 10010010
10000101 10001100 00100000 10000111 10000000 00100000 10010000 10000000 10010101 10010011 10001101 10001110
10001010 00100000 10000111 10001100 10011010 10001101 10001000 00100000 10010001 10010010 10000000 10010010
10001000 10010001 10010010 10001000 10010111 10001101 10001000 10010101 00100000 10000010 10001011 10000000
10010001 10010010 10001000 10000010 10001110 10010001 10010010 10000101 10001001 00100000 10001010 10001110
10000100 10001110 10000010 10001000 10010101 00100000 10000000 10001011 10010100 10000000 10000010 10011010
10010010 10011010 10000010 00100000 10001010 10010000 10001000 10001111 10010010 10001110 10000011 10010000
10000000 10010100 10011010 10010111 10001101 10001000 10010101 00100000 10011001 10001000 10010100 10010000
10011010 10000010 00100000 10000111 10010001 10010011 10000010 10010011 00100000 10000111 10000000 10010001
10010010 10001110 10010001 10010011 10000010 10000000 10001101 10001101 10011111 10001100 00100000 10001110

10001111 10010010 10001000 10001100 10000000 10001011 10011100 10001101 10001110 10000011 10001110 00100000
10001010 10001110 10000100 10010011 10000010 10000000 10001101 10001101 10011111 00100000 10010101 10000000
10010100 10010100 10001100 10000101 10001101 10000000 00101110 00001010 10001110 10000001 00100110 10011101
10001010 10010010 10001110 10001100 00100000 10000100 10001110 10010001 10001011 10011010 10000100 10000110
10000101 10001101 10001101 10011111 00100000 10011101 00100000 10001111 10010000 10001110 10010110 10000101
10010001 00100000 10001010 10010000 10001000 10001111 10010010 10001110 10000011 10010000 10000000 10010100
10011010 10010111 10001101 10001110 10000011 10001110 00100000 10011001 10001000 10010100 10010000 10010011
10000010 10000000 10001101 10001101 10011111 00100000 10000100 10000000 10001101 10001000 10010101 00100000
10000010 00100000 10001010 10001110 10001100 10001111 00100110 10011110 10010010 10000101 10010000 10001101
10001000 10010101 00100000 10010001 10001000 10010001 10010010 10000101 10001100 10000000 10010101 00100000
10011010 10000111 00100000 10000111 10000000 10010001 10010010 10001110 10010001 10010011 10000010 10000000
10001101 10001101 10011111 10001100 00100000 10001010 10010000 10001000 10001111 10010010 10001110 10000011
10010000 10000000 10010100 10011010 10010111 10001101 10001000 10010101 00100000 10011001 10001000 10010100
10010000 10011010 10000010 00100000 10000111 10010001 10010011 10000010 10010011 00101110 00001010 10001111
10010000 10000101 10000100 10001100 10000101 10010010 10001110 10001100 00100000 10000100 10001110 10010001
10001011 10011010 10000100 10000110 10000101 10001101 10001101 10011111 00100000 10011101 00100000 10001100
10000101 10010010 10001110 10000100 10001000 00100110 00100000 10000000 10001011 10000011 10001110 10010000
10001000 10010010 10001100 10001000 00100000 10010010 10000000 00100000 10000111 10000000 10010001 10001110
10000001 10001000 00100000 10001111 10011010 10000100 10000010 10001000 10011000 10000101 10001101 10001101
10011111 00100000 10001010 10010000 10001000 10001111 10010010 10001110 10010001 10010010 10011010 10001001
10001010 10001110 10010001 10010010 10011010 00100000 10001010 10010000 10001000 10001111 10010010 10001110
10000011 10010000 10000000 10010100 10011010 10010111 10001101 10001000 10010101 00100000 10000000 10001011
10000011 10001110 10010000 10001000 10010010 10001100 10011010 10000010 00100000 10011001 10001000 10010100
10010000 10010011 10000010 10000000 10001101 10001101 10011111 00100000 10010011 00100000 10000111 10000000
10010001 10010010 10001110 10010001 10010011 10000010 10000000 10001101 10001101 10011010 00100000 10000100
10001110 00100000 10001010 10010000 10001000 10001111 10010010 10001110 10000011 10010000 10000000 10010100
10011010 10010111 10001101 10001000 10010101 00100000 10011001 10001000 10010100 10010000 10011010 10000010
00100000 10000111 10010001 10010011 10000010 10010011 00101110 00001010 10000111 10000000 10000100 10000000
10010111 10011010 00100000 10000100 10001110 10010001 10001011 10011010 10000100 10000110 10000101 10001101
10011100 00100000 10010011 00100000 10010000 10001110 10000001 10001110 10010010 10011010 00100000 10010100
10001110 10010000 10001100 10010011 10001011 10011110 10011110 10010010 10011100 10010001 10011111 00100000
10001101 10000000 10010001 10010010 10010011 10001111 10001101 10001000 10001100 00100000 10010111 10001000
10001101 10001110 10001100 00101110 00001010 00101110 00100000 10000100 10001110 10010001 10001011 10011010
10000100 10001000 10010010 10001000 00100000 10000010 10001000 10001010 10001110 10010000 10001000 10010001
10010010 10001110 10000010 10010011 10000010 10000000 10001101 10011010 00100000 10000010 00100000 10001010
10001110 10001100 10001111 00100110 10011110 10010010 10000101 10010000 10001101 10001000 10010101 00100000
10010001 10001000 10010001 10010010 10000101 10001100 10000000 10010101 00100000 10010001 10001111 10001110
10010001 10001110 10000001 10001000 00100000 10011010 00100000 10001100 10000101 10010010 10001110 10000100
10001000 00100000 10001010 10001110 10000100 10010011 10000010 10000000 10001101 10001101 10011111 00100000
10010010 10000000 00100000 10011001 10001000 10010100 10010000 10010011 10000010 10000000 10001101 10001101
10011111 00100000 10011010 10001101 10010100 10001110 10010000 10001100 10000000 10010110 10011010 10011011
00100000 10001101 10000000 00100000 10001111 10010000 10000101 10000100 10001100 10000101 10010010 00100000
10001100 10001110 10000110 10001011 10001000 10000010 10001110 10010001 10010010 10011010 00100000 10011011
10010101 00100000 10001010 10001110 10001100 10000001 10011010 10001101 10001110 10000010 10000000 10001101
10001110 10000011 10001110 00100000 10000111 10000000 10010001 10010010 10001110 10010001 10010011 10000010
10000000 10001101 10001101 10011111 00100000 10000100 10001011 10011111 00100000 10001111 10011010 10000100
10000010 10001000 10011000 10000101 10001101 10001101 10011111 00100000 10000101 10010100 10000101 10001010
10010010 10001000 10000010 10001101 10001110 10010001 10010010 10011010 00100000 10000111 10000000 10010101
10001000 10010001 10010010 10010011 00100000 10011010 10001101 10010100 10001110 10010000 10001100 10000000
10010110 10011010 10001001 10001101 10001000 10010101 00100000 10010000 10000101 10010001 10010011 10010000
10010001 10011010 10000010 00101110 00100000 10001101 10000000 00100000 10001111 10011010 10000100 10010001
10010010 10000000 10000010 10011010 00100000 10000010 10001000 10011111 10000010 10001011 10000101 10001101
10001000 10010101 00100000 10000111 10000000 10001010 10001110 10001101 10001110 10001100 10011010 10010000
10001101 10001110 10010001 10010010 10000101 10001001 00100000 10000010 10001000 10000111 10001101 10000000
10010111 10001000 10010010 10001000 00100000 10001111 10000101 10010000 10010001 10001111 10000101 10001010
10010010 10001000 10000010 10001101 10011010 00100000 10001101 10000000 10001111 10010000 10011111 10001100
10001000 00100000 10001101 10000000 10010011 10001010 10001110 10000010 10001000 10010101 00100000 10000100
10001110 10010001 10001011 10011010 10000100 10000110 10000101 10001101 10011100 00100000 10010010 10000000
00100000 10010001 10010100 10001110 10010000 10001100 10010011 10001011 10011110 10000010 10000000 10010010
10001000 00100000 10001111 10001110 10010001 10010010 10000000 10001101 10001110 10000010 10001010 10010011
00100000 10000111 10000000 10000100 10000000 10010111 10011010 00101111 00001010 00101110 00100000 10010000
10001110 10000111 10010000 10001110 10000001 10001000 10010010 10001000 00100000 10001100 10000000 10010010
10000101 10001100 10000000 10010010 10001000 10010111 10001101 10010011 00100000 10001100 10001110 10000100
10000101 10001011 10011100 00100000 10000100 10001011 10011111 00100000 10010000 10000101 10000000 10001011
10011010 10000111 10000000 10010110 10011010 10011011 00100000 10001100 10000101 10010010 10001110 10000100
10010011 00100000 10010100 10001110 10010000 10001100 10010011 10000010 10000000 10001101 10001101 10011111
00100000 10011001 10001000 10010100 10010000 10011010 10000010 00100000 10000111 10010001 10010011 10000010
10010011 00100000 10011010 10000111 00100000 10000111 10000000 10010001 10010010 10001110 10010001 10010011
10000010 10000000 10001101 10001101 10011111 10001100 00100000 10001110 10001111 10010010 10001000 10001100
10000000 10001011 10011100 10001101 10001110 10000011 10001110 00100000 10001010 10001110 10000100 10010011
10000010 10000000 10001101 10001101 10011111 00100000 10010101 10000000 10010100 10010100 10001100 10000101
10001101 10000000 00100000 10000100 10001011 10011111 00100000 10001111 10011010 10000100 10000010 10001000
10011000 10000101 10001101 10001101 10011111 00100000 10000101 10010100 10000101 10001010 10010010 10001000
10000010 10001101 10001110 10010001 10010010 10011010 00100000 10000111 10000000 10010101 10001000 10010001
10010010 10010011 00100000 10001010 10001110 10001100 10001111 00100110 10011110 10010010 10000101 10010000
10001101 10001000 10010101 00100000 10010001 10001000 10010001 10010010 10000101 10001100 00101111 00001010
00101110 00100000 10000010 10001000 10000111 10001101 10000000 10010111 10001000 10010010 10001000 00100000
10001110 10010001 10001101 10001110 10000010 10001101 10011010 00100000 10001111 10001110 10001011 10001110
10000110 10000101 10001101 10001101 10011111 00100000 10001100 10000101 10010010 10001110 10000100 10010011
00100000 10010100 10001110 10010000 10001100 10010011 10000010 10000000 10001101 10001101 10011111 00100000

10011001 10001000 10010100 10010000 10011010 10000010 00100000 10000111 10010001 10010011 10000010 10010011
00100000 10011010 10000111 00100000 10000111 10000000 10010001 10010010 10001110 10010001 10010011 10000010
10000000 10001101 10001101 10011111 10001100 00100000 10001110 10001111 10010010 10001000 10001100 10000000
10001011 10011100 10001101 10001110 10000011 10001110 00100000 10001010 10001110 10000100 10010011 10000010
10000000 10001101 10001101 10011111 00100000 10010101 10000000 10010100 10010100 10001100 10000101 10001101
10000000 00100000 10000100 10001011 10011111 00100000 10001111 10011010 10000100 10000010 10001000 10011000
10000101 10001101 10001101 10011111 00100000 10000101 10010100 10000101 10001010 10010010 10001000 10000010
10001101 10001110 10010001 10010010 10011010 00100000 10000111 10000000 10010101 10001000 10010001 10010010
10010011 00100000 10001010 10001110 10001100 10001111 00100110 10011110 10010010 10000101 10010000 10001101
10001000 10010101 00100000 10010001 10001000 10010001 10010010 10000101 10001100 00100000 10010010 10000000
00100000 10010000 10001110 10000111 10010000 10001110 10000001 10001000 10010010 10001000 00100000 10000000
10001011 10000011 10001110 10010000 10001000 10010010 10001100 10001000 00100000 10001001 10001110 10000011
10001110 00100000 10010000 10000101 10000000 10001011 10011010 10000111 10000000 10010110 10011010 10011011
00101111 00001010 00101110 00100000 10001110 10000001 10100000 10010000 10010011 10001101 10010010 10010011
10000010 10000000 10010010 10001000 00100000 10000101 10010100 10000101 10001010 10010010 10001000 10000010
10001101 10011010 10010001 10010010 10011100 00100000 10000111 10000000 10001111 10010000 10001110 10001111
10001110 10001101 10001110 10000010 10000000 10001101 10001110 10000011 10001110 00100000 10001100 10000101
10010010 10001110 10000100 10010011 00100000 10010010 10000000 00100000 10000000 10001011 10000011 10001110
10010000 10001000 10010010 10001100 10011010 10000010 00100000 10001001 10001110 10000011 10001110 00100000
10010000 10000101 10000000 10001011 10011010 10000111 10000000 10010110 10011010 10011011 00100000 10001100
10001110 10000100 10000101 10001011 10011110 10000010 10000000 10001101 10001101 10011111 10001100 00101110
00001010 10001100 10000101 10010010 10001110 10000100 10001000 00100000 10000100 10001110 10010001 10001011
10011010 10000100 10000110 10000101 10001101 10011100 00100000 10000001 10000000 10000111 10010011 10011110
10010010 10011100 10010001 10011111 00100000 10001101 10000000 00100000 10001110 10010001 10001101 10001110
10000010 10001101 10001000 10010101 00100000 10001111 10001110 10001011 10001110 10000110 10000101 10001101
10001101 10011111 10010101 00100000 10010010 10000101 10001110 10010000 10011010 10011011 00100000 10001001
10001100 10001110 10000010 10011010 10010000 10001101 10001110 10010001 10010010 10011010 00100000 10010010
10000000 00100000 10001100 10000000 10010010 10000101 10001100 10000000 10010010 10001000 10010111 10001101
10001110 10011011 00100000 10010001 10010010 10000000 10010010 10001000 10010001 10010010 10001000 10001010
10001000 00100110 00100000 10010010 10000101 10001110 10010000 10011010 10011011 00100000 10011010 10001101
10010100 10001110 10010000 10001100 10000000 10010110 10011010 10011011 00100000 10010010 10000000 00100000
10001010 10001110 10000100 10010011 10000010 10000000 10001101 10001101 10011111 00100110 00100000 10001010
10010000 10001000 10001111 10010010 10001110 10000011 10010000 10000000 10010100 10011010 10011011 00100000
10010010 10000000 00100000 10001111 10010000 10001000 10001010 10001011 10000000 10000100 10001101 10001110
10011011 00100000 10001010 10010000 10001000 10001111 10010010 10001110 10001011 10001110 10000011 10011010
10011011 00101110 00001010 10001101 10000000 10010011 10001010 10001110 10000010 10000000 00100000 10001101
10001110 10000010 10001000 10000111 10001101 10000000 00100000 10001110 10010010 10010000 10001000 10001100
10000000 10001101 10001000 10010101 00100000 10010000 10000101 10000111 10010011 10001011 10011100 10010010
10000000 10010010 10011010 10000010 00101110 00001010 00101110 00100000 10010000 10001110 10000111 10010000
10001110 10000001 10001011 10000101 10001101 10001110 00100000 10001101 10001110 10000010 10010011 00100000
10001100 10000000 10010010 10000101 10001100 10000000 10010010 10001000 10010111 10001101 10010011 00100000
10001100 10001110 10000100 10000101 10001011 10011100 00100110 00100000 10011111 10001010 10000000 00100000
10000010 10010000 10000000 10010101 10001110 10000010 10010011 10011101 00100000 10001110 10010001 10001110
10000001 10001011 10001000 10000010 10001110 10010001 10010010 10011010 00100000 10010001 10010011 10001100
10011010 10010001 10001101 10001110 10000011 10001110 00100000 10000010 10001000 10001010 10001110 10010000
10001000 10010001 10010010 10000000 10001101 10001101 10011111 00100000 10001100 10000101 10010010 10001110
10000100 10011010 10000010 00100000 10001110 10001111 10010010 10001000 10001100 10000000 10001011 10011100
10001101 10001110 10000011 10001110 00100000 10001010 10001110 10000100 10010011 10000010 10000000 10001101
10001101 10011111 00100000 10010010 10000000 00100000 10011001 10001000 10010100 10010000 10011010 10000010
00100000 10000111 10010001 10010011 10000010 10010011 00100000 10000100 10001011 10011111 00100000 10001111
10011010 10000100 10000010 10001000 10011000 10000101 10001101 10001101 10011111 00100000 10001010 10010000
10001000 10001111 10010010 10001110 10010001 10010010 10011010 10001001 10001010 10001110 10010001 10010010
10011010 00100000 10010001 10001000 10010001 10010010 10000101 10001100 00100000 10000111 10000000 10010101
10001000 10010001 10010010 10010011 00100000 10011010 10001101 10010100 10001110 10010000 10001100 10000000
10010110 10011010 10001001 10001101 10001000 10010101 00100000 10010000 10000101 10010001 10010011 10010000
10010001 10011010 10000010 00100000 10001010 10001110 10001100 10001111 00100110 10011110 10010010 10000101
10010000 10001101 10001000 10010101 00100000 10010001 10001000 10010001 10010010 10000101 10001100 00101111
00001010 00101110 00100000 10000111 10000000 10001111 10010000 10001110 10001111 10001110 10001101 10001110
10000010 10000000 10001101 10001110 00100000 10010001 10001111 10001110 10010001 10011010 10000001 00100000
10000111 10000000 10010001 10010010 10001110 10010001 10010011 10000010 10000000 10001101 10001101 10011111
00100000 10001100 10000101 10010010 10001110 10000100 10010011 00100000 10001110 10001111 10010010 10001000
10001100 10000000 10001011 10011100 10001101 10001110 10000011 10001110 00100000 10001010 10001110 10000100
10010011 10000010 10000000 10001101 10001101 10011111 00100000 10010101 10000000 10010100 10010100 10001100
10000101 10001101 10000000 00100000 10000100 10001011 10011111 00100000 10000111 10001100 10011010 10001101
10001000 00100000 10010001 10010010 10000000 10010010 10001000 10010001 10010010 10001000 10010111 10001101
10001000 10010101 00100000 10000010 10001011 10000000 10010001 10010010 10001000 10000010 10001110 10010001
10010010 10000101 10001001 00100000 10001010 10001110 10000100 10001110 10000010 10001000 10010101 00100000
10000000 10001011 10010100 10000000 10000010 10011010 10010010 10011010 10000010 00100000 10001010 10010000
10001000 10001111 10010010 10001110 10000011 10010000 10000000 10010100 10011010 10010111 10001101 10001000
10010101 00100000 10011001 10001000 10010100 10010000 10011010 10000010 00100000 10000111 10010001 10010011
10000010 10010011 00100110 00100000 10011111 10001010 10001000 10001001 00100000 10010001 10010010 10000000
10000010 00100000 10001110 10010001 10001101 10001110 10000010 10001110 10011110 00100000 10000100 10001011
10011111 00100000 10010000 10001110 10000111 10010000 10001110 10000001 10001010 10001000 00100000 10001100
10000101 10010010 10001110 10000100 10010011 00100000 10010100 10001110 10010000 10001100 10010011 10000010
10000000 10001101 10001101 10011111 00100000 10011001 10001000 10010100 10010000 10011010 10000010 00100000
10000111 10010001 10010011 10000010 10010011 00100000 10011010 10000111 00100000 10000111 10000000 10010001
10010010 10001110 10010001 10010011 10000010 10000000 10001101 10001101 10011111 10001100 00100000 10001110
10001111 10010010 10001000 10001100 10000000 10001011 10011100 10001101 10001110 10000011 10001110 00100000
10001010 10001110 10000100 10010011 10000010 10000000 10001101 10001101 10011111 00100000 10010101 10000000
10010100 10010100 10001100 10000101 10001101 10000000 00100000 10000100 10001011 10011111 00100000 10001111

10011010 10000100 10000010 10001000 10011000 10000101 10001101 10001101 10011111 00100000 10000101 10010100
10000101 10001010 10010010 10001000 10000010 10001101 10001110 10010001 10010010 10011010 00100000 10000111
10000000 10010101 10001000 10010001 10010010 10010011 00100000 10001010 10001110 10001100 10001111 00100110
10011110 10010010 10000101 10010000 10001101 10001000 10010101 00100000 10010001 10001000 10010001 10010010
10000101 10001100 00101110 00001010 10001111 10010000 10000000 10001010 10010010 10001000 10010111 10001101
10000000 00100000 10010110 10011010 10001101 10001101 10011010 10010001 10010010 10011100 00100000 10010000
10001110 10000111 10010000 10001110 10000001 10001011 10000101 10001101 10001110 10000011 10001110 00100000
10001100 10000101 10010010 10001110 10000100 10010011 00100000 10001111 10001110 10001011 10011111 10000011
10000000 10011101 00100000 10010011 00100000 10001110 10000001 10100000 10010000 10010011 10001101 10010010
10010011 10000010 10000000 10001101 10001101 10011010 00100000 10001100 10001110 10000110 10001011 10001000
10000010 10001110 10010001 10010010 10011010 00100000 10001111 10011010 10000100 10000010 10001000 10011000
10000101 10001101 10001101 10011111 00100000 10000101 10010100 10000101 10001010 10010010 10001000 10000010
10001101 10001110 10010001 10010010 10011010 00100000 10001010 10010000 10001000 10001111 10010010 10001110
10000011 10010000 10000000 10010100 10011010 10010111 10001101 10001110 10000011 10001110 00100000 10000111
10000000 10010101 10001000 10010001 10010010 10010011 00100000 10011010 10001101 10010100 10001110 10010000
10001100 10000000 10010110 10011010 10001001 10001101 10001000 10010101 00100000 10010000 10000101 10010001
10010011 10010000 10010001 10011010 10000010 00100000 10001010 10001110 10001100 10001111 00100110 10011110
10010010 10000101 10010000 10001101 10001000 10010101 00100000 10010001 10001000 10010001 10010010 10000101
10001100 00100000 10000111 10000000 10010001 10010010 10001110 10010001 10010011 10000010 10000000 10001101
10001101 10011111 10001100 00100000 10001100 10000101 10010010 10001110 10000100 10011010 10000010 00100000
10001110 10001111 10010010 10001000 10001100 10000000 10001011 10011100 10001101 10001110 10000011 10001110
00100000 10001010 10001110 10000100 10010011 10000010 10000000 10001101 10001101 10011111 00100000 00100000
10001111 10000101 10010000 10000101 10000100 00100000 10010000 10000101 10000000 10001011 10011010 10000111
10000000 10010110 10011010 10011101 10011110 00100000 10001010 10010000 10001000 10001111 10010010 10001110
10000011 10010000 10000000 10010100 10011010 10010111 10001101 10001000 10010101 00100000 10001100 10000101
10010010 10001110 10000100 10011010 10000010 00100000 10000111 10000000 10010101 10001000 10010001 10010010
10010011 00100110 00100000 10000000 00100000 10010010 10000000 10001010 10001110 10000110 00100000 10010011
00100000 10010000 10001110 10000111 10010000 10001110 10000001 10010110 10011010 00100000 10000000 10001011
10000011 10001110 10010000 10001000 10010010 10001100 10011010 10000010 00100000 10010000 10000101 10000000
10001011 10011010 10000111 10000000 10010110 10011010 10011011 00100000 10001100 10000101 10010010 10001110
10000100 10010011 00101110 00001010

<u>Оптимальне нерівномірне кодування Хаффмена:</u>

Службова інформація для декодування:

– Маска ОНК:

00001 00001 0001 00001 0000001 0000001 0000001 000000001 0000000001 0000000001 00000001 0001
001 00001 000001 0000001 0000001 0001 0001 0001 0000001 00001 00000001 00000001 000001 0001 00001
000001 000001 00001 000000001 000000001 00000001 00000001 00000001 000001 0001 00001 0000111

– Коди ОНК (алфавіт S_2):

00000 00001 0001 00100 0010100 0010101 0010110 001011100 0010111010 0010111011 00101111 0011
010 01100 011010 0110110 0110111 0111 1000 1001 1010100 10100 10101010 10101011 101011 1011 11000
110100 110101 11001 110110000 110110001 11011001 11011010 11011011 110111 1110 11110 11111

– Рівномірні коди ASCII (алфавіт S_1):

10000111 10001111 10011010 10001010 00101111 10011011 10011110 00100110 10100000 00101110 10011001
10010001 00100000 10000100 10000011 10000001 10011100 10001000 10010010 10000000 10010111 10010011
10000110 10011001 10010101 10001101 10001100 10010100 10011111 10000101 00101100 10011101 00001010
10001001 10010110 10001011 10001110 10010000 10000010

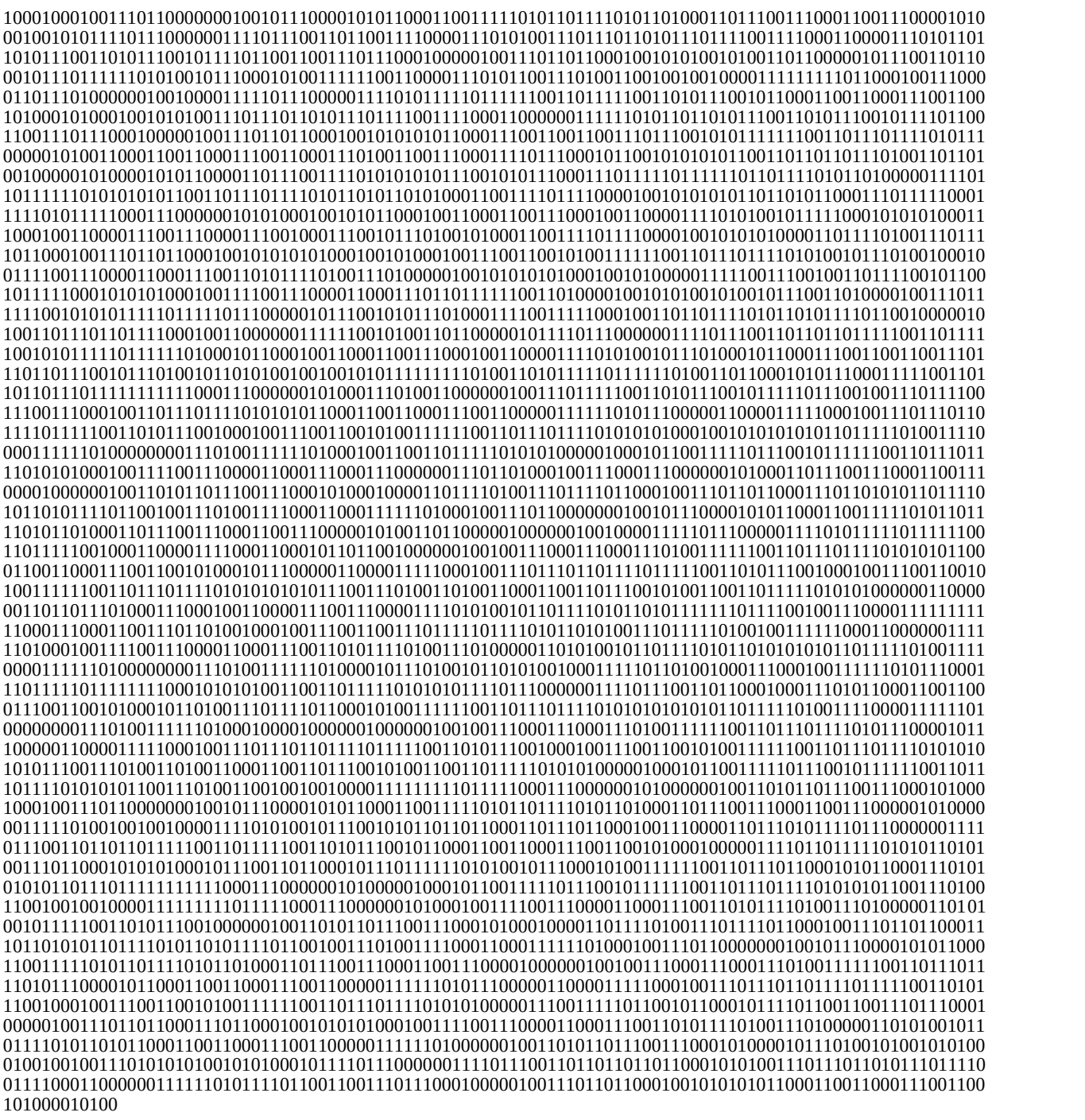
Закодований оптимальним нерівномірним кодом текст (T_{CH}):

11000 11001 1000 1001 010 11000 1001 011010 0001 0011 1000 11001 11110 0011 0110111 00100 1110 0010101 010 11110
1110 0110110 1110 1000 0111 010 00001 1110 110111 110101 011010 1001 110110001 010 11111 010 00001 0001 01100
11111 0111 00101111 11001 1011 1011 0001 010 11001 110100 11001 00100 1000 0111 11111 1011 1110 0011 1000 0001
010 00000 1001 101011 0111 0011 1000 10100 010 0001 1011 110100 1110 11110 11000 1001 11011011 0001 11011010 1011
0111 101011 010 11110 11001 0011 10100 11110 0011 0001 11111 010 00100 1110 11000 00001 001011100 0010101 1000
11001 11110 1011 0111 101011 010 0011 0111 0011 1000 11001 11000 010 00000 1001 010 11110 1001 101011 10100 1011
1110 00100 010 00000 11000 0001 1011 0111 010 0011 1000 1001 1000 0111 0011 1000 0111 1010100 1011 0111 101011 010
11111 110111 1001 0011 1000 0111 11111 1110 0011 1000 11001 11011010 010 00100 1110 01100 1110 11111 0111 101011
010 1001 110111 110100 1001 11111 0001 1000 0001 11111 010 00100 11110 0111 00001 1000 1110 011010 11110 1001
110100 0001 1010100 1011 0111 101011 010 10101011 0111 110100 11110 0001 11111 010 00000 0011 10100 11111 10100
010 00000 1001 0011 1000 1110 0011 10100 11111 1001 1011 1011 110101 11000 010 1110 00001 1000 0111 11000 1001
110111 0110111 1011 1110 011010 1110 010 00100 1110 01100 10100 11111 1001 1011 1011 110101 010 101011 1001
110100 110100 11000 11001 1011 1001 0010100 1110 0110110 001011100 110110001 00100 1000 1110 11000 010 01100
1110 0011 110111 0001 01100 10101010 11001 1011 1011 110101 010 110110001 010 00001 11110 1110 11011011 11001
0011 010 00100 11110 0111 00001 1000 1110 011010 11110 1001 110100 0001 1010100 1011 1110 011010 1110 010
10101011 0111 110100 11110 10100 11111 1001 1011 1011 110101 010 01100 1001 1011 0111 101011 010 11111 010 00100
1110 11000 00001 001011100 0010101 1000 11001 11110 1011 0111 101011 010 0011 0111 0011 1000 11001 11000 1001
101011 010 0001 00000 010 00000 1001 0011 1000 1110 0011 10100 11111 1001 1011 1011 110101 11000 010 00100 11110
0111 00001 1000 1110 011010 11110 1001 110100 0001 1010100 1011 0111 101011 010 10101011 0111 110100 11110 0001
11111 010 00000 0011 10100 11111 10100 0010100 00001 11110 11001 01100 11000 11001 1000 1110 11000 010 01100 1110
0011 110111 0001 01100 10101010 11001 1011 1011 110101 010 110110001 010 11000 11001 1000 1110 01100 0111
0010111010 010 1001 110111 011010 1110 11110 0111 1000 11000 0111 010 1000 1001 010 00000 1001 0011 1110 0110110
0111 010 00001 0001 01100 11111 0111 00101111 11001 1011 1011 110101 010 00100 11110 0111 00001 1000 1110 0011
1000 0001 11011010 00100 1110 0011 1000 0001 010 00100 11110 0111 00001 1000 1110 011010 11110 1001 110100 0001
1010100 1011 0111 101011 010 1001 110111 011010 1110 11110 0111 1000 11000 0001 11111 010 10101011 0111 110100

11110 10100 11111 1001 1011 1011 110101 010 10100 010 00000 1001 0011 1000 1110 0011 10100 11111 1001 1011 1011
0001 010 01100 1110 010 00100 11110 0111 00001 1000 1110 011010 11110 1001 110100 0001 1010100 1011 0111 101011
010 10101011 0111 110100 11110 0001 11111 010 00000 0011 10100 11111 10100 0010100 00000 1001 01100 1001 1010100
0001 010 01100 1110 0011 110111 0001 01100 10101010 11001 1011 0110111 010 10100 010 11110 1110 0110110 1110 1000
0001 010 110100 1110 11110 11000 10100 110111 0010101 0010101 1000 0110111 0011 110101 010 1011 1001 0011 1000
10100 00001 1011 0111 11000 010 010 1010100 0111 1011 1110 11000 0010100 110110000 010 01100 1110 0011 110111
0001 01100 0111 1000 0111 010 11111 0111 00100 1110 11110 0111 0011 1000 1110 11111 10100 11111 1001 1011 0001 010
11111 010 00100 1110 11000 00001 001011100 0010101 1000 11001 11110 1011 0111 101011 010 0011 0111 0011 1000
11001 11000 1001 101011 010 0011 00001 1110 0011 1110 0110110 0111 010 0001 010 11000 11001 1000 1110 01100 0111
010 00100 1110 01100 10100 11111 1001 1011 1011 110101 010 1000 1001 010 10101011 0111 110100 11110 10100 11111
1001 1011 1011 110101 010 0001 1011 110100 1110 11110 11000 1001 11011011 0001 0010101 010 1011 1001 010 00001
11110 11001 01100 11000 11001 1000 010 11000 1110 10101010 110111 0111 11111 1110 0011 1000 0001 010 0010101
101011 010 00100 1110 11000 0110110 0001 1011 1110 11111 1001 1011 1110 011010 1110 010 00000 1001 0011 1000 1110
0011 10100 11111 1001 1011 1011 110101 010 01100 110111 110101 010 00001 0001 01100 11111 0111 00101111 11001
1011 1011 110101 010 11001 110100 11001 00100 1000 0111 11111 1011 1110 0011 1000 0001 010 00000 1001 101011 0111
0011 1000 10100 010 0001 1011 110100 1110 11110 11000 1001 11011011 0001 11011010 1011 0111 101011 010 11110
11001 0011 10100 11110 0011 0001 11111 0010100 010 1011 1001 010 00001 0001 01100 0011 1000 1001 11111 0001 010
11111 0111 110101 11111 110111 11001 1011 0111 101011 010 00000 1001 00100 1110 1011 1110 11000 0001 11110 1011
1110 0011 1000 11001 11011010 010 11111 0111 00000 1011 1001 1010100 0111 1000 0111 010 00001 11001 11110 0011
00001 11001 00100 1000 0111 11111 1011 0001 010 1011 1001 00001 11110 110101 11000 0111 010 1011 1001 10100 00100
1110 11111 0111 101011 010 01100 1110 0011 110111 0001 01100 10101010 11001 1011 0110111 010 1000 1001 010 0011
110100 1110 11110 11000 10100 110111 0010101 11111 1001 1000 0111 010 00001 1110 0011 1000 1001 1011 1110 11111
00100 10100 010 00000 1001 01100 1001 1010100 0001 0010100 110110000 010 11110 1110 00000 11110 1110 0110110
0111 1000 0111 010 11000 1001 1000 11001 11000 1001 1000 0111 1010100 1011 10100 010 11000 1110 01100 11001
110111 0110111 010 01100 110111 110101 010 11110 11001 1001 110111 0001 00000 1001 11011011 0001 0010101 010
11000 11001 1000 1110 01100 10100 010 110100 1110 11110 11000 10100 11111 1001 1011 1011 110101 010 10101011 0111
110100 11110 0001 11111 010 00000 0011 10100 11111 10100 010 0001 00000 010 00000 1001 0011 1000 1110 0011 10100
11111 1001 1011 1011 110101 11000 010 1110 00001 1000 0111 11000 1001 110111 0110111 1011 1110 011010 1110 010
00100 1110 01100 10100 11111 1001 1011 1011 110101 010 101011 1001 110100 110100 11000 11001 1011 1001 010 01100
110111 110101 010 00001 0001 01100 11111 0111 00101111 11001 1011 1011 110101 010 11001 110100 11001 00100 1000
0111 11111 1011 1110 0011 1000 0001 010 00000 1001 101011 0111 0011 1000 10100 010 00100 1110 11000 00001
001011100 0010101 1000 11001 11110 1011 0111 101011 010 0011 0111 0011 1000 11001 11000 0010100 110110000 010
11111 0111 00000 1011 1001 1010100 0111 1000 0111 010 1110 0011 1011 1110 11111 1011 0001 010 00001 1110 110111
1110 10101010 11001 1011 1011 110101 010 10101011 0111 110100 11110 0001 11111 010 00000 0011 10100 11111 10100 010 0001
00000 010 00000 1001 0011 1000 1110 0011 10100 11111 1001 1011 1011 110101 11000 010 1110 00001 1000 0111 11000
1001 110111 0110111 1011 1110 011010 1110 010 00100 1110 01100 10100 11111 1001 1011 1011 110101 010 101011 1001
110100 110100 11000 11001 1011 1001 010 01100 110111 110101 010 00001 0001 01100 11111 0111 00101111 11001 1011
1011 110101 010 11001 110100 11001 00100 1000 0111 11111 1011 1110 0011 1000 0001 010 00000 1001 101011 0111 0011
1000 10100 010 00100 1110 11000 00001 001011100 0010101 1000 11001 11110 1011 0111 101011 010 0011 0111 0011 1000
11001 11000 010 1000 1001 010 11110 1110 00000 11110 1110 0110110 0111 1000 0111 010 1001 110111 011010 1110
11110 0111 1000 11000 0111 010 11011010 1110 011010 1110 010 11110 11001 1001 110111 0001 00000 1001 11011011
0001 0010101 0010100 110110000 010 1110 0110110 0010111011 11110 10100 1011 1000 10100 11111 1001 1000 0111 010
11001 110100 11001 00100 1000 0111 11111 1011 0001 0011 1000 0110111 010 00000 1001 00001 11110 1110 00001 1110
1011 1110 11111 1001 1011 1110 011010 1110 010 11000 11001 1000 1110 01100 10100 010 1000 1001 010 1001 110111
011010 1110 11110 0111 1000 11000 0001 11111 010 11011010 1110 011010 1110 010 11110 11001 1001 110111 0001 00000
1001 11011011 0001 0010101 010 11000 1110 01100 11001 110111 0010101 11111 1001 1011 1011 110101 11000 0010100
11000 11001 1000 1110 01100 0111 010 01100 1110 0011 110111 0001 01100 10101010 11001 1011 0110111 010 0110110
1001 00000 10100 0010101 1000 0110111 0011 110101 010 1011 1001 010 1110 0011 1011 1110 11111 1011 0111 101011
010 00001 1110 110111 1110 10101010 11001 1011 1011 110101 101011 010 1000 11001 1110 11110 0001 0010101 010
11011010 11000 1110 11111 0001 11110 1011 1110 0011 1000 0001 010 1000 1001 010 11000 1001 1000 11001 11000 1001
1000 0111 1010100 1011 1110 0010101 010 0011 1000 1001 1000 0111 0011 1000 0111 00100 0111 0010111010 010 1000
11001 1110 11110 0001 0010101 010 0001 1011 110100 1110 11110 11000 1001 11011011 0001 0010101 010 1000 1001 010
00100 1110 01100 10100 11111 1001 1011 1011 110101 0010111010 010 00100 11110 0111 00001 1000 1110 011010 11110
1001 110100 0001 0010101 010 1000 1001 010 00001 11110 0111 00100 110111 1001 01100 1011 1110 0010101 010 00100
11110 0111 00001 1000 1110 110111 1110 011010 0001 0010101 0010100 1011 1001 10100 00100 1110 11111 1001 010 1011
1110 11111 0111 00000 1011 1001 010 1110 1000 11110 0111 11000 1001 1011 0111 101011 010 11110 11001 00000 10100
110111 0110111 1000 1001 1000 0001 11111 0010100 110110000 010 11110 1110 00000 11110 1110 0110110 110111 11001
1011 1110 010 1011 1110 11111 10100 010 11000 1001 1000 11001 11000 1001 1000 0111 1010100 1011 10100 010 11000
1110 01100 11001 110111 0110111 0010111010 010 110101 00100 1001 010 11111 11110 1001 101011 1110 11111 10100
110110001 010 1110 0011 1110 0110110 110111 0111 11111 1110 0011 1000 0001 010 0011 10100 11000 0001 0011 1011
1110 011010 1110 010 11111 0111 00100 1110 11110 0111 0011 1000 1001 1011 1011 110101 010 11000 11001 1000 1110
01100 0001 11111 010 1110 00001 1000 0111 11000 1001 110111 0110111 1011 1110 011010 1110 010 00100 1110 01100
10100 11111 1001 1011 1011 110101 010 1000 1001 010 10101011 0111 110100 11110 0001 11111 010 00000 0011 10100
11111 10100 010 01100 110111 110101 010 00001 0001 01100 11111 0111 00101111 11001 1011 1011 110101 010 00100
11110 0111 00001 1000 1110 0011 1000 0001 11011010 00100 1110 0011 1000 0001 010 0011 0111 0011 1000 11001 11000
010 00000 1001 101011 0111 0011 1000 10100 010 0001 1011 110100 1110 11110 11000 1001 11011011 0001 11011010 1011
0111 101011 010 11110 11001 0011 10100 11110 0011 0001 11111 010 00100 1110 11000 00001 001011100 0010101 1000
11001 11110 1011 0111 101011 010 0011 0111 0011 1000 11001 11000 0010100 110110000 010 00000 1001 00001 11110
1110 00001 1110 1011 1110 11111 1001 1011 1110 010 0011 00001 1110 0011 0001 0110110 010 00000 1001 0011 1000 1110
0011 10100 11111 1001 1011 1011 110101 010 11000 11001 1000 1110 01100 10100 010 1110 00001 1000 0111 11000 1001
110111 0110111 1011 1110 011010 1110 010 00100 1110 01100 10100 11111 1001 1011 1011 110101 010 101011 1001
110100 110100 11000 11001 1011 1001 010 01100 110111 110101 010 00000 11000 0001 1011 0111 010 0011 1000 1001
1000 0111 0011 1000 0111 1010100 1011 0111 101011 010 11111 110111 1001 0011 1000 0111 11111 1110 0011 1000 11001
11011010 010 00100 1110 01100 1110 11111 0111 101011 010 1001 110111 110100 1001 11111 0001 1000 0001 11111 010
00100 11110 0111 00001 1000 1110 011010 11110 1001 110100 0001 1010100 1011 0111 101011 010 10101011 0111 110100
11110 0001 11111 010 00000 0011 10100 11111 10100 0010111010 010 110101 00100 0111 11011010 010 0011 1000 1001
11111 010 1110 0011 1011 1110 11111 1110 0010101 010 01100 110111 110101 010 11110 1110 00000 11110 1110 0110110
00100 0111 010 11000 11001 1000 1110 01100 10100 010 110100 1110 11110 11000 10100 11111 1001 1011 1011 110101
010 10101011 0111 110100 11110 0001 11111 010 00000 0011 10100 11111 10100 010 0001 00000 010 00000 1001 0011

1000 1110 0011 10100 11111 1001 1011 1011 110101 11000 010 1110 00001 1000 0111 11000 1001 110111 0110111 1011
1110 011010 1110 010 00100 1110 01100 10100 11111 1001 1011 1011 110101 010 101011 1001 110100 110100 11000 11001
1011 1001 010 01100 110111 110101 010 00001 0001 01100 11111 0111 00101111 11001 1011 1011 110101 010 11001
110100 11001 00100 1000 0111 11111 1011 1110 0011 1000 0001 010 00000 1001 101011 0111 0011 1000 10100 010 00100
1110 11000 00001 001011100 0010101 1000 11001 11110 1011 0111 101011 010 0011 0111 0011 1000 11001 11000 0010100
00001 11110 1001 00100 1000 0111 1010100 1011 1001 010 11011011 0001 1011 1011 0001 0011 1000 0110111 010 11110
1110 00000 11110 1110 0110110 110111 11001 1011 1110 011010 1110 010 11000 11001 1000 1110 01100 10100 010 00001
1110 110111 110101 011010 1001 110110001 010 10100 010 1110 0110110 0010111011 11110 10100 1011 1000 10100 11111
1001 1011 1011 0001 010 11000 1110 10101010 110111 0111 11111 1110 0011 1000 0001 010 00001 0001 01100 11111 0111
00101111 11001 1011 1011 110101 010 11001 110100 11001 00100 1000 0111 11111 1011 1110 0011 1000 0001 010 00100
11110 0111 00001 1000 1110 011010 11110 1001 110100 0001 1010100 1011 1110 011010 1110 010 00000 1001 101011 0111
0011 1000 10100 010 0001 1011 110100 1110 11110 11000 1001 11011011 0001 11011010 1011 0111 101011 010 11110
11001 0011 10100 11110 0011 0001 11111 010 00100 1110 11000 00001 001011100 0010101 1000 11001 11110 1011 0111
101011 010 0011 0111 0011 1000 11001 11000 010 00000 1001 0011 1000 1110 0011 10100 11111 1001 1011 1011 110101
11000 010 11000 11001 1000 1110 01100 0001 11111 010 1110 00001 1000 0111 11000 1001 110111 0110111 1011 1110
011010 1110 010 00100 1110 01100 10100 11111 1001 1011 1011 110101 010 00001 11001 11110 11001 01100 010 11110
11001 1001 110111 0001 00000 1001 11011011 0001 110110001 0010101 010 00100 11110 0111 00001 1000 1110 011010
11110 1001 110100 0001 1010100 1011 0111 101011 010 11000 11001 1000 1110 01100 0001 11111 010 00000 1001 101011
0111 0011 1000 10100 0010111010 010 1001 010 1000 1001 00100 1110 10101010 010 10100 010 11110 1110 00000 11110
1110 0110110 11011011 0001 010 1001 110111 011010 1110 11110 0111 1000 11000 0001 11111 010 11110 11001 1001
110111 0001 00000 1001 11011011 0001 0010101 010 11000 11001 1000 1110 01100 10100 0010100

Послідовний код (T'_{CH}) закодованого оптимальним нерівномірним кодом тексту:
1100011001100010010101100010010110100001001110001100111110001101101110010011100010101010111101110011011
0111010000111010000011110110111110101011010100111011000101011111010000010001011001111101110010111111001
1011101100010101100111010011001001001000011111111101111100011100000010100000001001101011011100111000101 0
0010000110111101001110111101100010011101101100011101101010110111101011010111101100100111010011110001100
01111110100010011101100000000100101110000101011000110011111010110111101011010001101110011100011001110000
1000000100101011110100110101110100101111100010001000000110000001101101110100011100010011000011100111000
0111101010010110111101011010111111101111001001110000111111111110001110001100111011010010001001110011001
1101111101111010110101001110111110100100111111000110000001111110100010011110011100001100011100110101111
0100111010000011010100101101111010110101010101101111101001111000011111101000000001110100111111010001000
0001001001110001110001110100111111001101110111101011100001011100000110000111110001001110111011011110111
1100110101110010001001110011001010011111100110111011110101010101011100111010011010011000110011011100100
1010011100110110001011100110110001001001000111011000010011001110001111011100010110010101010110011011101
1110101010110110001010000011111011101101101111001001101000100111100111000011000111001101011110100111010
0000110101001011111001101011100101010101101111101001111010100111111001101110111101010100110010011011011
11010110101111101000100111011000000001001011100001010110001100111110101101111010110100011011100111000110
01110001001101011010000100000001000000010010011100011100011101001111110011011101111010111000010001001111 0
0111000011000111001101011110100111010000011010100101101111010110101010101101111101001111000011111101000
00000111010011111101000010100000001111101100101100110001100110001110110000100110011100011110111000101100
1010101011001101110111101010101101100010101100011001100011100110001110010111010010100111011101101011101
11100111100011000011101010001001010000000100100111110011011001110100000010001011001111101110010111110011
0111011110101010001001111001110000110001110001110000001110110100010011100011100000010100010011110011100
0011000111001101011110100111010000011010100101101111010110101001110111011010111011110011110001100000011
1111010101010110111110100111101010011111100110111011110101010101000100000010010011100011100011101001111
1100110111011000101001100111001000100111100111000011000111001101011110100111010000011010100101101111010
1101010101011011111010011110000111111010000000011101001111110100001010000000100101100100110101000001010
01100111000111101110001011001010101011001101101101111010101000101111011100110110111010000001010110100111
0111101100010100110111001010100101011000011011100111101010101011100100111000101000000110110111110000100
10101010001111011111101100000101001101100000100110011100011110111000101100011110000111010111110111001001
11011110011100111000111011111101001111110011011000101011111010001001110110000000010010111000010101100011
0011111010110111101011010001101110011100011001110001001101011010001100001111000111110011011001110100001
0101100011001100011100110001110100010011100110010100111111001101110111101010101000100101010101011011111
0100111101010011111100110111011110101010000110111101001110111101100010011101101100010010101010101110010
1000001111101100101100110001100110000101100011101010101011011101111111111110001110000000101000101011011
01000100111011000011011000011011111011111100110111110011010111001000000010010011100011100011101001111110
0110111011110101010011001101111101010100000100010110011111011100101111110011011101111010101011001110100
11001001001000011111111101111100011100000001010000001001101011011100111000101000100001101111010011101111
0110001001110110110001110110101011011110101101011110110010011101001111000110001111110010100010101110010
10000010001011000011100010011111100010101111101111101011111111011111100110110111101011010000001001001001
1101011111011000000111110101111100011100011001110110100101111101110000010111001101010001111000011101000
0011100111110001100001110010010010000111111111011000101010111001000011111011010111000011101010111001101
0000100111011111011110101101001100111000111101110001011001010101011001101101101110101000100101000111101
0011101111011000101001101110010101111111001100001110100000111100011100010011011111011111001001010001000
0001001011001001101010000010010100110110000010111101110000001111011100110110011110000111010110001001100
01100111000100110000111101010010111010001011000111001100110011101110110111010011001101111110101010111101
1001100111011100010000010011101101100010010101010110001100110001110011001010001011010011101111011000101
0011111100110111011110101010101010110111110100111100001111110100000000111010011111101000100001000000100
0000100100111000111000111010011111100110111011110101110000101110000011000011111000100111011101101111011
1110011010111001000100111001100101001111110011011101111010101010101110011101001101001100011001101110010
1001100110111110101010000010001011001111101110010111111001101110111101010101100111010011001001001000011
1111111011111000111000000010100000001001101011011100111000101000100010011101100000000100101110000101011000
1100111110101101111010110100011011100111000110011100000101001101100000101111101110000010111001101010001
1110000111010111000111011111011111101100010100000111101101111110101010101100110111011110101010110001100
1100011100110010100010110100111011110110001010011111100110111011110101010101010110111110100111100001111
110100000000111010011111101000100001000000010000000100100111000111000111010011111100110111011110101110000
10111000000110000111110001001110111011011110111110011010111001000100111001100101001111110011011101111010
1010101011100111010011010011000110011011100101001100110111110101010000010001011001111101110010111111001
101110111101010101100111010011001001001000011111111101111100011100000001010000000100110101101110011100010



ДОДАТОК Г

(обов’язковий)
Реалізація шифру зсуву розрядності r=4

Розбиття послідовного коду закодованого оптимальним нерівномірним кодом тексту на чотирьохрозрядні коди (T'_{CH4}):

1100 0110 0110 0010 0101 0110 0010 0101 1010 0001 0011 1000 1100 1111 1000 1101 1011 1001 0011 1000 1010 1010 1111
0111 0011 0110 1110 1000 0111 0100 0001 1110 1101 1111 0101 0110 1010 0111 0110 0010 1011 1110 1000 0010 0010 1100
1111 1011 1001 0111 1110 0110 1110 1100 0101 0110 0111 0100 1100 1001 0010 0001 1111 1111 0111 1100 0111 0000 0010
1000 0001 0011 0101 1011 1001 1100 0101 0001 0000 1101 1110 1001 1101 1110 1100 0100 1110 1101 1000 1110 1101 0101
1011 1101 0110 1011 1101 1001 0011 1010 0111 1000 1100 0111 1110 1000 1001 1101 1000 0000 1001 0111 0000 1010 1100
0110 0111 1101 0110 1111 0101 1010 0011 0111 0011 1000 1100 1110 0001 0000 0010 0101 0111 1010 0110 1011 1010 0101
1111 0001 0001 0000 0011 0000 0011 0110 1110 1000 1110 0010 0110 0001 1100 1110 0001 1110 1010 0101 1011 1101 0110
1011 1111 1011 1100 1001 1100 0011 1111 1111 1000 1110 0011 0011 1011 0100 1000 1001 1100 1100 1110 1111 1011 1101
0110 1010 0111 0111 1101 0010 0111 1110 0011 0000 0011 1111 0100 0100 1111 0011 1000 0110 0011 1001 1010 1111 0100
1110 1000 0011 0101 0010 1101 1110 1011 0101 0101 0110 1111 1010 0111 1000 0111 1110 1000 0000 0111 0100 1111 1101
0001 0000 0010 0100 1110 0011 1000 1110 1001 1111 1001 1011 1011 1101 0111 0000 1011 1000 0011 0000 1111 1000 1001
1101 1101 1011 1101 1111 0011 0101 1100 1000 1001 1100 1100 1010 0111 1110 0110 1110 1111 0101 0101 0101 1100 1110
1001 1010 0110 0011 0011 0111 0010 0101 0011 1001 1011 0001 0111 0011 0110 0010 0100 1000 1110 1100 0010 0110 0111
0001 1110 1110 0010 1100 1010 1010 1100 1101 1101 1110 1010 1011 0110 0010 1000 0011 1110 1110 1101 1011 1100 1001
1010 0010 0111 1001 1100 0011 0001 1100 1101 0111 1010 0111 0100 0001 1010 1001 0111 1100 1101 0111 0010 1010 1011
0111 1101 0011 1101 0100 1111 1100 1101 1101 1110 1010 1001 1001 0011 0110 1111 0101 1010 1111 1010 0010 0111 0110
0000 0010 0101 1100 0010 1011 0001 1001 1111 0101 1011 1101 0110 1000 1101 1100 1110 0011 0011 1000 1001 1010 1101
0000 1000 0001 0000 0010 0100 1110 0011 1000 1110 1001 1111 1001 1011 1011 1101 0111 0000 1000 1001 1110 0111 0000
1100 0111 0011 0101 1110 1001 1101 0000 0110 1010 0101 1011 1101 0110 1010 1010 1101 1111 0100 1111 0000 1111 1101
0000 0000 1110 1001 1111 1010 0001 0100 0000 1111 1011 0010 1100 1100 0110 0110 0011 1011 0000 1001 1001 1100 0111
1011 1000 1011 0010 1010 1011 0011 0111 0111 1010 1010 1101 1000 1010 1100 0110 0110 0011 1001 1000 1110 0101 1101
0010 1001 1101 1101 1010 1110 1111 0011 1100 0110 0001 1101 0100 0100 1010 0000 0100 1001 1111 0011 0110 0111 0100
0001 0001 0110 0111 1101 1100 1011 1111 0011 0111 0111 1010 1010 0010 0111 1001 1100 0011 0001 1100 0111 0000 0011
1011 0100 0100 1110 0011 1000 0001 0100 0100 1111 0011 1000 0110 0011 1001 1010 1111 0100 1110 1000 0011 0101 0010
1101 1110 1011 0101 0011 1011 1011 0101 1101 1110 0111 1000 1100 0000 1111 1101 0101 0101 1011 1110 1001 1110 1010
0111 1110 0110 1110 1111 0101 0101 0100 0100 0000 1001 0011 1000 1110 0011 1010 0111 1110 0110 1110 1100 0101 0011
0011 1001 0001 0011 1100 1110 0001 1000 1110 0110 1011 1101 0011 1010 0000 1101 0100 1011 0111 1010 1101 0101 0101
1011 1110 1001 1110 0001 1111 1010 0000 0001 1101 0011 1111 0100 0010 1000 0000 1001 0110 0100 1101 0100 0001 0100
1100 1110 0011 1101 1100 0101 1001 0101 0101 1001 1011 0110 1110 1010 1000 1011 1101 1100 1101 1011 1010 0000 0101
0110 1001 1101 1110 1100 0101 0011 0111 0010 1010 0101 0110 0001 1011 1001 1110 1010 1010 1110 0100 1110 0010 1000
0001 1011 0111 1100 0010 0101 0101 0001 1110 1111 1011 0000 0101 0011 0110 0000 1001 1001 1100 0111 1011 1000 1011
0001 1110 0001 1101 0111 1101 1100 1001 1101 1110 0111 0011 1000 1110 1111 1101 0011 1111 0011 0110 0010 1011 1110
1000 1001 1101 1000 0000 1001 0111 0000 1010 1100 0110 0111 1101 0110 1111 0101 1010 0011 0111 0011 1000 1100 1110
0010 0110 1011 0100 0110 0001 1110 0011 1110 0110 1100 1110 1000 0101 0110 0011 0011 0001 1100 1100 0111 0100 0100
1110 0110 0101 0011 1111 0011 0111 0111 1010 1010 1000 1001 0101 0101 0110 1111 1010 0111 1010 1001 1111 1001 1011
1011 1101 0101 0000 1101 1110 1001 1101 1110 1100 0100 1110 1101 1000 1001 0101 0101 0111 0010 1000 0011 1110 1100
1011 0011 0001 1001 1000 0101 1000 1110 1010 1010 1101 1101 1111 1111 1100 0111 0000 0010 1000 1010 1101 0110 1000
1001 1101 1000 0110 1100 0011 0111 1101 1111 1001 1011 1110 0110 1011 1001 0000 0010 0100 1110 0011 1000 1110 1001
1111 1001 1011 1011 1101 0101 0011 0011 0111 1101 0101 0000 0100 0101 1001 1111 0111 0010 1111 1100 1101 1101 1110
1010 1011 0011 1010 0110 0100 1001 0000 1111 1111 1011 1110 0011 1000 0001 0100 0000 1001 1010 1101 1100 1110 0010
1000 1000 0110 1111 0100 1110 1111 0110 0010 0111 0110 1100 0111 0110 1010 1101 1110 1011 0101 1110 1100 1001 1101
0011 1100 0110 0011 1111 0010 1000 1010 1110 0101 0000 0100 0101 1000 0111 0001 0011 1111 0001 0101 1111 0111 1101
0111 1111 1011 1110 0110 1101 1110 1011 0100 0000 1001 0010 0111 0101 1111 0110 0000 0111 1101 0111 1100 0111 0001
1001 1101 1010 0101 1111 0111 0000 0101 1100 1101 0100 0111 1000 0111 0100 0001 1100 1111 1000 1100 0011 1001 0010
0100 0011 1111 1110 1100 0101 0101 1100 1000 0111 1101 1010 1110 0001 1101 0101 1100 1101 0000 1001 1101 1111 0111
1010 1101 0011 0011 1000 1111 0111 0001 0110 0101 0101 0110 0110 1101 1011 1010 1000 1001 0100 0111 1010 0111 0111
1011 0001 0100 1101 1100 1010 1111 1110 0110 0001 1101 0000 0111 1000 1110 0010 0110 1111 1011 1110 0100 1010 0010
0000 0100 1011 0010 0110 1010 0000 1001 0100 1101 1000 0010 1111 0111 0000 0011 1101 1100 1101 1001 1110 0001 1101
0110 0010 0110 0011 0011 1000 1001 1000 0111 1010 1001 0111 0100 0101 1000 1110 0110 0110 0111 0111 0110 1110 1001
1001 1011 1110 1010 1011 1101 1001 1001 1101 1100 0100 0001 0011 1011 0110 0010 0101 0101 0110 0011 0011 0001 1100
1100 1010 0010 1101 0011 1011 1101 1000 1010 0111 1110 0110 1110 1111 0101 0101 0101 0110 1111 1010 0111 1000 0111
1110 1000 0000 0111 0100 1111 1101 0001 0000 1000 0001 0000 0010 0100 1110 0011 1000 1110 1001 1111 1001 1011 1011
1101 0111 0000 1011 1000 0011 0000 1111 1000 1001 1101 1101 1011 1101 1111 0011 0101 1100 1000 1001 1100 1100 1010
0111 1110 0110 1110 1111 0101 0101 0101 1100 1110 1001 1010 0110 0011 0011 0111 0010 1001 1001 1011 1110 1010 1000
0010 0010 1100 1111 1011 1001 0111 1110 0110 1110 1111 0101 0101 1001 1101 0011 0010 0100 1000 0111 1111 1101 1111
0001 1100 0000 1010 0000 0100 1101 0110 1110 0111 0001 0100 0100 0100 1110 1100 0000 0100 1011 1000 0101 0110 0011
0011 1110 1011 0111 1010 1101 0001 1011 1001 1100 0110 0111 0000 0101 0011 0110 0000 1011 1110 1110 0000 1011 1001
1010 1000 1111 0000 1110 1011 1000 1110 1111 1011 1111 0110 0010 1000 0011 1101 1011 1111 0101 0101 0110 0110 1110
1111 0101 0101 1000 1100 1100 0111 0011 0010 1000 1011 0100 1110 1111 0110 0010 1001 1111 1001 1011 1011 1101 0101
0101 0101 1011 1110 1001 1110 0001 1111 1010 0000 0001 1101 0011 1111 0100 0100 0010 0000 0100 0000 1001 0011 1000
1110 0011 1010 0111 1110 0110 1110 1111 0101 1100 0010 1110 0000 1100 0011 1110 0010 0111 0111 0110 1111 0111 1100
1101 0111 0010 0010 0111 0011 0010 1001 1111 1001 1011 1011 1101 0101 0101 0111 0011 1010 0110 1001 1000 1100 1101
1100 1010 0110 0110 1111 1010 1010 0000 1000 1011 0011 1110 1110 0101 1111 1001 1011 1011 1101 0101 0110 0111 0100

1100 1001 0010 0001 1111 1111 0111 1100 0111 0000 0010 1000 0001 0011 0101 1011 1001 1100 0101 0001 0001 0011 1011
0000 0001 0010 1110 0001 0101 1000 1100 1111 1010 1101 1110 1011 0100 0110 1110 0111 0001 1001 1100 0010 1000 1001
0101 1110 1110 0000 0111 1011 1001 1011 0011 1100 0011 1010 1001 1101 1101 1010 1110 1111 0011 1100 0110 0001 1101
0110 1101 0111 0011 0101 1100 1011 1101 1001 1001 1101 1100 0100 0001 0011 1011 0110 0010 0101 0100 1010 0110 1100
0001 0111 0011 0110 0010 1110 1111 1101 0100 1011 1000 1010 0111 1110 0110 0001 1101 0110 0111 0100 1100 1001 0010
0001 1111 1111 0110 0010 0111 0000 1101 1101 0000 0010 0100 0011 1110 1110 0000 1111 0101 1111 0111 1110 0110 1111
1001 1010 1110 0101 1000 1100 1100 0111 0011 0010 1000 1010 0010 0101 0100 1110 1110 1101 0111 0111 1001 1110 0011
0000 0011 1111 0101 1011 0101 1100 1101 0111 0010 1111 0110 0110 0111 0111 0001 0000 0100 1110 1101 1000 1001 0101
0101 1000 1110 0110 0110 0111 0111 0010 1011 1111 1001 1011 1011 1101 0111 0000 0101 0011 0001 1001 1000 1110 0110
0011 1010 0110 0111 0001 1110 1110 0010 1100 1010 1010 1100 1101 1011 0111 0100 1101 1010 0100 0001 0100 0010 1011
0000 1101 1100 1111 0101 0101 0111 0010 1011 1000 1110 1111 1011 1111 0110 1111 0101 1010 0000 1111 0110 1111 1101
0101 0101 1001 1011 1011 1101 0110 1011 0101 0001 1001 1110 1111 0000 1001 0101 0101 1011 0101 1000 1110 1111 1000
1111 1010 1111 1000 1110 0000 0101 0100 0100 1010 1100 0100 1100 0110 0111 0001 0011 0000 1111 0101 0010 1111 1000
1010 1010 0011 1000 1001 1000 0111 0011 1000 0111 0010 0011 1001 0111 0100 1010 0011 0011 1101 1110 0001 0010 1010
1000 0110 1111 0100 1110 1111 0110 0010 0111 0110 1100 0100 1010 1010 1000 1001 0100 0100 1110 0110 0101 0011 1111
0011 0111 0111 1010 1001 0111 0100 1000 1001 1110 0111 0000 1100 0111 0011 0101 1110 1001 1101 0000 0100 1010 1010
1000 1001 0100 0001 1111 0011 1001 0011 0111 1001 0110 0101 1111 0001 0101 0100 0100 1111 0011 1000 0110 0011 1011
0111 1110 0110 1000 0100 1010 1001 0100 1011 1001 1010 0001 0011 1011 1111 0010 1010 1111 1011 1110 1110 0000 1011
1001 0101 1101 0001 1110 0111 1100 0100 1101 1011 1101 0110 1011 1101 1001 0000 0101 0011 0111 0110 1111 0001 0011
0000 0011 1111 0010 1001 1011 0000 0101 1110 1110 0000 0111 1011 1001 1011 0110 1111 1001 1011 1110 0101 0111 1101
1111 1010 0010 1100 0100 1100 0110 0111 0001 0011 0000 1111 0101 0010 1110 1000 1011 0001 1100 1100 1100 1110 1110
1101 1100 1011 1010 0101 1010 1001 0010 0101 0111 1111 1101 0011 0101 1111 0111 1110 1001 1011 0001 0101 1100 0111
1100 1101 1011 0111 0111 1111 1111 0001 1100 0000 1010 0011 1010 0110 0000 0100 1110 1111 1001 1010 1110 0101 1111
0111 0010 0111 0111 1001 1100 1110 0010 0110 1110 1111 0101 0101 1000 1100 1100 0111 0011 0000 0111 1110 1011 1000
0011 0000 1111 1000 1001 1101 1101 1011 1101 1111 0011 0101 1100 1000 1001 1100 1100 1010 0111 1110 0110 1110 1111
0101 0101 0001 0010 1010 1010 1101 1111 0100 1111 0000 1111 1101 0000 0000 1110 1001 1111 1010 0010 0110 0110 1111
1010 1010 0000 1000 1011 0011 1110 1110 0101 1111 1001 1011 1011 1101 0101 0001 0011 1100 1110 0001 1000 1110 0011
1000 0001 1101 1010 0010 0111 0001 1100 0000 1010 0011 0111 0011 1000 1100 1110 0001 0000 0010 0110 1011 0111 0011
1000 1010 0010 0001 1011 1101 0011 1011 1101 1000 1001 1101 1011 0001 1101 1010 1011 0111 1010 1101 0111 1011 0010
0111 0100 1111 0001 1000 1111 1101 0001 0011 1011 0000 0001 0010 1110 0001 0101 1000 1100 1111 1010 1101 1110 1011
0100 0110 1110 0111 0001 1001 1100 0001 0100 1101 1000 0010 0000 0100 1000 0111 1101 1100 0001 1110 1011 1110 1111
1100 1101 1111 0010 0011 0000 1111 0001 1000 1011 0110 0100 0000 1001 0011 1000 1110 0011 1010 0111 1110 0110 1110
1111 0101 0101 1000 1100 1100 0111 0011 0010 1000 1011 1000 0011 0000 1111 1000 1001 1101 1101 1011 1101 1111 0011
0101 1100 1000 1001 1100 1100 1010 0111 1110 0110 1110 1111 0101 0101 0101 1100 1110 1001 1010 0110 0011 0011 0111
0010 1001 1001 1011 1110 1010 1000 0001 1000 0001 1011 0111 0100 0111 0001 0011 0000 1110 0111 0000 1111 0101 0010
1101 1110 1011 0101 1111 1101 1110 0100 1110 0001 1111 1111 1100 0111 0001 1001 1101 1010 0100 0100 1110 0110 0111
0111 1101 1110 1011 0101 0011 1011 1110 1001 0011 1111 0001 1000 0001 1111 1010 0010 0111 1001 1100 0011 0001 1100
1101 0111 1010 0111 0100 0001 1010 1001 0110 1111 0101 1010 1010 1011 0111 1101 0011 1100 0011 1111 0100 0000 0011
1010 0111 1110 1000 0101 1101 0010 1101 0100 1000 1111 1011 0100 1000 1110 0010 0111 1110 1011 1000 1110 1111 1011
1111 1100 0101 0101 0011 0011 0111 1101 0101 0111 1011 1000 0001 1110 1110 0110 1100 0100 0111 0101 1000 1100 1100
0111 0011 0010 1000 1011 0100 1110 1111 0110 0010 1001 1111 1001 1011 1011 1101 0101 0101 0101 1011 1110 1001 1110
0001 1111 1010 0000 0001 1101 0011 1111 0100 0100 0010 0000 0100 0000 1001 0011 1000 1110 0011 1010 0111 1110 0110
1110 1111 0101 1100 0010 1110 0000 1100 0011 1110 0010 0111 0111 0110 1111 0111 1100 1101 0111 0010 0010 0111 0011
0010 1001 1111 1001 1011 1011 1101 0101 0101 0111 0011 1010 0110 1001 1000 1100 1101 1100 1010 0110 0110 1111 1010
1010 0000 1000 1011 0011 1110 1110 0101 1111 1001 1011 1011 1101 0101 0110 0111 0100 1100 1001 0010 0001 1111 1111
0111 1100 0111 0000 0010 1000 0001 0011 0101 1011 1001 1100 0101 0001 0001 0011 1011 0000 0001 0010 1110 0001 0101
1000 1100 1111 1010 1101 1110 1011 0100 0110 1110 0111 0001 1001 1100 0001 0100 0000 1111 1010 0100 1001 0000 1111
0101 0010 1110 0101 0110 1101 1000 1101 1101 1000 1001 1100 0011 0111 0101 1110 1110 0000 0111 1011 1001 1011 0110
1111 1001 1011 1110 0110 1011 1001 0110 0011 0011 0001 1100 1100 1010 0010 0000 1111 0110 1111 1010 1011 0101 0011
1011 0001 0101 0100 0101 1100 1101 1000 1011 1011 1111 0101 0010 1110 0010 1001 1111 1001 1011 1011 0001 0101 1000
1110 1010 1010 1101 1101 1111 1111 1100 0111 0000 0010 1000 0010 0010 1100 1111 1011 1001 0111 1110 0110 1110 1111
0101 0101 1001 1101 0011 0010 0100 1000 0111 1111 1101 1111 0001 1100 0000 1010 0010 0111 1001 1100 0011 0001 1100
1101 0111 1010 0111 0100 0001 1010 1001 0111 1100 1101 0111 0010 0000 0100 1101 0110 1110 0111 0001 0100 0100 0011
0111 1010 0111 0111 1011 0001 0011 1011 0110 0011 1011 0101 0110 1111 0101 1010 1111 0110 0100 1110 1001 1110 0011
0001 1111 1010 0010 0111 0110 0000 0010 0101 1100 0010 1011 0001 1001 1111 0101 1011 1101 0110 1000 1101 1100 1110
0011 0011 1000 0100 0000 1001 0011 1000 1110 0011 1010 0111 1110 0110 1110 1111 0101 1100 0010 1100 0110 0110 0011
1001 1000 0011 1111 0101 1100 0001 1000 0111 1100 0100 1110 1110 1101 1110 1111 1001 1010 1110 0100 0100 1110 0110
0101 0011 1111 0011 0111 0111 1010 1010 0000 1110 0111 1101 1001 0110 0010 1111 0110 0110 0111 0111 0001 0000 0100
1110 1101 1000 1110 1100 0100 1010 1010 0010 0111 1001 1100 0011 0001 1100 1101 0111 1010 0111 0100 0001 1010 1001
0110 1111 0101 1010 1100 0110 0110 0011 1001 1000 0011 1111 0100 0000 1001 1010 1101 1100 1110 0010 1000 0101 1101
0010 1001 0101 0001 0010 0100 1110 1010 1010 0101 0100 0101 1110 1110 0000 0111 1011 1001 1011 0110 1101 1000 1010
1001 1101 1101 1010 1110 1111 0011 1100 0110 0000 0111 1110 1011 1101 1001 1001 1101 1100 0100 0001 0011 1011 0110
0010 0101 0101 0110 0011 0011 0001 1100 1100 1010 0001 0100

Шифрований текст $T_{E5(4)}$, отриманий з використанням шифру зсуву кратності w=5 до тексту T'_{CH4}:

1100 0001 0001 1101 0000 0001 1101 0000 0101 1100 1110 0011 1100 1010 0011 1101 0110 0100 1110 0011 0101 0101 1010
0010 1110 0001 1001 0011 0010 1111 1100 1001 1000 1010 0000 0001 0101 0010 0001 1101 0110 1001 0011 1101 1101 0111
1010 0110 0100 0010 1001 0001 1001 0111 0000 0001 0010 1111 0111 0100 1101 1100 1010 1010 0010 0111 0010 1011 1101

0011 1100 1110 0000 0110 0100 0111 0000 1100 1011 1000 1001 0100 1000 1001 0111 1111 1001 1000 0011 1001 1000 0000
0110 1000 0001 0110 1000 0100 1110 0101 0010 0011 0111 0010 1001 0011 0100 1000 0011 1011 0100 0010 1011 0101 0111
0001 0010 1000 0001 1010 0000 0101 1110 0010 1110 0011 0111 1001 1100 1011 1101 0000 0010 0101 0001 0110 0101 0000
1010 1100 1100 1011 1110 1011 1110 0001 1001 0011 1001 1101 0001 1100 0111 1001 1100 1001 0101 0000 0110 1000 0001
0110 1010 0110 0111 0100 0111 1110 1010 1010 0011 1001 1110 1110 0110 1111 0011 0100 0111 0111 1001 1010 0110 1000
0001 0101 0010 0010 1000 1101 0010 1001 1110 1011 1110 1010 1111 1111 1010 1110 0011 0001 1110 0100 0101 1010 1111
1001 0011 1110 0000 1101 1000 1001 0110 0000 0000 0001 1010 0101 0010 0011 0010 1001 0011 1011 0010 1111 1010 1000
1100 1011 1101 1111 1001 1110 0011 1001 0100 1010 0100 0110 0110 1000 0010 1011 0110 0011 1110 1011 1010 0011 0100
1000 1000 0110 1000 1010 1110 0000 0111 0011 0100 0111 0111 0101 0010 1001 0001 1001 1010 0000 0000 0000 0111 1001
0100 0101 0001 1110 1110 0010 1101 0000 1110 0100 0110 1100 0010 1110 0001 1101 1111 0011 1001 0111 1101 0001 0010
1100 1001 1001 1101 0111 0101 0101 0111 1000 1000 1001 0101 0110 0001 1101 0011 1110 1001 1001 1000 0110 0111 0100
0101 1101 0010 0100 0111 1110 1100 0111 1000 0010 0101 0010 1111 1100 0101 0100 0010 0111 1000 0010 1101 0101 0110
0010 1000 1110 1000 1111 1010 0111 1000 1000 1001 0101 0100 0100 1110 0001 1010 0000 0101 1010 0101 1101 0010 0001
1011 1101 0000 0111 1101 0110 1100 0100 1010 0000 0110 1000 0001 0011 1000 0111 1001 1110 1110 0011 0100 0101 1000
1011 0011 1100 1011 1101 1111 1001 1110 0011 1001 0100 1010 0100 0110 0110 1000 0010 1011 0011 0100 1001 0010 1011
0111 0010 1110 0000 1001 0100 1000 1011 0001 0101 0000 0110 1000 0001 0101 0101 1000 1010 1111 1010 1011 1010 1000
1011 1011 1001 0100 1010 0101 1100 1111 1011 1010 0110 1101 0111 0111 0001 0001 1110 0110 1011 0100 0100 0111 0010
0110 0011 0110 1101 0101 0110 1110 0010 0010 0101 0101 1000 0011 0101 0111 0001 0001 1110 0100 0011 1001 0000 1000
1101 0100 1000 1000 0101 1001 1010 1110 0111 0001 1100 1000 1111 1111 0101 1011 1111 0100 1010 1110 0001 0010 1111
1100 1100 0001 0010 1000 0111 0110 1010 1110 0010 0010 0101 0101 1101 0010 0100 0111 1110 1100 0111 0010 1011 1110
0110 1111 1111 1001 1110 0011 1100 1111 1111 1010 1110 0011 0001 1110 0100 0101 1010 1111 1001 0011 1110 0000 1101
1000 1001 0110 0000 1110 0110 0110 0000 1000 1001 0010 0011 0111 1011 1010 1000 0000 0000 0110 1001 0100 1001 0101
0010 1001 0001 1001 1010 0000 0000 1111 1111 1011 0100 1110 0011 1001 1110 0101 0010 1001 0001 1001 0111 0000 1110
1110 0100 1100 1110 0111 1001 1100 0011 1001 0001 0110 1000 1110 0101 1011 1000 1111 0110 0010 0101 1000 0000 0000
0110 1001 0100 1001 1100 1010 0101 1011 1100 1000 1110 1010 1111 1101 0011 1011 0100 0001 1111 1000 1111 1100 1111
0111 1001 1110 1000 0111 0000 0100 0000 0000 0100 0110 0001 1001 0101 0011 0110 1000 0111 1000 0110 0101 1011 0000
0001 0100 1000 1001 0111 0000 1110 0010 1101 0101 0000 0001 1100 0110 0100 1001 0101 0101 1001 1111 1001 1101 0011
1100 0110 0010 0111 1101 0000 0000 1100 1001 1010 0110 1011 0000 1110 0001 1011 0100 0100 0111 0010 0110 0011 0110
1100 1001 1100 1000 0010 1000 0111 0100 1000 1001 0010 1110 0011 1001 1010 1000 1110 1010 1110 0001 1101 0110 1001
0011 0100 1000 0011 1011 0100 0010 1011 0101 0111 0001 0010 1000 0001 1010 0000 0101 1110 0010 1110 0011 0111 1001
1101 0001 0110 1111 0001 1100 1001 1110 1001 0001 0111 1001 0011 0000 0001 1110 1110 1100 0111 0111 0010 1111 1111
1001 0001 0000 1110 1010 1110 0010 0010 0101 0101 0011 0100 0000 0000 0001 1010 0101 0010 0101 0100 1010 0100 0110
0110 1000 0000 1011 1000 1001 0100 1000 1001 0111 1111 1001 1000 0011 0100 0000 0000 0010 1101 0011 1110 1001 0111
0110 1110 1100 0100 0011 0000 0011 1001 0101 0101 1000 1000 1010 1010 0111 0010 1011 1101 0011 0101 1000 0001 0011
0100 1000 0011 0001 0111 1110 0010 1000 1010 0100 0110 1001 0001 0110 0100 1011 1101 1111 1001 1110 0011 1001 0100
1010 0100 0110 0110 1000 0000 1110 1110 0010 1000 0000 1011 1111 0000 0100 1010 0010 1101 1010 0111 1000 1000 1001
0101 0110 1110 0101 0001 1111 0100 1011 1010 1010 0110 1001 1110 0011 1100 1111 1011 0100 0101 1000 0111 1001 1101
0011 0011 0001 1010 1111 1001 1010 0001 1101 0010 0001 0111 0010 0001 0101 1000 1001 0110 0000 1001 0111 0100 1000
1110 0111 0001 1110 1010 1101 0011 0101 1001 0000 1011 1111 0000 0011 0010 1100 1110 1010 1100 0000 1010 0010 1000
0010 1010 0110 1001 0001 1000 1001 0110 1111 1011 0100 1101 0010 0000 1010 0001 1011 0010 1000 0010 0111 0010 1100
0100 1000 0101 0000 1010 0010 1011 0000 0111 1000 1111 0010 0011 0010 1111 1100 0111 1010 0011 0111 1110 0100 1101
1111 1110 1010 1001 0111 0000 0000 0111 0011 0010 1000 0101 1001 1100 1000 0000 0111 1000 1011 0100 1000 1010 0010
0101 1000 1110 1110 0011 1010 0010 1100 0001 0000 0000 0001 0001 1000 0110 0101 0011 0100 1111 0010 0101 0010 0010
0110 1100 1111 1000 0111 0101 1010 1001 0001 1100 1000 1011 0010 0011 1001 1101 0001 1010 0110 1001 1111 0101 1101
1011 1111 0110 1101 0001 0101 1011 0100 1111 1000 0011 1101 1010 0010 1011 1110 1000 0111 1000 0100 1001 1100 1000
0001 1101 0001 1110 1110 0011 0100 0011 0010 0101 0100 0010 1111 0000 0011 1001 0001 0001 0010 0010 0001 1001 0100
0100 0110 1001 0101 0110 1000 0100 0100 1000 0111 1111 1100 1110 0110 0001 1101 0000 0000 0001 1110 1110 1100 0111
0111 0101 1101 1000 1110 0110 1000 0011 0101 0010 1001 0001 1001 1010 0000 0000 0000 0001 1010 0101 0010 0011 0010
1001 0011 1011 0010 1111 1010 1000 1100 1011 0011 1100 1011 1101 1111 1001 1110 0011 1001 0100 1010 0100 0110 0110
1000 0010 1011 0110 0011 1110 1011 1010 0011 0100 1000 1000 0110 1000 1010 1110 0000 0111 0011 0100 0111 0111 0101
0010 1001 0001 1001 1010 0000 0000 0000 0111 1001 0100 0101 0001 1110 1110 0010 1101 0100 0100 0110 1001 0101 0011
1101 1101 0111 1010 0110 0100 0010 1001 0001 1001 1010 0000 0000 0100 1000 1110 1101 1111 0011 0010 1010 1000 1010
1100 0111 1011 0101 1011 1111 1000 0001 1001 0010 1100 1111 1111 1111 1001 0111 1011 1111 0110 0011 0000 0001 1110
1110 1001 0110 0010 0101 1000 1100 0110 0100 0111 0001 0010 1011 0000 1110 0001 1011 0110 1001 1001 1011 0110 0100
0101 0011 1010 1011 1001 0110 0011 1001 1010 0110 1010 0001 1101 0011 1110 1000 0110 1010 0000 0000 0001 0001 1001
1010 0000 0000 0011 0111 0111 0010 1110 1101 0011 0110 1111 1001 1010 0001 1101 0100 1010 0100 0110 0110 1000 0000
0000 0000 0110 1001 0100 1001 1100 1010 0101 1011 1100 1000 1110 1010 1111 1111 1101 1011 1111 1011 0100 1110 0011
1001 1110 0101 0010 1001 0001 1001 1010 0000 0111 1101 1001 1011 0111 1110 1001 1101 0010 0010 0001 1010 0010 0111
1000 0010 1101 1101 0010 1110 1101 0100 1010 0100 0110 0110 1000 0000 0000 0010 1110 0101 0001 0100 0011 0111 1000
0111 0101 0001 0001 1010 0101 0101 1011 0011 0110 1110 1001 1001 0000 1010 0100 0110 0110 1000 0000 0001 0010 1111
0111 0100 1101 1100 1010 1010 0010 0111 0010 1011 1101 0011 1100 1110 0000 0110 0100 0111 0000 1100 1100 1110 0110
1011 1100 1101 1001 1100 0000 0011 0111 1010 0101 1000 1001 0110 1111 0001 1001 0010 1100 0100 0111 1101 0011 0100
0000 1001 1001 1011 0010 0110 0100 0110 1110 0111 1110 0101 0100 1000 1000 0101 1001 1010 1110 0111 0001 1100 1000
0001 1000 0010 1110 0000 0111 0110 1000 0100 0100 1000 0111 1111 1100 1110 0110 0001 1101 0000 1111 0101 0001 0111
1100 0010 1110 0001 1101 1001 1010 1000 1111 0110 0011 0101 0010 1001 0001 1100 1000 0001 0010 1111 0111 0100 1101
1100 1010 1010 0001 1101 0010 1011 1000 1000 1011 1101 1111 1110 1001 1001 1011 1010 0000 1010 0010 1001 0001 1010
0100 0101 1001 0000 0011 0111 0111 0010 1110 1101 0011 0101 1101 0000 1111 1001 1001 1000 0010 0010 0100 1001 1110
1011 1110 1010 0000 0110 0000 0111 1000 0010 1101 1010 0001 0001 0010 0010 1100 1011 1111 1001 1000 0011 0100 0000
0000 0011 1001 0001 0001 0010 0010 1101 0110 1010 0100 0110 0110 1000 0010 1011 0000 1110 1100 0100 0011 1001 0001
1110 0101 0001 0010 1100 1001 1001 1101 0111 0101 0101 0111 1000 0110 0010 1111 1000 0101 1111 1100 1111 1101 0110
1011 1000 0111 1010 0000 0000 0010 1101 0110 0011 1001 1010 0110 1010 0001 1010 0000 0101 1011 1010 0001 1010 1000
0000 0000 0100 0110 0110 1000 0001 0110 0000 1100 0100 1001 1010 1011 0100 0000 0000 0110 0000 0011 1001 1010 0011

1010 0101 1010 0011 1001 1011 0000 1111 1111 0101 0111 1111 0111 0001 0010 1100 1110 1011 1010 0000 1101 1010 0011
0101 0101 1110 0011 0100 0011 0010 1110 0011 0010 1101 1110 0100 0010 1111 0101 1110 1110 1000 1001 1100 1101 0101
0011 0001 1010 1111 1001 1010 0001 1101 0010 0001 0111 1111 0101 0101 0011 0100 1111 1111 1001 0001 0000 1110 1010
1110 0010 0010 0101 0100 0010 1111 0011 0100 1001 0010 1011 0111 0010 1110 0000 1001 0100 1000 1011 1111 0101 0101
0011 0100 1111 1100 1010 1110 0100 1110 0010 0100 0001 0000 1010 1100 0000 1111 1111 1010 1110 0011 0001 1110 0110
0010 1001 0001 0011 1111 0101 0100 1111 0110 0100 0101 1100 1110 0110 1010 1101 0101 1010 0110 1001 1001 1011 0110
0100 0000 1000 1100 1001 0010 0111 1111 1000 0110 1000 0001 0110 1000 0100 1011 0000 1110 0010 0001 1010 1100 1110
1011 1110 1010 1101 0100 0110 1011 0000 1001 1001 1011 0010 0110 0100 0110 0001 1010 0100 0110 1001 0000 0010 1000
1010 0101 1101 0111 1111 0111 0001 0010 1100 1110 1011 1010 0000 1101 1001 0011 0110 1100 0111 0111 0111 1001 1001
1000 0111 0110 0101 0000 0101 0100 1101 0000 0010 1010 1000 1110 0000 1010 0010 1001 0100 0110 1100 0000 0111 0010
0111 1000 0110 0010 0010 1010 1010 1100 0111 1011 0101 1110 0101 0001 1011 1111 1001 1010 0100 0101 1001 0000 1010
0010 1101 0010 0010 0100 0111 1001 1101 0001 1001 1010 0000 0000 0011 0111 0111 0010 1110 1011 0010 1001 0110 0011
1110 1011 1010 0011 0100 1000 1000 0110 1000 1010 1110 0000 0111 0011 0100 0111 0111 0101 0010 1001 0001 1001 1010
0000 0000 1100 1101 0101 0101 1000 1010 1111 1010 1011 1010 1000 1011 1011 1001 0100 1010 0101 1101 0001 0001 1010
0101 0101 1011 0011 0110 1110 1001 1001 0000 1010 0100 0110 0110 1000 0000 1100 1110 0111 1001 1100 0011 1001 1110
0011 1100 1000 0101 1101 0010 1100 0111 1011 0101 1110 0010 1110 0011 0111 1001 1100 1011 1101 0001 0110 0010 1110
0011 0101 1101 1100 0110 1000 1110 0110 1000 0011 0100 1000 0110 1100 1000 0101 0110 0010 0101 1000 0010 0110 1101
0010 1111 1010 1100 0011 1010 1000 1100 1110 0110 1011 1100 1101 1001 1100 0000 0011 0111 1010 0101 1000 1001 0110
1111 0001 1001 0010 1100 0100 0111 1100 1111 1000 0011 1101 1011 1111 0011 0010 1000 0111 1100 1001 0110 1001 1010
0111 1000 1010 1101 1110 1011 1010 1100 0011 0110 0001 1111 1011 0100 1110 0011 1001 1110 0101 0010 1001 0001 1001
1010 0000 0000 0011 0111 0111 0010 1110 1101 0011 0110 0011 1110 1011 1010 0011 0100 1000 1000 0110 1000 1010 1110
0000 0111 0011 0100 0111 0111 0101 0010 1001 0001 1001 1010 0000 0000 0000 0111 1001 0100 0101 0001 1110 1110 0010
1101 0100 0100 0110 1001 0101 0011 1100 0011 1100 0110 0010 1111 0010 1100 1110 1011 1001 0010 1011 1010 0000 1101
1000 1001 0110 0000 1010 1000 1001 1111 1001 1100 1010 1010 0111 0010 1100 0100 1000 0101 1111 1111 1001 0001 0010
0010 1000 1001 0110 0000 1110 0110 1001 0100 1110 1010 1100 0011 1100 1010 0101 1101 0010 0100 0111 1110 1100 0111
1000 0010 0101 0010 1111 1100 0101 0100 0001 1010 0000 0101 0101 0110 0010 1000 1110 0111 1110 1010 1111 1011 1110
0101 0010 1001 0011 0000 1000 1101 1000 1111 0011 1010 0110 1111 0011 1001 1101 0010 1001 0110 0011 1001 1010 0110
1010 0111 0000 0000 1110 1110 0010 1000 0000 0010 0110 0011 1100 1001 1001 0001 0111 1111 0010 0000 0011 0111 0111
0010 1110 1101 0011 0110 1111 1001 1010 0001 1101 0100 1010 0100 0110 0110 1000 0000 0000 0000 0110 1001 0100 1001
1100 1010 0101 1011 1100 1000 1110 1010 1111 1111 1101 1011 1111 1011 0100 1110 0011 1001 1110 0101 0010 1001 0001
1001 1010 0000 0111 1101 1001 1011 0111 1110 1001 1101 0010 0010 0001 1010 0010 0111 1000 0010 1101 1101 0010 1110
1101 0100 1010 0100 0110 0110 1000 0000 0000 0010 1110 0101 0001 0100 0011 0111 1000 0111 0101 0001 0001 1010 0101
0101 1011 0011 0110 1110 1001 1001 0000 1010 0100 0110 0110 1000 0000 0001 0010 1111 0111 0100 1101 1100 1010 1010
0010 0111 0010 1011 1101 0011 1100 1110 0000 0110 0100 0111 0000 1100 1100 1110 0110 1011 1100 1101 1001 1100 0000
0011 0111 1010 0101 1000 1001 0110 1111 0001 1001 0010 1100 0100 0111 1100 1111 1011 1010 0101 1111 0100 1011 1010
0000 1101 1001 0000 0001 1000 0011 1000 1000 0011 0100 0111 1110 0010 0000 1001 1001 1011 0010 0110 0100 0110 0001
1010 0100 0110 1001 0001 0110 0100 0001 1110 1110 1100 0111 0111 0101 1101 1011 1010 0001 1010 0101 0110 0000 1110
0110 1100 0000 1111 0000 0111 1000 0011 0110 0110 1010 0000 1101 1001 1101 0100 1010 0100 0110 0110 1100 0000 0011
1001 0101 0101 1000 1000 1010 1010 0111 0010 1011 1101 0011 1101 1101 0111 1010 0110 0100 0010 1001 0001 1001 1010
0000 0000 0100 1000 1110 1101 1111 0011 0010 1010 1000 1010 1100 0111 1011 0101 1101 0010 0100 0111 1110 1100 0111
1000 0010 0101 0010 1111 1100 0101 0100 0010 0111 1000 0010 1101 1011 1111 1000 0001 1001 0010 1100 1111 1111 1110
0010 0101 0010 0010 0110 1100 1110 0110 0001 1110 0110 0000 0001 1010 0000 0101 1010 0001 1111 1001 0100 1001 1110
1100 1010 0101 1101 0010 0001 1011 1101 0000 0111 1101 0110 1100 0100 1010 0000 0110 1000 0001 0011 1000 0111 1001
1110 1110 0011 1111 1011 0100 1110 0011 1001 1110 0101 0010 1001 0001 1001 1010 0000 0111 1101 0111 0001 0001 1110
0100 0011 1110 1010 0000 0111 1100 0011 0010 0111 1111 1001 1001 1000 1001 1010 0100 0101 1001 1111 1111 1001 0001
0000 1110 1010 1110 0010 0010 0101 0101 1011 1001 0010 1000 0100 0001 1101 1010 0001 0001 0010 0010 1100 1011 1111
1001 1000 0011 1001 0111 1111 0101 0101 1101 0010 0100 0111 1110 1100 0111 1000 0010 0101 0010 1111 1100 0101 0100
0001 1010 0000 0101 0111 0001 0001 1110 0100 0011 1110 1010 1111 1011 0100 0101 1000 0111 1001 1101 0011 0000 1000
1101 0100 0000 1100 1101 1111 1001 0101 0101 0000 1111 0000 1001 1001 1011 0010 0110 0100 0110 0001 1000 0011 0101
0100 1000 1000 0101 1001 1010 1110 0111 0001 1011 0010 1001 0110 1000 0100 0100 1000 0111 1111 1100 1110 0110 0001
1101 0000 0000 0001 1110 1110 1100 0111 0111 0101 1100 1111

ДОДАТОК Д

(обов’язковий)
Реалізація шифру зсуву розрядності r=5

Розбиття послідовного коду закодованого оптимальним нерівномірним кодом тексту на чотирьохрозрядні коди (T'_{CH5}):

00001 00001 00010 00010 00000 10000 00100 00001 00000 00010 00000 00010 00000 00010 00000 01000 10010 00010 00001
00000 01000 00010 00100 01000 10000 00100 00100 00000 10000 00010 00001 00010 00010 00001 00000 10000 10000 00001
00000 00010 00000 01000 00001 00000 00100 00010 00100 00100 00111 00000 00001 00010 01000 01010 00010 10100 10110
00101 11000 01011 10100 01011 10110 01011 11001 10100 11000 11010 01101 10011 01110 11110 00100 11010 10010 10010
10101 01010 10111 01011 10111 10001 10100 11010 11100 11101 10000 11011 00011 10110 01110 11010 11011 01111 01111
11011 11011 11110 00011 11000 11111 00110 10100 01010 00101 11110 01101 11001 11100 01001 10101 00000 00101 11010
01100 11001 00010 01000 00100 00100 10000 01110 00000 11001 11001 00010 00100 10010 10000 00010 01011 11001 00111
00001 10100 11001 10010 10110 00110 11000 11001 00101 00100 11111 10000 10100 10110 01001 11010 00010 10100 01001
10010 11010 00101 11000 11101 00100 00100 00010 11000 11001 10001 00101 01100 01001 01101 00001 00111 00011 00111
11000 11011 01110 01001 11000 10101 01011 11011 10011 01101 11010 00011 10100 00011 11011 01111 10101 01101 01001
11011 00010 10111 11010 00001 00010 11001 11110 11100 10111 11100 11011 10110 00101 01100 11101 00110 01001 00100
00111 11111 10111 11000 11100 00001 01000 00010 01101 01101 11001 11000 10100 01000 01101 11101 00111 01111 01100
01001 11011 01100 01110 11010 10110 11110 10110 10111 10110 01001 11010 01111 00011 00011 11110 10001 00111 01100
00000 10010 11100 00101 01100 01100 11111 01011 01111 01011 01000 11011 10011 10001 10011 10000 10000 00100 10101
11101 00110 10111 01001 01111 10001 00010 00000 11000 00011 01101 11010 00111 00010 01100 00111 00111 00001 11101
01001 01101 11101 01101 01111 11101 11100 10011 10000 11111 11111 10001 11000 11001 11011 01001 00010 01110 01100
11101 11110 11110 10110 10100 11101 11110 10010 01111 11000 11000 00011 11110 10001 00111 10011 10000 11000 11100
11010 11110 10011 10100 00011 01010 01011 01111 01011 01010 10101 10111 11010 01111 00001 11111 01000 00000 11101
00111 11101 00010 00000 10010 01110 00111 00011 10100 11111 10011 01110 11110 10111 00001 01110 00001 10000 11111
00010 01110 11101 10111 10111 11001 10101 11001 00010 01110 01100 10100 11111 10011 01110 11110 10101 01010 11100
11101 00110 10011 00011 00110 11100 10010 10011 10011 01100 01011 10011 01100 01001 00100 01110 11000 01001 10011
10001 11101 11000 10110 01010 10101 10011 01110 11110 10101 01101 10001 01000 00111 11011 10110 11011 11001 00110
10001 00111 10011 10000 11000 11100 11010 11110 10011 10100 00011 01010 01011 11100 11010 11100 10101 01011 01111
10100 11110 10100 11111 10011 01110 11110 10101 00110 01001 10110 11110 10110 10111 11010 00100 11101 10000 00010
01011 10000 10101 10001 10011 11101 01101 11101 01101 00011 01110 01110 00110 01110 00100 11010 11010 00010 00000
10000 00100 10011 10001 11000 11101 00111 11100 11011 10111 10101 11000 01000 10011 11001 11000 01100 01110 01101
01111 01001 11010 00001 10101 00101 10111 10101 10101 01010 11011 11101 00111 10000 11111 10100 00000 01110 10011
11110 10000 10100 00001 11110 11001 01100 11000 11001 10001 11011 00001 00110 01110 00111 10111 00010 11001 01010
10110 01101 11011 11010 10101 10110 00101 01100 01100 11000 11100 11000 11100 10111 01001 01001 11011 10110 10111
01111 00111 10001 10000 11101 01000 10010 10000 00100 10011 11100 11011 00111 01000 00100 01011 00111 11011 10010
11111 10011 01110 11110 10101 00010 01111 00111 00001 10001 11000 11100 00001 11011 01000 10011 10001 11000 00010
10001 00111 10011 10000 11000 11100 11010 11110 10011 10100 00011 01010 01011 01111 01011 01010 01110 11101 10101
11011 11001 11100 01100 00001 11111 01010 10101 10111 11010 01111 01010 01111 11001 10111 01111 01010 10101 00010
00000 10010 01110 00111 00011 10100 11111 10011 01110 11000 10100 11001 11001 00010 01111 00111 00001 10001 11001
10101 11101 00111 01000 00110 10100 10110 11110 10110 10101 01011 01111 10100 11110 00011 11110 10000 00001 11010
01111 11010 00010 10000 00010 01011 00100 11010 10000 01010 01100 11100 01111 01110 00101 10010 10101 01100 11011
01101 11010 10100 01011 11011 10011 01101 11010 00000 10101 10100 11101 11101 10001 01001 10111 00101 01001 01011
00001 10111 00111 10101 01010 11100 10011 10001 01000 00011 01101 11110 00010 01010 10100 01111 01111 10110 00001
01001 10110 00001 00110 01110 00111 10111 00010 11000 11110 00011 10101 11110 11100 10011 10111 10011 10011 10001
11011 11110 10011 11110 01101 10001 01011 11101 00010 01110 11000 00001 00101 11000 01010 11000 11001 11110 10110
11110 10110 10001 10111 00111 00011 00111 00010 01101 01101 00011 00001 11100 01111 10011 01100 11101 00001 01011
00011 00110 00111 00110 00111 01000 10011 10011 00101 00111 11100 11011 10111 10101 01010 00100 10101 01010 11011
11101 00111 10101 00111 11100 11011 10111 10101 01000 01101 11101 00111 01111 01100 01001 11011 01100 01001 01010
10101 11001 01000 00111 11011 00101 10011 00011 00110 00010 11000 11101 01010 10110 11101 11111 11111 00011 10000
00101 00010 10110 10110 10001 00111 01100 00110 11000 01101 11110 11111 10011 01111 10011 01011 10010 00000 10010
01110 00111 00011 10100 11111 10011 01110 11110 10101 00110 01101 11110 10101 00000 10001 01100 11111 01110 01011
11110 01101 11011 11010 10101 10011 10100 11001 00100 10000 11111 11110 11111 00011 10000 00101 00000 01001 10101
10111 00111 00010 10001 00001 10111 10100 11101 11101 10001 00111 01101 10001 11011 01010 11011 11010 11010 11110
11001 00111 01001 11100 01100 01111 11001 01000 10101 11001 01000 00100 01011 00001 11000 10011 11110 00101 01111
10111 11010 11111 11101 11110 01101 10111 10101 10100 00001 00100 10011 10101 11110 11000 00011 11101 01111 10001
11000 11001 11011 01001 01111 10111 00000 10111 00110 10100 01111 00001 11010 00001 11001 11110 00110 00011 10010
01001 00001 11111 11101 10001 01010 11100 10000 11111 01101 01110 00011 10101 01110 01101 00001 00111 01111 10111
10101 10100 11001 11000 11110 11100 01011 00101 01010 11001 10110 11011 10101 00010 01010 00111 10100 11101 11101
10001 01001 10111 00101 01111 11100 11000 01110 10000 01111 00011 10001 00110 11111 01111 10010 01010 00100 00001
00101 10010 01101 01000 00100 10100 11011 00000 10111 10111 00000 01111 01110 01101 10011 11000 01110 10110 00100
11000 11001 11000 10011 00001 11101 01001 01110 10001 01100 01110 01100 11001 11011 10110 11101 00110 01101 11110
10101 01111 01100 11001 11011 10001 00000 10011 10110 11000 10010 10101 01100 01100 11000 11100 11001 01000 10110
10011 10111 10110 00101 00111 11100 11011 10111 10101 01010 10101 10111 11010 01111 00001 11111 01000 00000 11101
00111 11101 00010 00010 00000 10000 00100 10011 10001 11000 11101 00111 11100 11011 10111 10101 11000 01011 10000
01100 00111 11000 10011 10111 01101 11101 11110 01101 01110 01000 10011 10011 00101 00111 11100 11011 10111 10101
01010 10111 00111 01001 10100 11000 11001 10111 00101 00110 01101 11110 10101 00000 10001 01100 11111 01110 01011
11110 01101 11011 11010 10101 10011 10100 11001 00100 10000 11111 11110 11111 00011 10000 00101 00000 01001 10101
10111 00111 00010 10001 00010 01110 11000 00001 00101 11000 01010 11000 11001 11110 10110 11110 10110 10001 10111

00111 00011 00111 00000 10100 11011 00000 10111 11011 10000 01011 10011 01010 00111 10000 11101 01110 00111 01111
10111 11101 10001 01000 00111 10110 11111 10101 01010 11001 10111 01111 01010 10110 00110 01100 01110 01100 10100
01011 01001 11011 11011 00010 10011 11110 01101 11011 11010 10101 01010 11011 11101 00111 10000 11111 10100 00000
01110 10011 11110 10001 00001 00000 01000 00010 01001 11000 11100 01110 10011 11110 01101 11011 11010 11100 00101
11000 00110 00011 11100 01001 11011 10110 11110 11111 00110 10111 00100 01001 11001 10010 10011 11110 01101 11011
11010 10101 01011 10011 10100 11010 01100 01100 11011 10010 10011 00110 11111 01010 10000 01000 10110 01111 10111
00101 11111 00110 11101 11101 01010 11001 11010 01100 10010 01000 01111 11111 01111 10001 11000 00010 10000 00100
11010 11011 10011 10001 01000 10001 00111 01100 00000 10010 11100 00101 01100 01100 11111 01011 01111 01011 01000
11011 10011 10001 10011 10000 10100 01001 01011 11011 10000 00111 10111 00110 11001 11100 00111 01010 01110 11101
10101 11011 11001 11100 01100 00111 01011 01101 01110 01101 01110 01011 11011 00110 01110 11100 01000 00100 11101
10110 00100 10101 00101 00110 11000 00101 11001 10110 00101 11011 11110 10100 10111 00010 10011 11110 01100 00111
01011 00111 01001 10010 01001 00001 11111 11101 10001 00111 00001 10111 01000 00010 01000 01111 10111 00000 11110
10111 11011 11110 01101 11110 01101 01110 01011 00011 00110 00111 00110 01010 00101 00010 01010 10011 10111 01101
01110 11110 01111 00011 00000 01111 11010 11011 01011 10011 01011 10010 11110 11001 10011 10111 00010 00001 00111
01101 10001 00101 01010 11000 11100 11001 10011 10111 00101 01111 11100 11011 10111 10101 11000 00101 00110 00110
01100 01110 01100 01110 10011 00111 00011 11011 10001 01100 10101 01011 00110 11011 01110 10011 01101 00100 00010
10000 10101 10000 11011 10011 11010 10101 01110 01010 11100 01110 11111 01111 11011 01111 01011 01000 00111 10110
11111 10101 01010 11001 10111 01111 01011 01011 01010 00110 01111 01111 00001 00101 01010 11011 01011 00011 10111
11000 11111 01011 11100 01110 00000 10101 00010 01010 11000 10011 00011 00111 00010 01100 00111 10101 00101 11110
00101 01010 00111 00010 01100 00111 00111 00001 11001 00011 10010 11101 00101 00011 00111 10111 10000 10010 10101
00001 10111 10100 11101 11101 10001 00111 01101 10001 00101 01010 10001 00101 00010 01110 01100 10100 11111 10011
01110 11110 10100 10111 01001 00010 01111 00111 00001 10001 11001 10101 11101 00111 01000 00100 10101 01010 00100
10100 00011 11100 11100 10011 01111 00101 10010 11111 00010 10101 00010 01111 00111 00001 10001 11011 01111 11001
10100 00100 10101 00101 00101 11001 10100 00100 11101 11111 00101 01011 11101 11110 11100 00010 11100 10101 11010
00111 10011 11100 01001 10110 11110 10110 10111 10110 01000 00101 00110 11101 10111 10001 00110 00000 11111 10010
10011 01100 00010 11110 11100 00001 11101 11001 10110 11011 11100 11011 11100 10101 11110 11111 10100 01011 00010
01100 01100 11100 01001 10000 11110 10100 10111 01000 10110 00111 00110 01100 11101 11011 01110 01011 10100 10110
10100 10010 01010 11111 11110 10011 01011 11101 11111 01001 10110 00101 01110 00111 11001 10110 11011 10111 11111
11100 01110 00000 10100 01110 10011 00000 01001 11011 11100 11010 11100 10111 11011 10010 01110 11110 01110 01110
00100 11011 10111 10101 01011 00011 00110 00111 00110 00001 11111 01011 10000 01100 00111 11000 10011 10111 01101
11101 11110 01101 01110 01000 10011 10011 00101 00111 11100 11011 10111 10101 01010 00100 10101 01010 11011 11101
00111 10000 11111 10100 00000 01110 10011 11110 10001 00110 01101 11110 10101 00000 10001 01100 11111 01110 01011
11110 01101 11011 11010 10100 01001 11100 11100 00110 00111 00011 10000 00111 01101 00010 01110 00111 00000 01010
00110 11100 11100 01100 11100 00100 00001 00110 10110 11100 11100 01010 00100 00110 11110 10011 10111 10110 00100
11101 10110 00111 01101 01011 01111 01011 01011 11011 00100 11101 00111 10001 10001 11111 01000 10011 10110 00000
01001 01110 00010 10110 00110 01111 10101 10111 10101 10100 01101 11001 11000 11001 11000 00101 00110 11000 00100
00001 00100 00111 11011 10000 01111 01011 11101 11111 00110 11111 00100 01100 00111 10001 10001 01101 10010 00000
10010 01110 00111 00011 10100 11111 10011 01110 11110 10101 01100 01100 11000 11100 11001 01000 10111 00000 11000
01111 10001 00111 01110 11011 11011 11100 11010 11100 10001 00111 00110 01010 01111 11001 10111 01111 01010 10101
01110 01110 10011 01001 10001 10011 01110 01010 01100 11011 11101 01010 00000 11000 00011 01101 11010 00111 00010
01100 00111 00111 00001 11101 01001 01101 11101 01101 01111 11101 11100 10011 10000 11111 11111 10001 11000 11001
11011 01001 00010 01110 01100 11101 11110 11110 10110 10100 11101 11110 10010 01111 11000 11000 00011 11110 10001
00111 10011 10000 11000 11100 11010 11110 10011 10100 00011 01010 01011 01111 01011 01010 10101 10111 11010 01111
00001 11111 01000 00000 11101 00111 11101 00001 01110 10010 11010 10010 00111 11011 01001 00011 10001 00111 11101
01110 00111 01111 10111 11111 00010 10101 00110 01101 11110 10101 01111 01110 00000 11110 11100 11011 00010 00111
01011 00011 00110 00111 00110 01010 00101 10100 11101 11101 10001 01001 11111 00110 11101 11101 01010 10101 01101
11110 10011 11000 01111 11010 00000 00111 01001 11111 01000 10000 10000 00100 00001 00100 11100 01110 00111 01001
11111 00110 11101 11101 01110 00010 11100 00011 00001 11110 00100 11101 11011 01111 01111 10011 01011 10010 00100
11100 11001 01001 11111 00110 11101 11101 01010 10101 11001 11010 01101 00110 00110 01101 11001 01001 10011 01111
10101 01000 00100 01011 00111 11011 10010 11111 10011 01110 11110 10101 01100 11101 00110 01001 00100 00111 11111
10111 11000 11100 00001 01000 00010 01101 01101 11001 11000 10100 01000 10011 10110 00000 01001 01110 00010 10110
00110 01111 10101 10111 10101 10100 01101 11001 11000 11001 11000 00101 00000 01111 10100 10010 01000 01111 01010
01011 10010 10110 11011 00011 01110 11000 10011 10000 11011 10101 11101 11000 00011 11011 10011 01101 10111 11001
10111 11001 10101 11001 01100 01100 11000 11100 11001 01000 10000 01111 01101 11110 10101 10101 00111 01100 01010
10100 01011 10011 01100 01011 10111 11101 01001 01110 00101 00111 11100 11011 10110 00101 01100 01110 10101 01011
01110 11111 11111 10001 11000 00010 10000 01000 10110 01111 10111 00101 11111 00110 11101 11101 01010 11001 11010
01100 10010 01000 01111 11111 01111 10001 11000 00010 10001 00111 10011 10000 11000 11100 11010 11110 10011 10100
00011 01010 01011 11100 11010 11100 10000 00100 11010 11011 10011 10001 01000 10000 11011 11010 01110 11110 11000
10011 10110 11000 11101 10101 01101 11101 01101 01111 01100 10011 10100 11110 00110 00111 11101 00010 01110 11000
00001 00101 11000 01010 11000 11001 11110 10110 11110 10110 10001 10111 00111 00011 00111 00001 00000 01001 00111
00011 10001 11010 01111 11001 10111 01111 01011 10000 10110 00110 01100 01110 01100 00011 11110 10111 00000 11000
01111 10001 00111 01110 11011 11011 11100 11010 11100 10001 00111 00110 01010 01111 11001 10111 01111 01010 10000
01111 11010 00000 10011 01011 01110 01110 00101 00001 01110 10010 10010 10100 01001 00100 11101 01010 10010 10100
01011 11011 10000 00111 10111 00110 11011 01101 10001 01010 01110 11101 10101 11011 11001 11100 01100 00001 11111
01011 11011 00110 01110 11100 01000 00100 11101 10110 00100 10101 01011 00011 00110 00111 00110 01010 00010 10000

11100 11100 11101 11101 11011 01011 11111 11100 11011 11101 11011 11101 11011 11101 11011 00011 01101 11101 11100
11011 00011 11101 11111 00011 01011 11111 11111 11011 01011 11101 11100 11101 11101 11100 11011 01011 01011 11100
11011 11101 11011 00011 11100 11011 11111 11101 11111 11111 00010 11011 11100 11101 00011 00101 11101 01111 10001
00000 10011 00110 01111 00110 10001 00110 10100 01111 10011 10101 01000 01110 01001 11001 11111 10101 01101 01101
10000 00101 10010 00110 10010 01100 01111 10101 10111 11000 01011 10110 11110 10001 01001 10101 10110 01010 01010
10110 10110 11001 11110 10011 11010 00001 01111 00101 00000 11001 01000 10100 10111 00100 10000 11011 00000 10101
00111 10100 11101 00011 11111 11111 01011 01001 11011 10100 10100 11101 11111 01101 01011 11101 00110 10100 00010
11100 01111 10100 01101 10001 00001 10011 10100 00000 11111 11010 01011 01111 10001 00100 10101 11101 01111 00100
01101 10101 00000 10011 11000 11111 11111 11101 10011 10100 01100 00000 00111 00100 01000 11100 00010 11110 00010
10011 10110 01001 00100 10011 10000 00110 10110 01110 01000 10101 11110 01111 11110 10110 01010 10000 01000 00100
10110 11101 10010 10101 11100 11101 10100 11001 10111 10010 10111 10110 10001 00000 00111 11000 00001 00100 11111
00010 11010 10010 10011 10111 11100 00011 11101 01000 01000 10100 10011 01111 00011 01000 11000 00010 01010 00111
00100 10110 00111 01001 10101 10001 11001 10001 10010 10001 00100 10101 01010 11110 11110 11001 01100 00010 00111
11011 01101 10111 00000 00111 00111 11010 00110 01010 00110 00011 10110 01110 01100 01110 01011 01011 11111 10000
11000 00001 10010 00100 01010 01100 11101 11011 10011 11110 01000 10101 00010 11101 00111 00010 00010 11100 11000
00100 01000 11000 01000 01010 11000 10111 01110 01011 11010 11010 01100 10011 10100 10110 00100 11101 01001 00111
11000 11001 11001 10001 01111 11000 11001 01101 01010 10011 10011 11110 11001 01100 00010 01110 01011 10011 10111
10101 11001 01110 01111 11110 00101 00110 01010 00110 00101 10000 10010 10101 01010 11100 11010 00011 11011 11000
00010 11000 11101 11011 01101 01001 00010 11110 01111 11010 01110 01001 11001 10010 11100 01001 11100 01011 11010
11101 01001 11000 10010 10010 10100 10000 10100 11101 01001 00111 01111 11010 01110 01001 11001 10000 00101 10111
11000 00001 01110 11110 00001 10111 01101 01110 01110 00111 00110 01110 00111 00100 11111 01001 10011 00100 01110
01100 11000 10011 10001 00101 10000 01110 01001 11001 10000 01000 01100 00011 00010 10110 10001 10110 10100 00001
01100 00010 01110 01011 10011 10111 10101 11001 01110 01111 11110 00101 00110 10111 10101 10111 10000 00110 01010
01111 11001 01111 11010 01110 01001 11001 10000 00001 00100 10001 11001 10001 10010 10101 11111 11000 01011 11101
00110 01011 10000 01100 01110 11000 01000 11000 01000 11110 01001 01001 00001 01001 11111 10101 10101 11101 11011
01011 11111 01110 01100 10011 11000 00010 10111 10110 10010 10000 10011 00011 01110 10100 10011 00111 01001 01000
01010 00100 10101 11100 10000 00000 10010 10000 10000 00101 10110 11000 00010 01011 11010 01111 11011 01001 01110
11001 01011 01111 11100 11001 10100 00111 10011 10100 01100 10110 11100 00001 01001 00010 10010 11101 10100 00101
10001 01000 10110 10101 10000 10001 00000 00111 00111 10011 10111 10011 10111 10010 00100 00100 10110 10001 10010
01010 00010 01100 01011 11000 00011 01101 01011 11111 01110 10111 10110 00010 00011 11111 00110 00010 10110 01101
11010 01110 01001 11001 10000 11101 01010 00010 11100 01100 10011 10111 11100 10110 00011 01110 01100 10011 11101
01100 00010 01110 01011 10011 10111 10101 11001 01110 01111 11110 00101 00110 01010 00110 00101 01001 11000 10000
10110 10100 10111 00111 11100 11010 00101 10000 10010 10101 01010 00101 01010 10100 10010 01010 00101 10000 11101
11011 01101 01001 00010 11110 01111 11010 01110 01001 10011 01111 10100 10100 11101 01010 00010 11100 01100 10100
10000 11000 00010 00011 00001 01111 10001 11001 10001 10000 00110 01010 01111 11001 11110 11001 01011 11100 10101
01010 10101 11101 01011 11101 00110 11111 10101 01011 00101 00111 10111 01010 01001 00000 01101 10000 00111 10110
01000 10101 01111 00110 10110 01110 01000 10101 11011 10000 01111 11000 11000 01100 00100 10010 00000 00100 00110
11100 10010 00010 10000 00101 10111 01110 01100 00011 11110 01000 11001 11101 00101 01111 01010 01010 10001 11100
00100 10001 11100 00001 01001 00010 10010 11101 10011 11001 11110 10000 11001 10111 01110 10010 01110 01110 01100
10110 11001 01110 11001 01000 01100 00110 11000 11101 01001 10011 11100 00000 10011 00101 10011 10100 11001 10001
11001 10001 01100 10010 00010 11110 00010 11101 01000 01000 11110 11100 10111 01010 01110 00111 11000 11100 00110
11110 00001 00010 00001 00010 00011 01110 01110 00000 00010 10111 10110 10010 10000 00101 11111 10000 00101 10110
11000 00010 10000 00010 10111 10110 10010 10000 00011 01000 11000 00010 01010 00111 00100 10110 00111 00100 00101
10000 10100 00011 00010 10110 00000 01110 11110 00001 11101 10011 11000 00101 10001 11000 11010 11010 11110 01011
00000 11101 10001 10001 01100 00010 00111 00001 10011 01000 11001 11010 01110 01010 01110 00110 01101 11011 01101
01001 00010 11110 01111 11010 01110 01001 11001 10000 00001 01000 11001 10000 11011 01100 00111 11010 01001 00110
11001 01000 10110 10101 10000 01110 01111 10100 11111 01011 11010 11001 11010 11110 01011 00000 11011 00100 10000
10010 00010 11101 01100 11100 10010 01111 11000 11000 01100 00010 01000 01100 10110 00101 10110 10101 10101 11001
10100 00010 00100 10111 00111 01010 10100 00011 10000 10100 00011 11111 00110 11100 10011 01110 11001 00000 01010
10010 10101 11010 11000 11001 01000 10010 10000 01111 11100 11111 01110 10000 11001 10011 11110 11000 01010 01100
10011 10100 10110 00100 01010 10010 11011 10010 00001 01111 01010 11100 10101 11100 10100 11001 00001 11110 01101
00100 11100 11010 11000 01100 00101 10111 01011 11010 01000 01001 11110 10000 01001 01000 11100 00010 01010 10010
10000 01111 10100 10011 11001 10111 00110 00000 00101 10100 10001 10110 10000 11101 00101 00010 01111 11000 11000
01100 00100 10010 00000 01010 10111 10011 01001 01011 01010 11110 01100 00001 11010 01010 01101 00101 11111 11100
00000 01101 01000 00011 11111 01111 10110 11011 10010 10010 11011 01010 01001 01000 01110 10011 01001 10001 11111
10011 10100 10011 01110 11100 11000 00100 01001 01100 00111 01001 00111 10100 10110 10001 11000 00001 01000 11001
10000 01010 00111 10100 10110 01100 11011 01110 10001 10011 01101 10000 00111 00111 10011 10111 10100 00011 10001
01110 10010 10001 00000 00010 10111 10110 10010 10000 00101 10000 10010 10101 01010 11100 11010 00011 11011 11000
00010 11000 11101 11101 11011 01011 11111 01110 01100 10011 11000 00010 10111 10110 10010 10000 10011 00110 01011
00111 00010 10011 01110 10010 01000 11000 11001 01000 01001 00011 01110 01110 00000 00010 10111 10110 10010 10000
00101 10010 00010 00100 01111 10011 10100 10010 00000 00001 01000 11001 10000 11011 01100 00111 11010 01001 00110
11001 01000 10110 10101 10000 01110 01111 10100 11111 01011 11010 11001 11010 11110 01011 00000 11011 00100 10000
10010 00010 11101 01100 11100 10010 01111 11000 11000 01100 00010 01000 01100 10110 00101 10110 10101 10101 11001
10100 00010 00100 10111 00111 01010 10100 00011 10000 10100 00011 11111 00110 11100 10011 01110 11001 00000 01010
10010 10101 11010 11000 11001 01000 10010 10000 01111 11100 11111 01110 10000 11001 10011 11110 11000 01010 01100
10011 10100 10110 00100 01010 10010 11011 10010 00001 01111 01010 11100 10101 11100 10100 11001 00001 11110 01101
00100 11100 11010 11000 01100 00101 10111 01011 11010 01000 01001 11110 10000 01001 01000 11100 00010 01010 10010
10000 01111 10100 10011 11001 10111 00110 00000 00101 10100 10001 10110 10000 11101 00101 00010 01111 11000 11000
01100 00100 10010 00000 01010 10111 10011 01001 01011 01010 11110 01100 00001 11010 01010 01101 00101 11111 11100
00000 01101 01000 00011 11111 01111 10110 11011 10010 10010 11011 01010 01001 01000 01110 10011 01001 10001 11111
10011 10100 10011 01110 11100 11000 00100 01001 01100 00111 01001 00111 10100 10110 10001 11000 00001 01000 11001
10000 01010 00111 10100 10110 01100 11011 01110 10001 10011 01101 10000 00111 00111 10011 10111 10100 00011 10001
01110 10010 10001 00000 00010 10111 10110 10010 10000 00101 10000 10010 10101 01010 11100 11010 00011 11011 11000
00010 11000 11101 11101 11011 01011 11111 01110 01100 10011 11000 00010 10111 10110 10010 10000 10011 00110 01011
00111 00010 10011 01110 10010 01000 11000 11001 01000 01001 00011 01110 01110 00000 00010 10111 10110 10010 10000
00101 10010 00010 00100 01111 10011 10100 10010 00000 00001 01000 11001 10000 11011 01100 00111 11010 01001 00110
11001 01000 10110 10101 10000 01110 01111 10100 11111 01011 11010 11001 11010 11110 01011 00000 11011 00100 10000
10010 00010 11101 01100 11101 01001 10011 11100 00000 10011 00101 10011 10100 11001 10001 11001 10001 01100 10010
00010 11110 00010 11011 01111 10110 11011 10010 10110 01011 00110 01110 00101 00010 01011 11000 01001 00010 01010
10010 11000 01100 00011 00010 10001 11010 10000 00101 10100 10010 01010 00101 10001 00001 00111 01001 00111 01111
00110 00100 10110 10110 11101 01110 11001 01000 10110 10101 10000 00101 10110 11000 00010 01011 11010 01111 11011
01001 01110 11001 01100 11100 11011 00011 11101 00100 10011 10111 01001 01110 11001 01000 10110 10101 10111 00000
10011 00001 11110 10111 00100 10110 10001 11001 11010 00001 10010 11111 00100 10100 01101 01110 11001 01000 10110
10101 10000 00110 01110 01111 10101 00111 00111 10110 01101 01110 00001 11010 00101 01011 00011 10001 01010 10010
00000 11010 00001 11000 11000 00101 10100 10101 00111 01101 00011 01010 11010 01010 01100 10011 11101 01011 11111
10101 10110 01110 01100 00011 01100 00010 00111 11011 01101 10111 00000 00111 00111 11010 00110 01010 00110 00011
10110 01110 01100 01110 01011 01111 00100 00110 10110 01011 00010 10010 00001 10100 10111 00010 00101 01001 11000

10000 10110 10100 10111 00111 00010 00110 01000 01001 01000 01001 00110 10110 00001 01001 10111 00011 11111 11000
10001 11111 10000 00000 00001 10011 00000 10100 10001 00000 10110 11001 01111 10010 11101 01110 11001 00111 00010
00110 00010 00100 01101 00100 11100 11010 11000 01100 00010 11100 10010 00011 11101 00011 01010 10010 11011 11001
10010 10110 11001 01000 11001 01000 01001 00110 11110 00001 00010 00001 00101 00000 11101 00101 01110 10010 01000
01001 11001 01010 11110 11011 01010 10101 10110 00110 01110 00110 01101 11001 10100 01110 10010 11101 11100 00010
01000 01100 00000 00101 10011 10111 10100 01110 10010 00000 01010 10111 10110 10010 10000 10011 00000 00001 00001
00111 01001 00111 01001 01110 00010 11110 10110 01100 00111 10000 00110 00001 10110 01001 01110 01000 11111 11101
01011 10000 01011 10110 01110 10101 10000 01001 00101 10111 01001 11010 01010 10110 01010 00110 00011 00010 10001
11010 10000 00101 10100 10010 01010 00110 00110 00101 00001 01010 01010 11100 00000 00101 10110 00110 11110 10010
10011 11010 00110 10111 01001 11011 10000 11101 00101 10011 01110 11110 00010 11101 00111 00010 10000 00000 11001
00000 00101 00010 11101 00111 00010 00010 11100 10100 11110 01101 11000 00000 11110 00010 10010 01011 01101 10000
11100 10010 01111 11000 11000 01100 00010 01000 01100 00000 00101 01100 00000 11101 01001 00111 01111 11010 01110
01001 11001 01111 10010 00100 11101 01010 00010 11100 01100 10100 10000 11000 00010 00011 11111 10000 00101 11111
01111 11110 10111 10111 01110 01010 00000 01101 11010 11101 10000 11101 01010 00010 11100 01100 10110 01010 10100
01111 11111 10000 00000 00000 10100 01111 11111 11000 11010 00000 00110 11000 11001 10111 11101 10111 10000 10101
00010 01110 10111 00100 10001 11001 10001 10010 10001 00011 00000 00001 11000 10010 01100 00001 11011 11010 01101
01110 00111 11101 11001 10111 11100 11000 10100 10001 10110 10111 10110 10111 10000 11001 11010 01111 00110 11101
00111 00111 10111 00100 01011 11001 01111 10010 00011 10001 00010 00001 00111 11000 10110 01001 00110 01111 10001
01111 01101 00101 11010 11001 01110 00110 11000 11010 00100 10001 00000 01001 00010 10100 10001 10110 10010 11010
10111 01001 11011 01111 01001 01110 11011 00100 10110 10111 10101 10111 10010 10110 01101 01001 11001 01001 01001
11111 10110 10010 10000 00110 11110 00001 00010 00001 11100 11010 00110 01011 00111 00010 10011 01110 10010 01000
11000 11001 01000 01001 00011 01110 01110 00000 00010 10111 10110 10010 10000 00101 11111 10000 00101 10110 11000
00010 01011 11010 01111 11011 01001 01110 11001 01100 00001 01000 11001 10000 11011 01100 00111 11010 01001 00110
11001 01000 10110 10101 01111 00100 10111 10111 00001 00010 11110 01011 00010 01000 11101 01001 00010 11011 00101
00001 10111 10111 00111 10111 11111 11100 00001 10001 10111 10111 00101 11111 00001 11001 01110 10010 10001 11111
11000 10001 00010 01000 00110 01010 00110 00110 10110 11111 11000 00010 01100 01100 11010 00011 01110 10001 11011
00100 01001 11101 10001 00001 01010 10000 10010 10000 01111 01000 10100 10011 10100 10011 00000 00001 10011 11111
11100 11111 00010 10110 01011 01010 00110 11000 11010 00001 11010 11111 00111 00010 01100 01100 01000 01101 11011
01101 01001 00010 11110 01111 11010 01110 01001 11001 10000 00111 00111 10011 10111 10100 00011 10010 11011 10011
01010 01100 00010 01001 10110 10110 10111 10101 10111 01100 00010 00001 00101 01010 10100 10010 01010 00101 10000
01001 01001 01110 00100 01100 01110 01001 00101 00111 10110 11000 00101 11011 10011 11110 01000 10101 00010 11101
00111 00010 00010 11100 11000 00100 01000 11000 01000 01010 11000 10111 01110 01011 11010 11010 01100 10011 10100
10110 00100 11101 01001 00111 11000 11001 11001 10001 01111 11000 11001 01101 01010 10011 10011 11110 11001 01100
00010 01110 01011 10011 10111 10101 11001 01110 01111 11110 00101 00110 01010 00110 00101 10000 10010 10101 01010
11100 11010 00011 11011 11000 00010 11000 11100 01001 01101 10101 01101 00010 10110 00100 11110 01100 00010 11000
01001 00010 01010 10010 11010 11101 10000 00001 01000 11001 10000 01010 01001 11011 11001 10111 10110 11101 00010
00110 11110 00001 00010 00001 00101 00000 01111 11000 11000 01100 00100 11010 00001 11000 11000 00101 10000 01000
11001 01110 10011 01010 10101 11011 00010 00100 11010 00011 01011 01011 11111 11100 11111 10111 01001 00010 00100
11010 00001 11000 11000 01001 11101 10111 11110 11100 11001 11111 11000 10110 01010 01010 01110 00110 01101 11111
10111 10100 00100 11010 00001 11000 11000 00101 10000 10100 10101 01000 00001 00001 01000 10100 00100 01110 01010
10000 00011 11111 00110 00010 10110 01101 11010 01110 01001 11001 10000 00111 11000 00001 00100 11111 00010 11010
10010 10011 10111 11100 00011 11101 01000 01000 10100 10011 01111 00011 01110 10001 11011 00100 01001 11101 10001
00001 01010 10000 10010 10000 01111 01000 10100 10011 10100 10011 00000 11011 01010 01111 01101 00011 01010 00101
00110 01101 10001 10110 11110 01001 10011 01110 01011 10110 10000 11000 10011 11110 10110 01110 01000 10010 10100
10010 10100 10000 10100 00111 00111 10011 10111 10100 00011 01011 01010 01000 11001 10000 10000 00010 00111 00101
01111 00110 01110 00111 00110 10010 11000 00100 01001 00000 00010 10111 10110 10001 00000 00111 01001 10000 00110
01001 11010 11010 01100 10011 11101 01011 00011 10001 01010 10010 00000 11010 00001 11000 11000 00101 10100 10101
00111 01101 00011 01010 11010 01010 01100 10011 11101 01100 00010 01110 01011 10011 10111 10101 11001 01110 01111
11110 00101 00110 10111 10101 10111 01011 11111 10101 10110 01110 01100 00011 01011 10110 10101 01001 11001 10011
01110 10001 10011 11000 10000 01000 11000 01000 01010 00111 01110 01111 11001 00001 00010 11000 11101 01001 10011
11100 00000 10011 00101 10011 10100 11001 10001 11001 10001 01100 10010 00010 11110 00010 11100 11011 00100 00010
11110 01100 10101 01010 10100 10010 01010 00110 01011 10001 00001 00111 01001 00111 11110 11001 10010 11011 10011
01010 01100 00010 01001 10110 10110 10111 10101 10111 01100 00010 00001 00101 01010 10100 10010 01010 00101 01011
01001 01010 10001 00110 11101 11001 10100 01110 10010 11101 11100 00010 01000 01100 10110 11101 00101 01111 00100
10111 10111 00001 00010 00001 10010 01111 11000 11011 10101 01101 10110 10101 10101 10011 10100 01100 10100 01011
01010 10101 11011 01110 00110 01001 01001 00000 11100 01001 01101 01101 01111 00100 11111 11000 00101 01101 01111
00110 10110 01011 00010 10010 00001 10110 01000 01100 00101 01001 11000 10000 10110 10100 10111 00111 11100 11010
00110 10110 00001 01001 10111 00011 11111 11000 10001 11111 10000 00110 11110 00001 00010 00001 00101 11101 01011

ДОДАТОК Е

(обов’язковий)
Реалізація шифру зсуву розрядності r=6

Розбиття послідовного коду закодованого оптимальним нерівномірним кодом тексту на чотирьохрозрядні коди (T'_{CH6}):

110001 100110 001001 010110 001001 011010 000100 111000 110011 111000 110110 111001 001110 001010 101011 110111
001101 101110 100001 110100 000111 101101 111101 010110 101001 110110 001010 111110 100000 100010 110011 111011
100101 111110 011011 101100 010101 100111 010011 001001 001000 011111 111101 111100 011100 000010 100000 010011
010110 111001 110001 010001 000011 011110 100111 011110 110001 001110 110110 001110 110101 011011 110101 101011
110110 010011 101001 111000 110001 111110 100010 011101 100000 001001 011100 001010 110001 100111 110101 101111
010110 100011 011100 111000 110011 100001 000000 100101 011110 100110 101110 100101 111100 010001 000000 110000
001101 101110 100011 100010 011000 011100 111000 011110 101001 011011 110101 101011 111110 111100 100111 000011
111111 111000 111000 110011 101101 001000 100111 001100 111011 111011 110101 101010 011101 111101 001001 111110
001100 000011 111101 000100 111100 111000 011000 111001 101011 110100 111010 000011 010100 101101 111010 110101
010101 101111 101001 111000 011111 101000 000001 110100 111111 010001 000000 100100 111000 111000 111010 011111
100110 111011 110101 110000 101110 000011 000011 111000 100111 011101 101111 011111 001101 011100 100010 011100
110010 100111 111001 101110 111101 010101 010111 001110 100110 100110 001100 110111 001001 010011 100110 110001
011100 110110 001001 001000 111011 000010 011001 110001 111011 100010 110010 101010 110011 011101 111010 101011
011000 101000 001111 101110 110110 111100 100110 100010 011110 011100 001100 011100 110101 111010 011101 000001
101010 010111 110011 010111 001010 101011 011111 010011 110101 001111 110011 011101 111010 101001 100100 110110
111101 011010 111110 100010 011101 100000 001001 011100 001010 110001 100111 110101 101111 010110 100011 011100
111000 110011 100010 011010 110100 001000 000100 000010 010011 100011 100011 101001 111110 011011 101111 010111
000010 001001 111001 110000 110001 110011 010111 101001 110100 000110 101001 011011 110101 101010 101011 011111
010011 110000 111111 010000 000011 101001 111110 100001 010000 001111 101100 101100 110001 100110 001110 110000
100110 011100 011110 111000 101100 101010 101100 110111 011110 101010 110110 001010 110001 100110 001110 011000
111001 011101 001010 011101 110110 101110 111100 111100 011000 011101 010001 001010 000001 001001 111100 110110
011101 000001 000101 100111 110111 001011 111100 110111 011110 101010 001001 111001 110000 110001 110001 110000
001110 110100 010011 100011 100000 010100 010011 110011 100001 100011 100110 101111 010011 101000 001101 010010
110111 101011 010100 111011 101101 011101 111001 111000 110000 001111 110101 010101 101111 101001 111010 100111
111001 101110 111101 010101 010001 000000 100100 111000 111000 111010 011111 100110 111011 000101 001100 111001
000100 111100 111000 011000 111001 101011 110100 111010 000011 010100 101101 111010 110101 010101 101111 101001
111000 011111 101000 000001 110100 111111 010000 101000 000010 010110 010011 010100 000101 001100 111000 111101
110001 011001 010101 011001 101101 101110 101010 001011 110111 001101 101110 100000 010101 101001 110111 101100
010100 110111 001010 100101 011000 011011 100111 101010 101011 100100 111000 101000 000110 110111 110000 100101
010100 011110 111110 110000 010100 110110 000010 011001 110001 111011 100010 110001 111000 011101 011111 011100
100111 011110 011100 111000 111011 111101 001111 110011 011000 101011 111010 001001 110110 000000 100101 110000
101011 000110 011111 010110 111101 011010 001101 110011 100011 001110 001001 101011 010001 100001 111000 111110
011011 001110 100001 010110 001100 110001 110011 000111 010001 001110 011001 010011 111100 110111 011110 101010
100010 010101 010101 101111 101001 111010 100111 111001 101110 111101 010100 001101 111010 011101 111011 000100
111011 011000 100101 010101 011100 101000 001111 101100 101100 110001 100110 000101 100011 101010 101011 011101
111111 111100 011100 000010 100010 101101 011010 001001 110110 000110 110000 110111 110111 111001 101111 100110
101110 010000 001001 001110 001110 001110 100111 111001 101110 111101 010100 110011 011111 010101 000001 000101
100111 110111 001011 111100 110111 011110 101010 110011 101001 100100 100100 001111 111110 111110 001110 000001
010000 001001 101011 011100 111000 101000 100001 101111 010011 101111 011000 100111 011011 000111 011010 101101
111010 110101 111011 001001 110100 111100 011000 111111 001010 001010 111001 010000 010001 011000 011100 010011
111100 010101 111101 111101 011111 111011 111001 101101 111010 110100 000010 010010 011101 011111 011000 000111
110101 111100 011100 011001 110110 100101 111101 110000 010111 001101 010001 111000 011101 000001 110011 111000
110000 111001 001001 000011 111111 101100 010101 011100 100001 111101 101011 100001 110101 011100 110100 001001
110111 110111 101011 010011 001110 001111 011100 010110 010101 010110 011011 011011 101010 001001 010001 111010
011101 111011 000101 001101 110010 101111 111001 100001 110100 000111 100011 100010 011011 111011 111001 001010
001000 000100 101100 100110 101000 001001 010011 011000 001011 110111 000000 111101 110011 011001 111000 011101
011000 100110 001100 111000 100110 000111 101010 010111 010001 011000 111001 100110 011101 110110 111010 011001
101111 101010 101111 011001 100111 011100 010000 010011 101101 100010 010101 010110 001100 110001 110011 001010
001011 010011 101111 011000 101001 111110 011011 101111 010101 010101 011011 111010 011110 000111 111010 000000
011101 001111 110100 010000 100000 010000 001001 001110 001110 001110 100111 111001 101110 111101 011100 001011
100000 110000 111110 001001 110111 011011 110111 110011 010111 001000 100111 001100 101001 111110 011011 101111
010101 010101 110011 101001 101001 100011 001101 110010 100110 011011 111010 101000 001000 101100 111110 111001
011111 100110 111011 110101 010110 011101 001100 100100 100001 111111 110111 110001 110000 001010 000001 001101
011011 100111 000101 000100 010011 101100 000001 001011 100001 010110 001100 111110 101101 111010 110100 011011
100111 000110 011100 000101 001101 100000 101111 101110 000010 111001 101010 001111 000011 101011 100011 101111
101111 110110 001010 000011 110110 111111 010101 010110 011011 101111 010101 011000 110011 000111 001100 101000
101101 001110 111101 100010 100111 111001 101110 111101 010101 010101 101111 101001 111000 011111 101000 000001
110100 111111 010001 000010 000001 000000 100100 111000 111000 111010 011111 100110 111011 110101 110000 101110
000011 000011 111000 100111 011101 101111 011111 001101 011100 100010 011100 110010 100111 111001 101110 111101
010101 010111 001110 100110 100110 001100 110111 001010 011001 101111 101010 100000 100010 110011 111011 100101
111110 011011 101111 010101 011001 110100 110010 010010 000111 111111 011111 000111 000000 101000 000100 110101
101110 011100 010100 010001 001110 110000 000100 101110 000101 011000 110011 111010 110111 101011 010001 101110
011100 011001 110000 101000 100101 011110 111000 000111 101110 011011 001111 000011 101010 011101 110110 101110

111100 111100 011000 011101 011011 010111 001101 011100 101111 011001 100111 011100 010000 010011 101101 100010
010101 001010 011011 000001 011100 110110 001011 101111 110101 001011 100010 100111 111001 100001 110101 100111
010011 001001 001000 011111 111101 100010 011100 001101 110100 000010 010000 111110 111000 001111 010111 110111
111001 101111 100110 101110 010110 001100 110001 110011 001010 001010 001001 010100 111011 101101 011101 111001
111000 110000 001111 110101 101101 011100 110101 110010 111101 100110 011101 110001 000001 001110 110110 001001
010101 011000 111001 100110 011101 110010 101111 111001 101110 111101 011100 000101 001100 011001 100011 100110
001110 100110 011100 011110 111000 101100 101010 101100 110110 110111 010011 011010 010000 010100 001010 110000
110111 001111 010101 010111 001010 111000 111011 111011 111101 101111 010110 100000 111101 101111 110101 010101
100110 111011 110101 101011 010100 011001 111011 110000 100101 010101 101101 011000 111011 111000 111110 101111
100011 100000 010101 000100 101011 000100 110001 100111 000100 110000 111101 010010 111110 001010 101000 111000
100110 000111 001110 000111 001000 111001 011101 001010 001100 111101 111000 010010 101010 000110 111101 001110
111101 100010 011101 101100 010010 101010 100010 010100 010011 100110 010100 111111 001101 110111 101010 010111
010010 001001 111001 110000 110001 110011 010111 101001 110100 000100 101010 101000 100101 000001 111100 111001
001101 111001 011001 011111 000101 010100 010011 110011 100001 100011 101101 111110 011010 000100 101010 010100
101110 011010 000100 111011 111100 101010 111110 111110 111000 001011 100101 011101 000111 100111 110001 001101
101111 010110 101111 011001 000001 010011 011101 101111 000100 110000 001111 110010 100110 110000 010111 101110
000001 111011 100110 110110 111110 011011 111001 010111 110111 111010 001011 000100 110001 100111 000100 110000
111101 010010 111010 001011 000111 001100 110011 101110 110111 001011 101001 011010 100100 100101 011111 111101
001101 011111 011111 101001 101100 010101 110001 111100 110110 110111 011111 111111 000111 000000 101000 111010
011000 000100 111011 111001 101011 100101 111101 110010 011101 111001 110011 100010 011011 101111 010101 011000
110011 000111 001100 000111 111010 111000 001100 001111 100010 011101 110110 111101 111100 110101 110010 001001
110011 001010 011111 100110 111011 110101 010100 010010 101010 101101 111101 001111 000011 111101 000000 001110
100111 111010 001001 100110 111110 101010 000010 001011 001111 101110 010111 111001 101110 111101 010100 010011
110011 100001 100011 100011 100000 011101 101000 100111 000111 000000 101000 110111 001110 001100 111000 010000
001001 101011 011100 111000 101000 100001 101111 010011 101111 011000 100111 011011 000111 011010 101101 111010
110101 111011 001001 110100 111100 011000 111111 010001 001110 110000 000100 101110 000101 011000 110011 111010
110111 101011 010001 101110 011100 011001 110000 010100 110110 000010 000001 001000 011111 011100 000111 101011
111011 111100 110111 110010 001100 001111 000110 001011 011001 000000 100100 111000 111000 111010 011111 100110
111011 110101 010110 001100 110001 110011 001010 001011 100000 110000 111110 001001 110111 011011 110111 110011
010111 001000 100111 001100 101001 111110 011011 101111 010101 010101 110011 101001 101001 100011 001101 110010
100110 011011 111010 101000 000110 000001 101101 110100 011100 010011 000011 100111 000011 110101 001011 011110
101101 011111 110111 100100 111000 011111 111111 000111 000110 011101 101001 000100 111001 100111 011111 011110
101101 010011 101111 101001 001111 110001 100000 011111 101000 100111 100111 000011 000111 001101 011110 100111
010000 011010 100101 101111 010110 101010 101101 111101 001111 000011 111101 000000 001110 100111 111010 000101
110100 101101 010010 001111 101101 001000 111000 100111 111010 111000 111011 111011 111111 000101 010100 110011
011111 010101 011110 111000 000111 101110 011011 000100 011101 011000 110011 000111 001100 101000 101101 001110
111101 100010 100111 111001 101110 111101 010101 010101 101111 101001 111000 011111 101000 000001 110100 111111
010001 000010 000001 000000 100100 111000 111000 111010 011111 100110 111011 110101 110000 101110 000011 000011
111000 100111 011101 101111 011111 001101 011100 100010 011100 110010 100111 111001 101110 111101 010101 010111
001110 100110 100110 001100 110111 001010 011001 101111 101010 100000 100010 110011 111011 100101 111110 011011
101111 010101 011001 110100 110010 010010 000111 111111 011111 000111 000000 101000 000100 110101 101110 011100
010100 010001 001110 110000 000100 101110 000101 011000 110011 111010 110111 101011 010001 101110 011100 011001
110000 010100 000011 111010 010010 010000 111101 010010 111001 010110 110110 001101 110110 001001 110000 110111
010111 101110 000001 111011 100110 110110 111110 011011 111001 101011 100101 100011 001100 011100 110010 100010
000011 110110 111110 101011 010100 111011 000101 010100 010111 001101 100010 111011 111101 010010 111000 101001
111110 011011 101100 010101 100011 101010 101011 011101 111111 111100 011100 000010 100000 100010 110011 111011
100101 111110 011011 101111 010101 011001 110100 110010 010010 000111 111111 011111 000111 000000 101000 100111
100111 000011 000111 001101 011110 100111 010000 011010 100101 111100 110101 110010 000001 001101 011011 100111
000101 000100 001101 111010 011101 111011 000100 111011 011000 111011 010101 101111 010110 101111 011001 001110
100111 100011 000111 111010 001001 110110 000000 100101 110000 101011 000110 011111 010110 111101 011010 001101
110011 100011 001110 000100 000010 010011 100011 100011 101001 111110 011011 101111 010111 000010 110001 100110
001110 011000 001111 110101 110000 011000 011111 000100 111011 101101 111011 111001 101011 100100 010011 100110
010100 111111 001101 110111 101010 100000 111001 111101 100101 100010 111101 100110 011101 110001 000001 001110
110110 001110 110001 001010 101000 100111 100111 000011 000111 001101 011110 100111 010000 011010 100101 101111
010110 101100 011001 100011 100110 000011 111101 000000 100110 101101 110011 100010 100001 011101 001010 010101
000100 100100 111010 101010 010101 000101 111011 100000 011110 111001 101101 101101 100010 101001 110111 011010
111011 110011 110001 100000 011111 101011 110110 011001 110111 000100 000100 111011 011000 100101 010101 100011
001100 011100 110010 100001 010000

Шифрований текст $T_{E5(4)}$, отриманий з використанням шифру зсуву кратності w=5 до тексту T'_{CH6}:

110110 101011 001110 011011 001110 011111 001001 111101 111000 111101 111011 111110 010011 001111 110000 111100
010010 110011 100110 111001 001100 110010 000010 011011 101110 111011 001111 000011 100101 100111 111000 000000
101010 000011 100000 110001 011010 101100 011000 001110 001101 100100 000010 000001 100001 000111 100101 011000
011011 111110 110110 010110 001000 100011 101100 100011 110110 010011 111011 010011 111010 100000 111010 110000
111011 011000 101110 111101 110110 000011 100111 100010 100101 001110 100001 001111 110110 101100 111010 110100
011011 101000 100001 111101 111000 100110 000101 101010 100011 101011 110011 101010 000001 010110 000101 110101
010010 110011 101000 100111 011101 100001 111101 100011 101110 100000 111010 110000 000011 000001 101100 001000
000100 111101 111101 111000 110010 001101 101100 010001 000000 000000 111010 101111 100010 000010 001110 000011
010001 001000 000010 001001 000001 111101 011101 111110 110000 111001 111111 001000 011001 110010 111111 111010

011010 110100 101110 111101 100100 101101 000110 111001 000100 010110 000101 101001 111101 111101 111111 100100
101011 000000 111010 110101 110011 001000 001000 111101 101100 100010 110100 100100 010010 100001 100111 100001
110111 101100 111110 110011 000010 011010 011100 010011 101011 101011 010001 111100 001110 011000 101011 110110
100001 111011 001110 001101 000000 000111 011110 110110 000000 100111 110111 101111 111000 100010 111111 110000
011101 101101 010100 110011 111011 000001 101011 100111 100011 100001 010001 100001 111010 111111 100010 000110
101111 011100 111000 011100 001111 110000 100100 011000 111010 010100 111000 100010 111111 101110 101001 111011
000010 011111 000011 100111 100010 100101 001110 100001 001111 110110 101100 111010 110100 011011 101000 100001
111101 111000 100111 011111 111001 001101 001001 000111 011000 101000 101000 101110 000011 100000 110100 011100
000111 001110 111110 110101 110110 111000 011100 101110 111001 001011 101110 100000 111010 101111 110000 100100
011000 110101 000100 010101 001000 101110 000011 100110 010101 010100 110001 110001 110110 101011 010011 110101
101011 100001 100011 111101 110001 101111 110001 111100 100011 101111 111011 001111 110110 101011 010011 011101
111110 100010 001111 100010 111011 110011 000001 000001 011101 100010 010110 001111 000110 001110 000001 111011
100010 000110 001010 101100 111100 010000 000001 111100 100011 101111 001110 111110 110101 110110 110110 110101
010011 111001 011000 101000 100101 011001 011000 111000 100110 101000 101011 110100 011000 101101 010010 010111
111100 110000 011001 000000 110010 100010 111110 111101 110101 010100 111010 011010 110100 101110 111111 101100
111110 110011 000010 011010 010110 000101 101001 111101 111101 111111 100100 101011 000000 001010 010001 111110
001001 000001 111101 011101 111110 110000 111001 111111 001000 011001 110010 111111 111010 011010 110100 101110
111101 100100 101101 000110 111001 000100 010101 101101 000111 011011 011000 011001 001010 010001 111101 000010
110110 011110 011010 011110 110010 110011 101111 010000 111100 010010 110011 100101 011010 101110 111100 110001
011001 111100 001111 101010 011101 100000 101100 101111 110000 101001 111101 101101 001011 111100 110101 101010
011001 100011 000011 110101 011001 111011 000111 011110 110110 000000 100111 110110 111101 100010 100100 100001
101100 100011 100001 111101 000000 000010 010100 111000 011101 110000 111111 001110 111011 000101 101010 110101
110000 001011 100100 011011 000010 011111 010010 111000 101000 010011 001110 110000 010110 100110 111101 000011
100000 010011 100110 011011 010001 110110 111000 001100 010110 010011 011110 011000 000001 111100 100011 101111
100111 011010 011010 110100 101110 111111 101100 111110 110011 000010 011001 010010 111111 100010 000000 001001
000000 011101 101010 011010 100001 101101 010100 110001 110001 110110 101011 001010 101000 101111 110000 100010
000100 000001 100001 000111 100111 110010 011111 001110 111011 001011 110101 111100 111100 111110 110100 101011
110011 010101 001110 010011 010011 010011 101100 111110 110011 000010 011001 111000 100100 011010 000110 001010
101100 111100 010000 000001 111100 100011 101111 111000 101110 101001 101001 010100 000011 000011 010011 000110
010101 001110 110000 100001 111101 101101 100110 110100 011000 110100 011101 101100 100000 001100 011111 110010
111111 111010 000000 001110 111001 000001 011101 000100 001111 001111 111110 010101 010110 011101 100001 011000
000001 011010 000010 000010 100100 000000 111110 110010 111111 111001 000111 010111 100010 100100 011101 001100
111010 000001 100001 011110 111011 101010 000010 110101 011100 010010 010110 111101 100010 000110 111000 111101
110101 111110 001110 001000 000100 110001 011010 100001 100110 000010 110000 100110 111010 100001 111001 001110
111100 111100 110000 011000 010011 010100 100001 011011 011010 011011 100000 100000 101111 001110 010110 111111
100010 000000 001010 010010 110111 110100 111110 100110 111001 001100 101000 100111 100000 000000 111110 001111
001101 001001 110001 101011 101101 001110 011000 011101 010000 111100 000101 000010 111000 011110 111101 100010
011101 101011 010001 111101 101011 001100 101111 011100 010110 011101 111110 101011 100010 111011 111111 011110
110100 101111 110100 011110 101100 100001 010101 011000 110010 100111 011010 011011 010001 110110 111000 001111
010000 011000 110100 011101 101110 000011 100000 110100 011010 011010 100000 111111 100011 001100 111111 000101
100010 010100 111001 010101 100101 010101 001110 010011 010011 010011 101100 111110 110011 000010 100001 010000
100101 110101 000011 001110 111100 100000 111100 111000 011100 001101 101100 010001 101110 000011 100000 110100
011010 011010 111000 101110 101110 101000 010010 110111 101011 100000 111111 101101 001101 110001 000011 111110
100100 101011 000000 111010 011011 100010 010001 101001 100110 000100 111100 110110 110101 001111 000110 010010
100000 101100 001010 001001 011000 110001 000110 010000 100110 011011 010001 000011 110010 111111 111001 100000
101100 001011 100001 001010 010010 100101 110100 110011 000111 111110 101111 010100 001000 110000 101000 110100
110100 111011 001111 001000 111011 000100 011010 011011 100000 110100 011010 011101 111000 001100 010001 101101
110010 010011 000010 100111 101100 111110 110011 000010 011010 011010 110100 101110 111101 100100 101101 000110
111001 000100 010110 000111 000110 000101 101001 111101 111101 111111 100100 101011 000000 111010 110101 110011
001000 001000 111101 101100 100010 110100 100100 010010 100001 100111 100001 110111 101100 111110 110011 000010
011010 011100 010011 101011 101011 010001 111100 001111 011110 110100 101111 100101 100111 111000 000000 101010
000011 100000 110100 011010 011110 111001 110111 010111 001100 000100 100100 001100 000101 101101 001001 111010
110011 100001 011001 010110 010011 110101 001001 110011 001010 011101 111000 111111 111100 110000 010110 110011
100001 011110 110101 101101 101010 100011 111101 001100 110011 100000 010100 001000 101111 100010 111011 110011
000001 000001 011101 100010 100000 011100 010010 100001 110100 011110 101100 100001 010101 011000 110010 100111
011010 001111 100000 000110 100001 111011 010000 110100 111010 010000 100111 101100 111110 100110 111010 101100
011000 001110 001101 100100 000010 100111 100001 010010 111001 000111 010101 000011 111101 010100 011100 111100
111110 110100 101011 110011 011011 010001 110110 111000 001111 001111 001110 011001 000000 110010 100010 111110
111101 110101 010100 111010 110010 100001 111010 110111 000010 101011 100010 110110 000110 010011 111011 001110
011010 011101 111110 101011 100010 110111 110100 111110 110011 000010 100001 001010 010001 011110 101000 101011
010011 101011 100001 100011 111101 110001 101111 110001 111011 111100 011000 011111 010101 011001 001111 110101
111100 010100 011010 011100 001111 111101 000000 000000 000010 110100 011011 100101 000010 110100 111010 011010
101011 000000 111010 110000 011001 011110 000000 110101 101010 011010 110010 011101 000000 111101 000011 110100
101000 100101 011010 001001 110000 001001 110110 101100 001001 110101 000010 010111 000011 001111 101101 111101
101011 001100 010011 001100 001101 111110 100010 001111 010001 000010 111101 010111 101111 001011 000010 010011
000010 100111 100010 110001 010111 101111 100111 011001 011000 101011 011001 000100 010010 111100 101111 011100
010111 001110 111110 110101 110110 111000 011100 101110 111001 001001 101111 101101 101010 000110 000001 111110
010010 111110 011110 100100 001010 011001 011000 111000 100110 101000 110010 000011 011111 001001 101111 011001
110011 011111 001001 000000 000001 101111 000011 000011 111101 010000 101010 100010 001100 101100 110110 010010
110100 011011 110100 011110 000110 011000 100010 110100 001001 110101 010100 110111 101011 110101 011100 110011
000110 000000 101011 111011 000011 100000 111110 011100 111100 111111 010000 001001 110110 101100 001001 110101
000010 010111 111111 010000 001100 010001 111000 110011 111100 010000 101110 011111 101001 101010 100100 000010

010010 100100 100100 101110 110001 011010 110110 000001 111011 111100 100100 000100 001100 000101 101101 111111
011101 001001 000000 111110 110000 101010 000010 110111 100010 111110 111000 100111 100000 110100 011010 011101
111000 001100 010001 001100 111111 111101 010001 010100 100111 100010 111011 000010 000001 111010 110111 001110
111000 001111 100100 101011 000000 111010 011001 010111 101111 110010 000010 010100 001000 000010 000101 010011
101100 111111 001110 101011 000011 101111 000111 010000 010100 110011 011100 111110 110011 000010 011001 011000
111000 100110 101000 101000 100101 100010 101101 101100 001100 000101 101101 111100 010011 010001 111101 010101
001110 110000 100001 111101 101101 100110 110100 011000 110100 011101 101100 100000 001100 011111 110010 111111
111010 000000 001110 111001 000001 011101 000100 010110 010011 110101 001001 110011 001010 011101 111000 111111
111100 110000 010110 110011 100001 011110 110101 011001 111011 000111 000110 001101 100100 100001 001100 110000
000000 000001 111100 110111 010001 010100 001011 010000 011110 000101 101001 111101 111101 111111 100100 101011
000000 111010 011011 010001 110110 111000 001111 010000 100101 110101 000011 001110 111100 100000 111100 111000
011100 001101 101100 010001 101110 000011 100000 110100 011010 011010 111000 101110 101110 101000 010010 110111
101011 100000 111111 101101 001011 000110 110010 111001 100001 011000 001000 101100 001000 111010 010000 100011
110010 100100 111100 101001 111101 100100 000100 001100 001011 100010 101110 001001 111110 101100 100100 100011
110010 011000 110100 101110 010100 110110 100101 100100 101101 101100 101100 001000 001100 010010 100011 101100
010101 011111 101010 110100 011011 101111 110010 000010 010100 001000 000010 000101 010011 101100 111111 001010
111001 110010 010111 010100 110010 001101 111101 101100 111111 111101 000000 000000 000100 001010 011001 111000
100100 011010 100011 111101 001100 110011 100000 001001 100010 011101 111000 001100 010001 101101 110010 010011
000010 100111 101100 111110 110011 000010 011010 011010 110100 101110 111101 100100 101101 000110 111001 000100
010110 000111 000110 000101 101001 111101 111101 111111 100100 101011 000000 111010 110101 110011 001000 001000
111101 101100 100010 110100 100100 010010 100001 100111 100001 110111 101100 111110 110011 000010 011010 011100
010011 101011 101011 010001 111100 001111 011110 110100 101111 100101 100111 111000 000000 101010 000011 100000
110100 011010 011110 111001 110111 010111 001100 000100 100100 001100 000101 101101 001001 111010 110011 100001
011001 010110 010011 110101 001001 110011 001010 011101 111000 111111 111100 110000 010110 110011 100001 011110
110101 011001 001000 111111 010111 010101 000010 010111 111110 011011 111011 010010 111011 001110 110101 111100
011100 110011 000110 000000 101011 111011 000011 100000 111110 110000 101010 101000 010001 100001 110111 100111
001000 111011 000011 110000 011001 000000 001010 011001 011100 010010 100111 000000 000010 010111 111101 101110
000011 100000 110001 011010 101000 101111 110000 100010 000100 000001 100001 000111 100101 100111 111000 000000
101010 000011 100000 110100 011010 011110 111001 110111 010111 001100 000100 100100 001100 000101 101101 101100
101100 001000 001100 010010 100011 101100 010101 011111 101010 000001 111010 110111 000110 010010 100000 101100
001010 001001 010010 111111 100010 000000 001001 000000 011101 000000 011010 110100 011011 110100 011110 010011
101100 101000 001100 111111 001110 111011 000101 101010 110101 110000 001011 100100 011011 000010 011111 010010
111000 101000 010011 001001 000111 011000 101000 101000 101110 000011 100000 110100 011100 000111 110110 101011
010011 011101 010100 111010 110101 011101 100100 001001 000000 110010 000000 111110 110000 101001 011000 101011
011001 000100 010010 111100 101111 100101 111110 000010 101010 100111 000010 101011 100010 110110 000110 010011
111011 010011 110110 001111 101101 101100 101100 001000 001100 010010 100011 101100 010101 011111 101010 110100
011011 110001 011110 101000 101011 001000 000010 000101 101011 110010 111000 100111 100110 100010 001111 011010
001001 101001 111111 101111 011010 001010 000000 100101 100011 111110 110010 110010 100111 101110 111100 011111
000000 111000 110110 100101 100100 110000 111011 011110 111100 001001 001001 000000 011101 101010 011010 101000
010001 100001 110111 100110 010101

Tue Nov 24 21:39:31 EET 2020, Муляра І.В, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 10%**

ID: 81181 Название: Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем Добавлено в БД: 2020-11-24 Авторы: Гончар Р.М. Руководители: Чешун В.М. Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	110099	1575	1032 (1%)	19 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

UNICHECK

User name:
Kafedra kiberbezpeky

Check ID:
1005325787

Check date:
02.12.2020 10:39:30 EET

Check type:
Doc vs Internet + Library

Report date:
02.12.2020 10:41:42 EET

User ID:
100005590

File name: **Гончар робота на Unicheck(1)**

Page count: **101** Word count: **19307** Character count: **143930** File size: **1.50 MB** File ID: **1005448733**

6.26% Matches

Highest match: **0.83%** with Internet source (**https://msn.khnu.km.ua/pluginfile.php/400615/mod_resource/content/1/%D0%9F%D.**

6.26% Internet sources 401 ..... Page 103

0.37% Library sources 30 ..... Page 107

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters 177

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем

Автор: Гончар Ростислав Михайлович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: Програмування та захист комп'ютерних систем і мереж

Науковий керівник: Чепун Віктор Миколайович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 6.26% і адресується до 401 першоджерела, усі запозичення фрагментарні або мають належним чином оформлені посилання, що відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи як оригінального тексту. Всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів і україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Керівник роботи _____ В.М. Чепун

Завідувач кафедри КбКСМ, гарант ОП _____ Ю.П. Клапп

Дата: 03.12.2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «магістр»

Магістр Гончар Ростислав Михайлович

Тема Метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем

Спеціальність 123 – Комп'ютерна інженерія

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень 11 ; кількість сторінок записки 101

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі розроблено метод формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині роботи

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подана загальна характеристика поставленої задачі, чітко визначено об'єкт, предмет та методи дослідження, сформульована актуальність. Визначені задачі, які необхідно вирішити для досягнення поставленої мети, практична цінність отриманих результатів, їхня новизна та наведені відомості про публікації. У першому розділі проведено огляд використовуваних в комп'ютерних системах методів кодування та шифрування інформації на предмет можливості їх комбінованого застосування для підвищення ефективності захисту інформаційних ресурсів., виконане обґрунтування актуальності теми дослідження і виконана постановка задачі. В другому розділі виконана розробка математичної моделі для реалізації методу, яка базується на основних положеннях теорії ймовірності та математичної статистики, теорії інформації та кодування, криптографії та прикладної криптології. В третьому розділі визначено основні положення методу та розроблено алгоритми його реалізації. Четвертий розділ присвячено апробації методу та алгоритмів його реалізації моделюванням.

4. Позитивні сторони роботи Кваліфікаційна робота має комплексну наукову і практичну цінність. Наукова цінність полягає у розробці методу формування шифрів зсуву із застосуванням оптимального кодування Хаффмена для підвищення ефективності захисту комп'ютерних систем, визначенні способу застосування методу оптимального кодування Хаффмена для зміни статистичних властивостей кодових алфавітів криптографічних шифрів зсуву, який став основою запропонованого методу, розробці нової математичної моделі, яка враховує особливості сумісного використання методів оптимального кодування та шифрів зсуву для підвищення криптостійкості систем захисту інформаційних ресурсів комп'ютерних систем. Практична цінність результатів дослідження полягає у обґрунтуванні можливості підвищення ефективності криптографічного захисту інформаційних ресурсів комп'ютерних систем застосуванням методів оптимального кодування перед реалізацією криптографічних методів захисту, а також у розробці алгоритмів реалізації методу

5. Негативні сторони проекту В роботі відсутній опис технічних засобів реалізації методу (програмних або програмно-апаратних).

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження Окремі описи в пояснювальній записці подано занадто деталізовано, що ускладнює сприйняття матеріалу наукової роботи фахівцями в обраній предметній галузі. Не на всі джерела з переліку є посилання в тексті пояснювальної записки

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Гурман Іван Васильович
кандидат технічних наук, доцент,
доцент кафедри інженерії програмного забезпечення

«01» грудня 2020. Гурман І.В. підпис

ХНУ Зам. 54, тир. 6000, 2007