

МІНІСТЕРСТВО ОСВІТИ І НАУКИ

Запорізький національний технічний університет

Методичні вказівки

до лабораторних робіт з дисципліни
“Прикладна криптологія” для студентів
спеціальності 6.170101 "Безпека інформаційних і
комунікаційних систем " і з дисципліни
“Криптографія і стеганографія ” для студентів
спеціальності 6.170102 "Системи технічного
захисту інформації"
для всіх форм навчання

Частина 1



2015

Методичні вказівки до лабораторних робіт з дисципліни
“Прикладна криптологія” для студентів спеціальності 6.170101
"Безпека інформаційних і комунікаційних систем " і з дисципліни
“Криптографія і стеганографія” для студентів спеціальності 6.170102
"Системи технічного захисту інформації" для всіх форм навчання.
Частина 1 /Укл.: Г.Л.Козіна, Г.І.Нікуліщев. – Запоріжжя: ЗНТУ, 2015.
– 55 с.

Укладачі: Г.Л.Козіна, доцент, к.ф.-м.н.
 Г.І.Нікуліщев, ст. викладач

Рецензент: Л.М. Карпуков, проф., д.т.н.

Відповідальний
за випуск: Г.Л.Козіна, доцент, к.ф.-м.н.

ЗМІСТ

Вступ	4
Лабораторна робота №1. Поняття теорії чисел	5
Лабораторна робота №2. Прості шифри	7
Лабораторна робота №3. Лінійні рекурентні послідовності	8
Лабораторна робота №4. Стандарт блочного шифрування ДСТУ ГОСТ 28147:2009	11
Лабораторна робота №5. Розв'язання порівнянь першого степеня	14
Лабораторна робота №6. Піднесення до степеня за модулем	16
Лабораторна робота №7. Перевірка чисел на простоту.....	17
Лабораторна робота №8. Факторизація чисел.....	19
Література.....	21
Додаток А Перетворення між текстом та числами.....	22
Додаток Б Порівняння та їхні властивості.....	24
Додаток В Прості шифри.....	28
Додаток Г Рекурентні співвідношення групі.....	36
Додаток Д Інтерфейс програми «ГОСТ»	41
Додаток Е ДСТУ ГОСТ 28147:2009	43
Додаток Ж Порівняннями першого степеня	46
Додаток К Алгоритм послідовного зведення в квадрат.....	49
Додаток Л Тест на простоту.....	51
Додаток М Розкладання чисел на множники	53

ВСТУП

*Єдина гарантія надійності
криптографічних алгоритмів –
це їх абсолютна відкритість.*

Криптологія - наука , що складається з двох гілок : криптографія і криптоаналіз.

Криптографія – наука про способи перетворення інформації з метою її захисту від незаконних користувачів .

Сучасна криптографія розвивається за такими напрямками: шифрування, генерування [псевдо]випадкових послідовностей, хешування, цифровий підпис.

Шифрування (зашифрування) – процес застосування шифру до інформації (відкритого тексту), що захищається, в шифроване повідомлення (шифртекст, криптограму) за допомогою певних правил та ключової інформації.

Дешифрування – процес, зворотний шифруванню, тобто перетворення шифрованого повідомлення в інформацію, що захищається, за допомогою певних правил та ключової інформації.

Під ключем в криптографії розуміють змінний елемент шифру, який застосовується для криптоперетворення конкретного повідомлення.

[Псевдо]випадкові послідовності широко використовуються в криптоалгоритмах.

Хешування – криптоперетворення повідомлення в бітовий образ фіксованої довжини – використовується в схемах цифрового підпису, при зберіганні паролів.

Криптоаналіз – наука (і практика її застосування) про методи та способи розкриття криптоалгоритмів.

Здатність криптоалгоритму протистояти всіляким атакам на нього називають стійкістю алгоритму. Під атакою розуміють спробу розкриття криптоперетворення.

ЛАБОРАТОРНА РОБОТА № 1

ПОНЯТТЯ ТЕОРІЇ ЧИСЕЛ

Мета роботи: Ознайомитися з основними поняттями теорії чисел.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

1.1 Завдання на лабораторну роботу

1. Представити власне прізвище записане латиницею у вигляді десяткового числа (див. Додаток А).
2. Перетворити десяткове число T на текст.
3. Виписати в конспект всі прості числа від 2 до 99.
4. Виписати в конспект таблицю відповідності шістнадцятеричних цифр десятковим числам та двійковим тетрадам (наприклад, $8_{16} = 8_{10} = 1000_2$, $C_{16} = 12_{10} = 1100_2$).
5. Знайти серед чисел a, b, c, d пари взаємно простих чисел (див. Додаток Б).
6. Знайти $НСД(a,b)$, $НСД(a,b,c)$, $НСД(a,b,c,d)$.
7. Знайти серед чисел a,b,c,d пари чисел, порівняних за модулем m .
8. Знайти суму і добуток чисел a, b, c, d за модулем m .
9. Обчислити значення функції Ейлера для кожного з чисел a,b,c,d .

Значення параметрів T, a, b, c, d, m візьміть із таблиці 1.1 згідно з номером варіанту N.

1.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Результати обчислень.
3. Висновки по роботі.

1.3 Контрольні питання

1. Які числа називаються простими? Взаємно простими?
2. Для чого використовується алгоритм Евкліда?
3. Які числа називаються порівняними за модулем m ?
4. За яким модулем всі цілі числа порівняні між собою?
5. Якими властивостями володіють порівняння?
6. Що таке клас залишків за модулем?
7. Що показує функція Ейлера? Як її обчислити?
8. Як формулюється теорема Ейлера?
9. Як формулюється мала теорема Ферма?
10. Як пов'язані теорема Ейлера і мала теорема Ферма?

Таблиця 1.1 – Варіанти завдань

N	T	a	b	c	d	m
1	6009318633877632121	45	37	27	111	17
2	88772934743480459988660078	76	24	19	33	9
3	22066599077835371	41	26	15	69	11
4	4859222852730381682	42	36	70	53	14
5	22643820939338093	71	52	39	87	15
6	5291294591267267956	18	67	35	49	4
7	24604245826172787	90	45	28	73	14
8	1206849699407272896621	35	23	92	15	8
9	5792036358531605868	64	79	39	22	6
10	91712171958636	77	42	59	60	17
11	1483985444626882588530	34	63	45	29	11
12	5071443942490206059	30	78	61	55	12
13	23775192852751730	51	91	43	27	8
14	360696279886943885423225	25	31	46	50	7
15	5075397850727345260	47	82	28	39	9

ЛАБОРАТОРНА РОБОТА № 2

ПРОСТІ ШИФРИ

Мета роботи: Ознайомитися з прикладами простих шифрів.

Використовуване програмне забезпечення: програми «Шифр_Полібія» та «Частотний криптоаналіз».

2.1 Завдання на лабораторну роботу

1. Запустити програму «Шифр_Полібія» (див. Додаток В).
2. Обрати мову, задати ключ, сформувати квадрат Полібія.
3. Виконати шифрування та розшифрування повідомлення.
4. Запустити програму «Частотний криптоаналіз».
5. Завантажити один з зашифрованих текстових файлів.
6. Скориставшись методом частотного криптоаналіза розкрити зашифрований текст. Виписати отриману таблицю заміन.
7. Порівняти частоти заміненних літер з еталоном.

2.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Повідомлення у відкритому та зашифрованому за квадратом Полібія вигляді.
3. Таблиця замін з частотами зустрічання літер. Уривок розкритого тексту.
4. Висновки по роботі.

2.3 Контрольні питання

1. Що є ключем в шифрі Полібія? Які вимоги висуваються до ключа?
2. За яким принципом формується квадрат Полібія?
3. Які ще відомі варіанти шифра Полібія?
4. Що таке таблиця замін? Які до неї вимоги?
5. Які різновиди шифру простої заміни існують?
6. Що таке частотний криптоаналіз?
7. Чи є шифр Полібія різновидом шифру простої заміни?

ЛАБОРАТОРНА РОБОТА № 3

ЛІНІЙНІ РЕКУРЕНТНІ ПОСЛІДОВНОСТІ

Мета роботи: Ознайомитися з генеруванням псевдовипадкових послідовностей, що задаються лінійними рекурентними співвідношеннями.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

3.1 Завдання на лабораторну роботу

1. Визначити глибину n лінійної рекурентної послідовності S (див. Додаток Г). Сформувати характеристичний многочлен лінійної рекурентної послідовності S і перевірити його на незвідність. Обчислити період лінійної рекурентної послідовності S при довільних n початкових значеннях x_i , $i = 1, 2, \dots, n$. З'ясувати, чи зміниться і як саме період лінійної рекурентної послідовності S при інших початкових значеннях x_i , $i = 1, 2, \dots, n$.

Послідовність S взяти із таблиці 3.1 згідно з номером варіанта N.

2. Визначити рекурентне співвідношення, що задає лінійну рекурентну послідовність глибини $n = 3$, ділянкою якої є $x_1, x_2, x_3, x_4, x_5, x_6$.

Параметри $x_1, x_2, x_3, x_4, x_5, x_6$ взяти із таблиці 3.2 згідно з номером варіанта N.

3.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Результати обчислень.
3. Висновки по роботі.

Таблиця 3.1 – Варіанти для завдання 1

N	S
1	$x_{k+8} = x_{k+5} \oplus x_{k+3} \oplus x_{k+2}$
2	$x_{k+4} = x_{k+3} \oplus x_{k+1} \oplus x_k$
3	$x_{k+5} = x_{k+4} \oplus x_{k+3} \oplus x_{k+2} \oplus x_k$
4	$x_{k+6} = x_{k+5} \oplus x_{k+3} \oplus x_{k+2} \oplus x_{k+1}$
5	$x_{k+6} = x_{k+2} \oplus x_k$
6	$x_{k+7} = x_{k+1} \oplus x_k$
7	$x_{k+7} = x_{k+4} \oplus x_{k+2}$
8	$x_{k+8} = x_{k+7} \oplus x_{k+2}$
9	$x_{k+7} = x_{k+5} \oplus x_{k+2}$
10	$x_{k+7} = x_{k+3} \oplus x_k$
11	$x_{k+7} = x_{k+2} \oplus x_k$
12	$x_{k+7} = x_{k+6} \oplus x_{k+5} \oplus x_{k+3} \oplus x_{k+2}$
13	$x_{k+7} = x_{k+5} \oplus x_k$
14	$x_{k+6} = x_{k+5} \oplus x_{k+2}$
15	$x_{k+6} = x_{k+5} \oplus x_{k+4} \oplus x_{k+1} \oplus x_k$

3.3 Контрольні питання

1. Як задається бітова лінійна рекурентна послідовність (ЛРП)?
2. Для чого використовуються ЛРП в криптографії?
3. Що таке глибина ЛРП?
4. Що таке характеристичний багаточлен ЛРП?
5. Який багаточлен називається незвідним?
6. Як перевірити багаточлен на незвідність?
7. Що таке період ЛРП?
8. Від чого залежить період ЛРП?
9. Як визначити рекурентне співвідношення, що задає ЛРП, знаючи ділянку послідовності?
10. Що таке регістри зсуву?

Таблиця 3.2 – Варіанти для завдання 2

N	x_1	x_2	x_3	x_4	x_5	x_6
1	1	1	1	1	1	1
2	1	1	0	0	1	1
3	1	1	1	0	0	1
4	1	0	1	0	1	0
5	0	1	1	0	0	1
6	1	0	0	1	0	1
7	0	0	1	1	0	0
8	1	1	1	0	1	0
9	1	0	0	1	1	0
10	1	0	1	1	1	0
11	0	0	1	1	1	0
12	0	1	1	1	0	0
13	0	1	0	1	0	1
14	0	0	1	0	1	1
15	1	0	0	1	1	1

ЛАБОРАТОРНА РОБОТА № 4

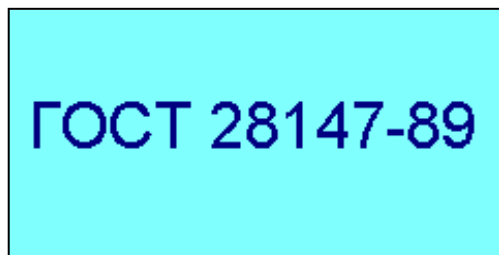
СТАНДАРТ БЛОЧНОГО ШИФРУВАННЯ ДСТУ ГОСТ 28147:2009

Мета роботи: Ознайомитись з методом шифрування й дешифрування даних з використанням алгоритму шифрування ДСТУ ГОСТ 28147:2009.

Використовуване програмне забезпечення: демонстраційна програма «ГОСТ».

4.1 Завдання на лабораторну роботу

1. Запустіть програму «ГОСТ» (див. Додаток Д).
2. Створіть 24-х розрядний графічний файл, для наочності рекомендується створити файл з однотонним фоном й крупним написом, наприклад:



3. Виконайте операцію шифрування в режимах простої заміни, гамування та гамування зі зворотнім зв'язком. (див. Додаток Е). Порівняйте зашифровані файли між собою та з оригіналом. Результати порівняння занесіть в таблицю 4.1 в графи 1-2.
4. Відкрийте зашифровані зображення в графічному редакторі та внесіть в них декілька змін. Виконайте розшифрування і порівняйте отримані файли з оригіналом. Результати порівняння занесіть в таблицю 4.1 в графу 3.
5. Поясніть характер впливу змін у зашифрованих файлах на результат розшифрування.

6. Сформуйте текстовий файл. Рекомендується вибрати рядок тексту без спецсимволів довжиною 8-10 блоків (64-80 символів). Перший і другий блоки тексту зробіть однаковими.
7. Виконайте операцію шифрування в режимах простої заміни, гамування та гамування зі зворотнім зв'язком. Порівняйте зашифровані файли між собою та з оригіналом. Результати порівняння занесіть в таблицю 4.1 в графу 4.
8. Замініть один символ в 5-му блоці зашифрованих файлів і виконайте розшифрування. Порівняйте отримані тексти з оригіналом. Результати порівняння занесіть в таблицю 4.1 в графу 5-6.
9. Поясніть характер впливу змін у зашифрованих файлах на результат розшифрування.
10. Сформуйте для текстового файлу імітоприкладку. Замініть декілька символів у файлі з імітоприкладкою і виконайте її перевірку. Результати занесіть в таблицю 4.1 в графу 7-8.

4.2 Зміст звіту

5. Титульний лист, тема і мета роботи.
6. Зашифровані та розшифровані файли.
7. Порівняльна таблиця.
8. Висновки.

4.3 Контрольні питання

11. Що таке блочний шифр?
12. Які основні характеристики має шифр ДСТУ ГОСТ 28147:2009?
13. Опишіть режими ГОСТ?
14. Чому режим простої заміни не рекомендується використовувати для шифрування?
15. Чим відрізняються режими гамування і гамування зі зворотнім зв'язком?
16. Як впливають помилки в зашифрованих текстах на результат розшифрування в різних режимах алгоритму ГОСТ?
17. Як формується імітоприкладка?
18. Для чого призначений режим імітоприкладки?
19. Що таке сітка Фейстеля?

Таблиця 4.1 – Аналіз шифру ДСТУ ГОСТ 28147:2009

№	Критерій порівняння	Режим		
		Проста заміна	Гамування	Гамування зі зворотнім зв'язком
Графічний файл				
1.	Чи є схожість з оригіналом?			
2.	Чи є схожість з іншими режимами?			
3.	Як зміни вплинули на результат розшифрування?			
Текстовий файл				
4.	Чи є однакові блоки?			
5.	Скільки символів змінилося?			
6.	Які блоки змінились?			
Імітоприкладка				
7.	Чи можна з'ясувати, що текст було змінено?			
8.	Чи можна з'ясувати, які саме зміни відбулись?			

ЛАБОРАТОРНА РОБОТА № 5

РОЗВ'ЯЗАННЯ ПОРІВНЯНЬ ПЕРШОГО СТЕПЕНЯ

Мета роботи: Ознайомитися з застосуванням розширеного алгоритму Евкліда для розв'язання порівнянь першого степеня.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

5.1 Завдання на лабораторну роботу

1. Розв'язати порівняння $a \cdot x = 1 \bmod b$ (див. Додаток Ж).
2. Розв'язати порівняння $b \cdot x = 1 \bmod m$.
3. Розв'язати порівняння $b \cdot x = a \bmod m$.
4. Розв'язати порівняння $a \cdot x = b \bmod m$.
5. Перевірити власні обчислення за допомогою математичного пакету Maple.

Параметри a , b , m взяти із таблиці 5.1 згідно з номером варіанта N.

5.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Результати обчислень.
3. Висновки.

5.3 Контрольні питання

1. Що називається розв'язком порівняння першого степеня?
2. За яких умов порівняння першого степеня мають розв'язок?
3. Скільки може бути розв'язків порівняння першого степеня?
4. Якими способами можна вирішити порівняння першого степеня?
5. В чому полягає розширений алгоритм Евкліда?

Таблиця 5.1 – Варіанти завдань

N	a	b	m
1	16	311	2129
2	42	137	2213
3	24	269	2287
4	38	151	1223
5	18	373	1291
6	46	149	1373
7	36	223	1451
8	20	313	1511
9	30	199	1583
10	40	139	1657
11	44	149	1733
12	32	233	1811
13	22	307	1889
14	34	197	1987
15	26	271	2053

ЛАБОРАТОРНА РОБОТА № 6

ПІДНЕСЕННЯ ДО СТЕПЕНЯ ЗА МОДУЛЕМ

Мета роботи: Ознайомлення з алгоритмом піднесення числа до певного степеня за заданим модулем.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

6.1 Завдання на лабораторну роботу

1. Знайти $2^{N+100} \bmod a$ (див. Додаток К).
2. Знайти $(N+10)^a \bmod b$.
3. Знайти $a^b \bmod m$.
4. Знайти $b^a \bmod m$.
5. Перевірити власні обчислення за допомогою математичного пакету Maple.

Параметри a , b , m взяти із таблиці 5.1 згідно з номером варіанта N.

6.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Результати обчислень.
3. Висновки.

6.3 Контрольні питання

1. Для чого використовується алгоритм послідовного зведення в квадрат?
2. Зі скількох кроків складається алгоритм послідовного зведення в квадрат?
3. В яких криптоалгоритмах виникає потреба підносити числа до степеня за певним модулем?

ЛАБОРАТОРНА РОБОТА № 7

ПЕРЕВІРКА ЧИСЕЛ НА ПРОСТОТУ

Мета роботи: Ознайомитися з алгоритмом перевірки натурального непарного числа на простоту.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

7.1 Завдання на лабораторну роботу

1. Перевірити, чи є число a свідком простоти числа b (див. Додаток Л).
2. Знайти ще один свідок простоти числа b .
3. З'ясувати, чи є числа c і d простими з імовірністю не менше $1 - \frac{1}{32}$.
4. Перевірити власні обчислення за допомогою математичного пакету Maple.

Параметри a , b , c і d взяти із таблиці 7.1 згідно з номером варіанта N.

7.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Результати обчислень.
3. Висновки.

7.3 Контрольні питання

1. Яке число називається простим?
2. Про що каже мала теорема Ферма?
3. Що таке свідок простоти числа?
4. Скільки свідків простоти числа необхідно знайти, щоб вважати його простим?
5. В чому полягає тест Ферма на простоту?

6. Які числа називають числами Кармайкла?
7. Які властивості мають числа Кармайкла?

Таблиця 7.1 – Варіанти завдань

N	a	b	c	d
1	22	343	701	927
2	17	351	833	739
3	25	377	837	967
4	16	327	961	887
5	24	331	941	871
6	15	359	771	811
7	32	397	951	883
8	27	383	919	831
9	19	341	967	851
10	26	357	973	863
11	31	361	859	781
12	20	347	947	841
13	29	389	881	767
14	33	307	843	977
15	23	339	911	867

ЛАБОРАТОРНА РОБОТА № 8

ФАКТОРИЗАЦІЯ ЧИСЕЛ

Мета роботи: Ознайомитися з алгоритмами розкладання чисел на множники.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

8.1 Завдання на лабораторну роботу

1. Знайти хоча б один простий множник числа Z (див. Додаток М).

2. Розкласти числа u , v і w на прості множники.

Перевірити власні обчислення за допомогою математичного пакету Maple.

Параметри Z , u , v і w взяти із таблиці 8.1 згідно з номером варіанта N.

8.2 Зміст звіту

1. Титульний лист, тема і мета роботи.
2. Результати обчислень.
3. Висновки по роботі.

8.3 Контрольні питання

1. Яке число називається простим?
2. Що таке факторизація?
3. Стійкість яких криптоалгоритмів базується на задачі факторизації?
4. В чому полягає алгоритм факторизації ρ -Полларда?
5. Яким чином факторизується будь-яке складене число?
6. Чим обмежується діапазон підбору множників в переборному алгоритмі?

Таблиця 8.1 – Варіанти завдань

N	<i>z</i>	<i>u</i>	<i>v</i>	<i>w</i>
1	217	2279	21939	33111
2	361	1739	51121	399497
3	209	1067	31565	101097
4	259	2257	8493	80575
5	391	1501	22765	124623
6	203	1633	7527	134469
7	287	2059	16851	70209
8	299	2419	20305	507787
9	247	1157	17969	308725
10	221	2701	34013	234073
11	323	1411	51611	237725
12	301	2449	26243	56097
13	253	1679	6567	92619
14	341	2077	36001	131575
15	237	1219	23241	662039

ЛІТЕРАТУРА

1. Фергюсон Н., Шнайер Б. Практическая криптография – М.: Диалектика, 2005. – 424 с.
2. Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях. М.Ж КУДИЦ-ОБРАЗ, 2001 – 386с.
3. Баричев С.Г., Гончаров В.В. Основы современной криптографии. М.: Горячая линия - Телеком, 2001.
4. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях / Под ред. В.Ф. Шаньгина. – 2-е изд., перераб. и доп. – М.: Радио и связь, 2001. – 376 с.
5. Виноградов И.М. Основы теории чисел. – М.: Наука, 1981. – 168 с.
6. Курош А.Г. Курс высшей алгебры. – М.:Наука, 1971. – 423 с.
7. Жельников В. Криптография от папируса до компьютера. — М.: АБФ, 1996. — 335 с.
8. Ростовцев А.Г. Алгебраические основы криптографии. – СПб.: НПО «Мир и семья, ООО «Интерлайн», 2000. – 354 с.
9. Menezes A., van Oorschot P., Vanstone S. Handbook of applied cryptography. CRC Press, 1997.
10. Введение в криптографию / Под общ. Ред. В.В. Яценко. – 2-е изд., испр. – М.:МЦНМО: «ЧеРо», 1999. – 272с.
11. Нечаев В.И. Элементы криптографии (Основы теории защиты информации): Учеб. пособие для ун-тов и пед. вузов / Под ред. В.А.Садовниченко – М.: Высш.шк., 1999. – 109с.
12. Вербіцький О.В. Вступ до криптології. – Львів: ВНТЛ, 1998. – 248.
13. Саломаа А. Криптография с открытым ключом: Пер. с англ. – М.: Мир, 1995. – 318с.
14. Сمارт Н. Криптография. – Москва: Техносфера, 2005. -528 с.

Додаток А

Перетворення між текстом та числами

Криптографія вивчає методи та засоби перетворення інформації з відкритого вигляду у закритий (зрозумілий лише тим, хто володіє ключем). Інформація, як правило, подається в текстовому вигляді. Тим не менш, більшість криптографічних алгоритмів оперують числовими величинами. Таким чином потребується попередня обробка тексту для представлення його у вигляді числа – кодування.

При кодуванні кожному символу тексту надається певний унікальний цифровий код. В сучасних інформаційних технологіях використовуються багато способів кодування: ASCII (American Standard Code for Information Interchange), Unicode тощо.

Для перетворення тексту на число кожна літера або знак замінюється на запис у цифрах, згідно з таблицею кодування. Стандартна таблиця кодування ASCII наведена в таблиці А.1.

Приклади.

Перетворимо на десяткове число повідомлення «Student» (без лапок). Для цього замінимо кожну літеру відповідним кодом з таблиці А.1. і переведемо отримане шістнадцятеричне число у десяткове згідно з правилами перетворення між системами числення:

$$\text{Student} = 53747564656\text{E}74_{16} = 23490470611349108_{10}.$$

Для зворотного перетворення необхідно перевести десяткове число у шістнадцятеричне і замінити кожну пару цифр на відповідний символ з таблиці кодування.

Знайдемо, яке повідомлення записується десятковим числом 1515082837.

$$1515082837_{10} = 5\text{A}4\text{E}5455_{16} = \text{ZNTU}$$

Таблиця А.1 –Таблиця ASCII-кодів

	00	10	20	30	40	50	60	70
0		►		0	@	P	`	p
1	☐	◄	!	1	A	Q	a	q
2	☐	↑	"	2	B	R	b	r
3	♥	!!	#	3	C	S	c	s
4	♦	¶	\$	4	D	T	d	t
5	♣	§	%	5	E	U	e	u
6	♠	=	&	6	F	V	f	v
7	•	⌚	'	7	G	W	g	w
8	☐	↑	<	8	H	X	h	x
9	o	↓	>	9	I	Y	i	y
A	☐	→	*	:	J	Z	j	z
B	♂	←	+	;	K	[	k	{
C	♀	└	,	<	L	\	l	!
D	♂	↔	-	=	M	]	m	}
E	♂	▲	.	>	N	^	n	~
F	※	▼	/	?	O	_	o	Δ

Допоміжні функції пакету MAPLE

convert (r,hex) – перетворює десяткове число r в 16-ричне

r:=30; a:=convert(r,hex); $r := 30$ $a := 1E$

convert (S,decimal, hex) – перетворює 16-ричне число S в десятичне

r:=convert("1E", decimal, hex); $r := 30$

r:=convert("1E2A", decimal, hex); $r := 7722$

convert (z,binary) – перетворює десяткове число Z в двоичне

z:=19; z2:=convert(z,binary); $z := 19$ $z2 := 10011$

convert(z,base,2) – перетворює десяткове число Z в двоичний масив

zar:=convert(z,base,2); $zar := [1, 1, 0, 0, 1]$

Додаток Б

Порівняння та їхні властивості

Натуральне число називається *простим*, якщо воно не має дільників, крім себе і 1.

Натуральне число називається *складеним*, якщо воно має, принаймні, два дільники, більших за 1.

Позначимо через $НСД(a, b)$ *найбільший спільний дільник* двох чисел a і b .

Якщо два числа a і b не мають спільних дільників, крім 1, вони називаються *взаємно простими*, тобто $НСД(a, b) = 1$ або $(a, b) = 1$.

Алгоритм Евкліда пошуку найбільшого спільного дільника

Нехай маємо два натуральних числа a і b , $a < b$. Знайдемо залишки від ділення

$$b = a \cdot q_0 + r_1$$

$$a = r_1 \cdot q_1 + r_2$$

$$r_1 = r_2 \cdot q_2 + r_3$$

$$r_2 = r_3 \cdot q_3 + r_4$$

.....

$$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n$$

$$r_{n-1} = r_n \cdot q_n$$

$$НСД(a, b) = r_n.$$

Приклад 1.

Знайдемо $НСД(114, 534)$.

$$534 = 114 \cdot 4 + 78,$$

$$114 = 78 \cdot 1 + 36,$$

$$78 = 36 \cdot 2 + 6,$$

$$36 = 6 \cdot 6.$$

$$НСД(114, 534) = 6.$$

Приклад 2.

Знайдемо $НСД(114, 534, 22)$.

$$НСД(114, 534, 22) = НСД(НСД(114, 534), 22) = НСД(6, 22) = 2.$$

$$НСД(114, 534, 22) = 2.$$

Порівняння і їхні властивості

Два цілих числа називаються порівняними за модулем m , якщо вони обидва мають однаковий залишок від ділення на m . Загальноприйнятий запис « a порівняне з b за модулем m »:

$$a = b \bmod m.$$

Порівняння мають наступні властивості:

1. $a = b \bmod m \Rightarrow a = b + m \cdot t$, де t – ціле число
2. $a = a \bmod m$
3. $a = b \bmod m \Rightarrow b = a \bmod m$
4. $a = b \bmod m$ і $b = c \bmod m \Rightarrow a = c \bmod m$
5. $a = b \bmod m \Rightarrow a + s = b + s \bmod m$
6. $a = b \bmod m \Rightarrow a \cdot s = b \cdot s \bmod m$
7. $a \cdot s = b \cdot s \bmod m$ і $(s, m) = 1 \Rightarrow a = b \bmod m$
8. $a = b \bmod m$ і $c = d \bmod m \Rightarrow a + c = b + d \bmod m$
9. $a = b \bmod m$ і $c = d \bmod m \Rightarrow a \cdot c = b \cdot d \bmod m$
10. $a = b \bmod m \Rightarrow a^n = b^n \bmod m$ для будь-якого цілого n

Всі цілі числа, які мають однаковий залишок від ділення на m , порівняні між собою за модулем m і належать до одного класу залишків за цим модулем. Клас залишків позначається за найменшим додатнім числом (або 0), яке до нього належить. Наприклад, за модулем 5 існують наступні класи залишків $\{0, 1, 2, 3, 4\}$ (стовпці в таблиці Б.1).

Згідно з властивостями 8-10, при виконанні обчислень за модулем m операнди, що належать до одного класу залишків, взаємозамінні.

Наприклад, $(13 \cdot 12 - 8) \bmod 5 = 156 - 8 = 1 - 3 = -2 = 3$,

$$2^8 \bmod 10 = 2^5 \cdot 2^3 = 32 \cdot 8 = 2 \cdot 8 = 16 = 6$$

$$\frac{7}{8} \bmod 5 = \frac{2}{3} = \frac{12}{3} = 4,$$

Таблиця Б.1 –Класи залишків за модулем 5

-5	-4	-3	-2	-1
0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

Функція Ейлера¹

Функцією Ейлера $\varphi(n)$ називається число натуральних чисел, що не перевищують n і взаємно простих з n .

Наприклад, для $n = 8$ функція Ейлера $\varphi(8) = 4$, оскільки числа 1, 3, 5, 7 є взаємно простими з числом 8 і не перевищують його.

¹ Леонард Ейлер (1701-1783)

Функція Ейлера має наступні властивості:

1. $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$, якщо a і b взаємно прості;
2. $\varphi(p) = p - 1$, якщо p – просте число;
3. $\varphi(p^\alpha) = p^{\alpha-1} \cdot (p-1)$, якщо p – просте число;
4. Якщо $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_k^{\alpha_k}$, де $p_1, p_2, \dots, p_k$ – попарно різні прості числа, то

$$\varphi(n) = p_1^{\alpha_1-1} \cdot p_2^{\alpha_2-1} \cdot \dots \cdot p_k^{\alpha_k-1} \cdot (p_1-1) \cdot (p_2-1) \cdot \dots \cdot (p_k-1).$$

Приклад.

$$\varphi(45) = \varphi(5) \cdot \varphi(9) = (5-1) \cdot \varphi(3^2) = 4 \cdot 3 \cdot (3-1) = 24.$$

Теорема Ейлера. Якщо a і n взаємно прості, то $a^{\varphi(n)} = 1 \bmod n$.

Мала теорема Ферма. Якщо p – просте число і $1 \leq a \leq p-1$, то $a^{p-1} = 1 \bmod p$.

Приклади.

$$3^4 = 1 \bmod 8, \quad \varphi(8) = 4.$$

$$6^{10} = 1 \bmod 11.$$

Допоміжні функції пакету MAPLE

$a \bmod m$ або $\text{modp}(a, m)$ – обчислює залишок від ділення a на m

$17 \bmod 5$; 2 $\text{modp}(55, 7)$; 6 $5+7+2 \bmod 10$; 4
 $5*7*2 \bmod 10$; 0 $\text{modp}(5+7+2, 10)$; 4

$\text{gcd}(a, b)$ – обчислює найбільший спільний дільник двох чисел a і b

$\text{gcd}(6, 18)$; 6 $\text{gcd}(6, 25)$; 1 $\text{gcd}(\text{gcd}(6, 18), 3)$; 3

$\text{ifactor}(n)$ – розкладає ціле число n на множники

```
ifactor(60); (2)2 (3) (5)
```

`isprime(n)` – повертає `true`, якщо n – просте число

```
isprime(61); true
```

```
isprime(60); false
```

Додаток В

Прості шифри

Історія криптографії налічує не одне тисячоліття. Згадки про примітивні шифри зустрічаються ще до нашої ери. У криптографії древніх часів використалися два види шифрів: заміна й перестановка. Клас шифрів, які зводяться до замін або перестановок літер відкритого тексту прийнято називати простими.

Шифр Цезаря

Історичним прикладом шифру заміни є шифр Цезаря (I в. до н.е.), описаний істориком Древнього Рима Светонієм. Гай Юлій Цезар використав у своїй переписці шифр власного винаходу. Історик Светоній не приводить фактів дешифрування переписки Цезаря. Сам Цезар все життя використав той самий ключ. Цим шифром він користувався, зокрема, для обміну посланнями із Цицероном.

Ключем у шифрі Цезаря є число 3. Кожна буква у вихідному тексті зміщується по алфавіту на 3 позиції.

Використовуємо латинський алфавіт, що містить 26 літер і пробіл, який стоїть після букви Z:

ABCD EFGH IJKL MNOP QRST UVWX YZ *Space*

Вихідний текст:

«STATUS BELLI» – СТАН ВІЙНИ

Зашифрований текст

«VWDWXV C EHOOL»

Шифр перестановки

Задається перестановка з n елементів.

Текст, що підлягає зашифруванню, розбивається на блоки довжини n . У кожному блоці символи переставляються відповідно до заданої перестановки.

Розглянемо приклад шифрування для перестановки

1	2	3	4	5
3	2	5	1	4

Вихідний текст:

«СВЯЩЕ ННАЯР ИМСКА ЯИМПЕ РИЯБЭ»

Зашифрований текст

«ЩВСЕЯЯННРАКМИАСПИЯЕМЬИРЭЯ»

Зашифрований текст виписується без пропусків.

Шифр «Считала»

Одним з перших фізичних приладів, що реалізують шифр перестановки, є так званий жезл Считала. Він був винайдений в «варварській» Спарті в часи Лікурга (V в. до н.е.). Шифр Считала застосовувався в часи війни Спарти проти Афін. Рим швидко скористався цим шифром. Для зашифрування тексту використався циліндр заздалегідь обумовленого діаметра. На циліндр намотувався тонкий ремінь із пергаменту, і текст виписувався порядкове по утворюючому циліндрі (уздовж його осі). Потім ремінь змотувався й відправлявся одержувачеві повідомлення. Одержувач намотував його на циліндр того ж діаметра й читав текст по осі циліндра. Ключем шифру є діаметр циліндра і його довжина, які по суті породжують дворядковий запис, аналогічний підстановці.

Винахід дешифровального пристрою приписується великому Аристотелю. Він запропонував використати конусоподібний «спис», на який намотується ремінь: цей ремінь пересувався по осі до того положення, поки не з'являвся осмислений текст.

Квадрат Полібія²

Один з найдавніших простих шифрів приписують грецькому громадському діячеві та науковцю Полібію. В літературі описано декілька варіантів його шифру «Квадрат Полібія».

За Жельниковим [7], шифрування відбувається наступним чином: обирається слово-ключ, кількість літер в якому залежить від мови; ключ записується в першому рядку квадрата, в подальших рядках виписуються літери алфавіта, відсутні в ключі; кожна літера повідомлення замінюється на ту, що стоїть в квадраті на рядок нижче (при розшифруванні – на рядок вище).

² Полібій - грецький державний діяч, полководець, історик, жив в III в. до н.е.

На рисунку В.1 показано скріншот з програми «Шифр_Полібія» з квадратом, сформованим за допомогою ключа «весна».

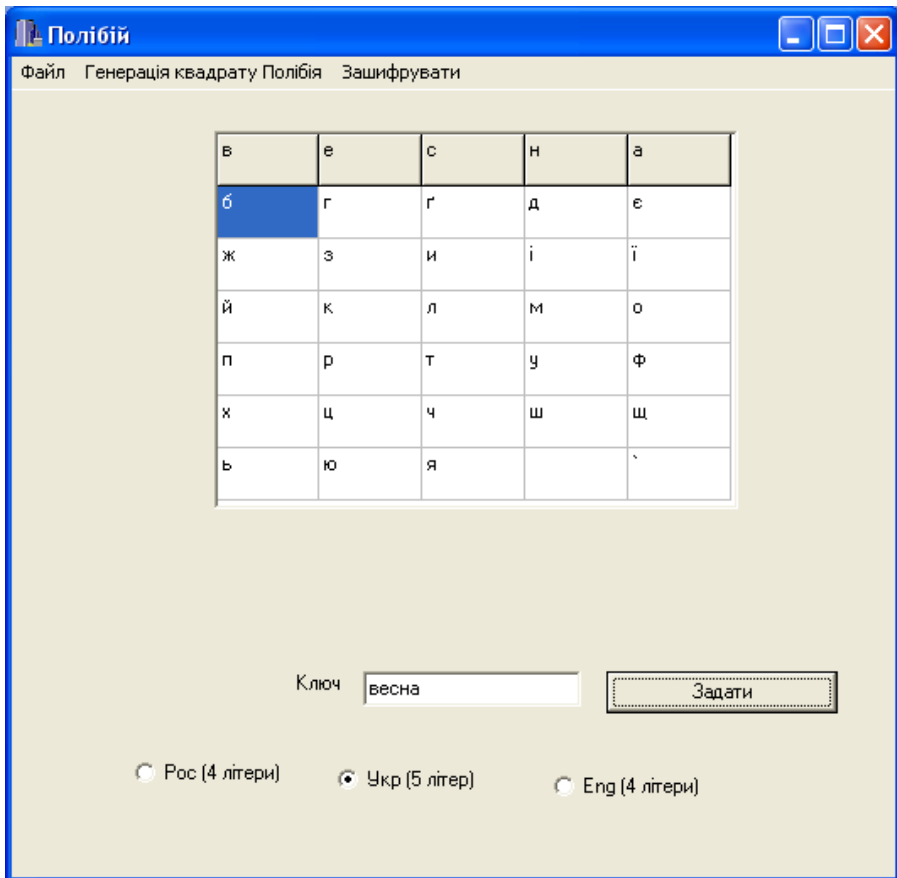


Рисунок В.1 – Квадрат Полібія

Шифр простої заміни

До простих шифрів, зокрема, належать шифри простої заміни – сукупність шифрів, які зводяться до побудови таблиці замін символів алфавіту і підстановки літер повідомлення за таблицею.

Якщо символи одного алфавіту замінюються на символи того ж самого алфавіту, шифр називається моноалфавітним, якщо іншого (наприклад, літери на числа чи позначки) – поліалфавітним. Якщо заміні підлягає не одна літера, а буквосполучення, шифр може бути біграмним (2 літери), триграмним (3 літери) тощо.

Частотний криптоаналіз

Процеси шифрування і розшифрування в шифрі простої заміни неможливі без таблиці заміन. Однак, при необхідності дізнатися вміст зашифрованого повідомлення без знання таблиці замін можна вдатися до методу частотного криптоаналізу.

Частотний криптоаналіз полягає в порівнянні статистичних характеристик зашифрованого тексту з еталонною статистикою текстів відповідною мовою.

Для кожного символу зашифрованого повідомлення підраховується частота його зустрічання в повідомленні.

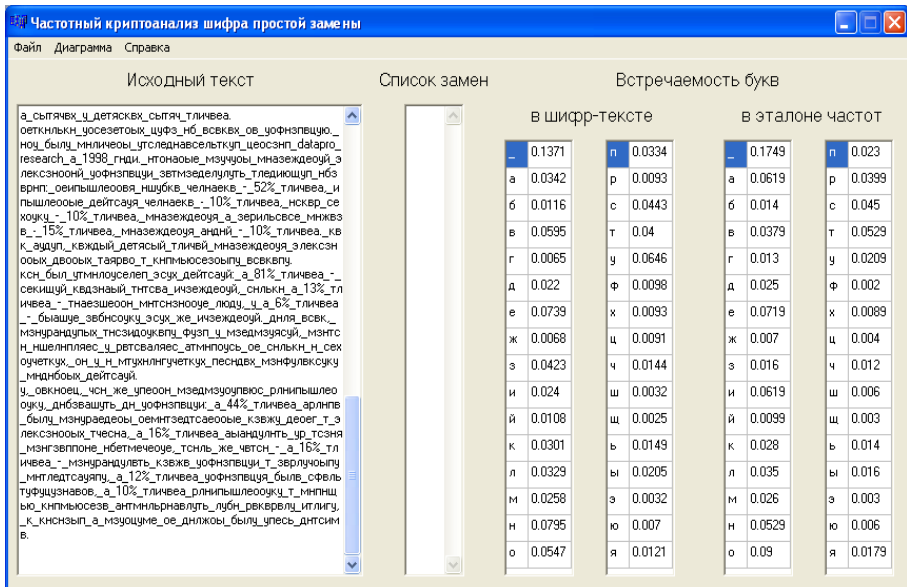


Рисунок В.2 – Частотный криптоанализ

В еталоні для відповідної мови розшукується символ із найближчою частотою зустрічань і приймається за потенційну заміну для обраного символу. Таким чином, будується можлива таблиця замін і застосовується до зашифрованого повідомлення. Підбір потенційних замін повторюється до повного розкриття повідомлення.

На рисунку В.2 наведено скріншот з програми «Частотний криптоаналіз» з порівняльною таблицею частот зустрічання символів в зашифрованому повідомленні та еталоні. Програма дозволяє по черзі проводити заміни літер зі схожими частотами. Для проведення заміни необхідно в лівій частині таблиці обрати літеру, яку треба замінити, а в правій – літеру, на яку замінювати. За проведеними замінами в середній частині вікна буде сформована таблиця замін.

Шифр Гронсфельда³

Ключем є деяке десяткове число. Цифри цього числа циклічно записуються під символами відкритого тексту. При шифруванні кожна буква відкритого тексту зсувається за алфавітом на число позицій, зазначених під нею.

1	2	3	4	5	6	7	8	9	10	11	12	13
A	B	C	D	E	F	G	H	I	J	K	L	M
14	15	16	17	18	19	20	21	22	23	24	25	26
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Вихідний текст						G	E	R	M	A	N	Y
Ключ						1	3	5	7	9	1	3
Зашифрований текст						H	H	W	T	J	O	B

³ Граф Гронсфельд був начальником першого дешифрувального відділення у Німеччині. У 1734г. він модернізував ідеї Альберті і Вижинера – творців шифрів багатоалфавитної заміни.

Альберті, Леон Баттіста (Alberti, Leon Battista) (1404-1472), італійський філолог, математик, криптограф, архітектор.

Блез де Віженер – придворний короля Франції Генріха III, французький посол в Римі. У запропонованому ним в 1585 році шифрі використовується секретне слово або фраза. Ця система періодичної лозунгової заміни стала першим великим відкриттям в криптографії з часів Юлія Цезаря. «Квадратний шифр» Віженера протягом 350 років вважався однією з найбільш надійних систем. Головною перевагою методу Віженера була його простота.

Біграмні шифри

У біграмних шифрах повідомлення розбивається на біграми – блоки по дві букви

Біграмні шифри були запропоновані Іоганном Трісемусом (Німеччина). У 1508 році він опублікував першу друковану роботу з криптології «Поліграфія». Шифр Playfair, заснований на біграмному шифрі, використовувався Великобританією в Першу світову війну.

Для зашифрування застосовується квадрат Полібія, заповнений літерами алфавіту випадковим чином або з використанням ключового слова. За певним правилом біграма відкритого тексту замінюється на біграму: кожна буква біграми поміщається в квадрат Полібія.

c	i	p	h	e
r	a	b	d	f
g	k	l	m	n
o	q	s	t	u
v	w	x	y	z

Рисунок В.3 – Квадрат Полібія

Якщо обидві літери опинилися в одному рядку, наприклад, *rb*, то при шифруванні беруться букви, що стоять праворуч від них: *rb* → *ad*.

Якщо обидві літери опинилися в одному стовпці, наприклад, *bх*, то при шифруванні беруться букви, що стоять під ними: *bх* → *lp*.

Якщо букви біграми лежать в різних рядках і стовпцях, наприклад, *rt*, то при шифруванні беруться букви з «кутів прямокутника»: *rt* → *do*.

Вихідний текст	gr	ea	tb	ri	ta	in	is	th	el	ar	ge
Зашифрований текст	og	if	sd	ac	qd	ek	pq	yd	pn	ba	nc

У другу світову війну застосовувався біграмний шифр Double Playfair, що використовує подвійний квадрат Полібія. Цей шифр був запропонований англійцем Чарльзом Уїнстоном в 1854г.

Для зашифрування біграми використовуються два квадрата. Перша буква біграми поміщається в перший квадрат, друга – у другій.

c	i	p	h	e	d	o	u	b	l
r	a	b	d	f	e	s	q	a	r
g	k	l	m		c	f	g	h	i
o	q	s	t	u	k	m	n	p	t
v	w	x	y	z	v	w	x	y	z

Рисунок В.4 – Подвійний квадрат Полібія

Якщо букви біграми утворюють прямокутник, то беруться букви з «кутів прямокутника».

Якщо обидві літери лежать в одному рядку, то беруться букви з того ж рядка, що стоять на тих же місцях в протилежних таблицях.

Вихідний текст tw ok ey wo rd sa re ch os en
Зашифрований текст my ko bz wi ec pb er bg mr uu

Шифр Вернама⁴

Шифр Вернама здійснює побітове додавання n -бітового відкритого x_i тексту і n -бітового ключа k_i :

$$y_i = x_i \oplus k_i, \quad i = 1, 2 \dots n.$$

⁴ Гілберт Вернам (Gilbert S. Vernam, 1890-1960) – американський інженер. Закінчив Массачусетський коледж (де був президентом Асоціації бездротового зв'язку) і в 1914 році вступив на роботу в AT & T у відділ телеграфного зв'язку. Незабаром його запрошують в спеціальну секцію, зайняту питаннями секретності телеграфного зв'язку.

У 1917 році Вернам винаходить перший автоматичне шифрувальний - дешифрувальний пристрій, а наступного року, після консультацій з колегами з галузі і військових відомств, пропонує прийом «нескінченного ключа» – те, що ми сьогодні і називаємо «шифром Вернама», або «одноразовим блокнотом». Через три з лишком десятки років Клод Шеннон строго доведе абсолютну теоретико - інформаційну стійкість шифру Вернама.

До криптологічних ідей Вернама дуже чуйно поставилися в Німеччині (використання «одноразових блокнотів» – модифікації шифру Вернама – введено в практику німецьких дипломатів між 1921 і 1923 рр.) і Радянському Союзу, але не в США, хоча в Америці система Вернама висвітлювалася під багатьох публікаціях.

Вихідний текст	0	0	1	1	0	1	0	1
Гамма шифра	1	0	1	1	0	0	0	1
Зашифрований текст	1	0	0	0	0	1	0	0
Гамма шифра	1	0	1	1	0	0	0	1
Вихідний текст	0	0	1	1	0	1	0	1

Шифр Вернама є абсолютно стійким шифром.

Для абсолютної стійкості шифр необхідні: повна випадковість (рівноймовірно) ключа (це, зокрема, означає, що ключ не можна виробляти за допомогою якого-небудь детермінованого пристрої); рівність довжини ключа і довжини відкритого тексту; однократність використання ключа.

У разі порушення хоча б однієї з цих умов шифр перестає бути абсолютно стійким і з'являються принципові можливості для його розкриття (хоча вони можуть бути важко реалізованими) .

Додаток Г

Рекурентні співвідношення

В симетричному шифруванні широко використовується метод гамування – накладання на відкритий текст T певної послідовності γ (гами шифру). При розшифруванні та ж сама послідовність γ накладається на шифртекст C , тобто гама шифру γ є ключем шифрування. Під накладанням, як правило, маються на увазі операції додавання-віднімання.

В якості гами шифру γ нерідко виступає двійкова (бітова) послідовність, оскільки дуже зручно використовувати для накладання гами бітову операцію XOR (додавання за модулем 2):

$$\begin{aligned} C &= T \oplus \gamma && \text{зашифрування} \\ T &= C \oplus \gamma && \text{розшифрування.} \end{aligned}$$

В симетричному шифруванні один і той самий ключ має бути наявний і у відправника, і у отримувача. Гама шифру має таку ж довжину, як і повідомлення, тому недоцільно передавати її по каналам зв'язку.

Одним зі способів вирішення цієї проблеми є незалежне генерування гами шифру на боці відправника і на боці отримувача. При цьому використовуються генератори псевдовипадкових послідовностей, а по каналам зв'язку передається лише початковий стан генератора.

Розглянемо генератор псевдовипадкових послідовностей на основі лінійних рекурентних послідовностей (ЛРП).

ЛРП задається наступним співвідношенням для :

$$x_{k+n} = a_1 \cdot x_{k+n-1} \oplus a_2 \cdot x_{k+n-2} \oplus \dots \oplus a_{n-1} \cdot x_{k+1} \oplus a_n \cdot x_k,$$

де $k = 1, 2, \dots$

Члени послідовності x_i і коефіцієнти a_i приймають значення із множини $\{0, 1\}$.

Величина n називається *глибиною* послідовності.

Початковим станом псевдовипадкової послідовності виступають перші n значень x_i ($i = 1, 2, \dots, n$).

Характеристичним многочленом ЛРП називається многочлен вигляду:

$$f(\lambda) = \lambda^n + a_1 \cdot \lambda^{n-1} + a_2 \cdot \lambda^{n-2} + \dots + a_{n-1} \cdot \lambda + a_n.$$

Наприклад, лінійній рекурентній послідовності $x_{k+3} = x_{k+1} \oplus x_k$ глибини $n=3$ відповідає характеристичний многочлен $f(\lambda) = \lambda^3 + \lambda + 1$, лінійній рекурентній послідовності $x_{k+6} = x_{k+5} \oplus x_{k+4} \oplus x_{k+2}$ глибини $n=4$, яка співпадає зі співвідношенням $x_{k+4} = x_{k+3} \oplus x_{k+2} \oplus x_k$, відповідає характеристичний многочлен $f(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + 1$.

Многочлен, який не можна розкласти на множники нижчого ступеня, називається *незвідним* (аналог простого числа).

Для того щоб з'ясувати, чи є многочлен $f(\lambda)$ степеня n незвідним, треба розділити на нього багаточлен $g(\lambda) = \lambda^{2^{n-1}} + 1$.

Якщо $\frac{g(\lambda)}{f(\lambda)}$ не має залишку, многочлен $f(\lambda)$ є незвідним.

Наприклад, многочлен $f(\lambda) = \lambda^3 + \lambda + 1$ є незвідним, оскільки многочлен $g(\lambda)$ ділиться на многочлен $f(\lambda)$:

$$\frac{g(\lambda)}{f(\lambda)} = \frac{\lambda^7 + 1}{\lambda^3 + \lambda + 1} = \lambda^4 + \lambda^2 + \lambda + 1.$$

Многочлен $f(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + 1$ не є незвідним, оскільки многочлен $g(\lambda)$ не ділиться на многочлен $f(\lambda)$:

$$\frac{g(\lambda)}{f(\lambda)} = \frac{\lambda^{15} + 1}{\lambda^4 + \lambda^3 + \lambda^2 + 1} = \lambda^{11} + \lambda^{10} + \lambda^8 + \lambda^4 + \lambda^3 + \lambda + \frac{\lambda + 1}{\lambda^4 + \lambda^3 + \lambda^2 + 1}$$

Дійсно, многочлен $f(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + 1$ можна розкласти на множники: $f(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + 1 = (\lambda + 1) \cdot (\lambda^3 + \lambda + 1)$.

Важливою з точки зору криптографічної стійкості характеристикою генератора на ЛРП є період генерованої послідовності. Максимальне значення періоду залежить від глибини ЛРП та співвідношення, що її задає, а конкретне значення – від початкового стану послідовності.

Максимальний період послідовності глибини n дорівнює $T_{\max} = 2^{n-1}$.

Наприклад, $T_{\max} = 7$ при $n = 3$, $T_{\max} = 15$ при $n = 4$.

Якщо характеристичний многочлен ЛРП є незвідним, період послідовності T буде дільником значення $T_{\max}$.

Якщо характеристичний многочлен ЛРП є незвідним і $T_{\max}$ – просте число, то період послідовності дорівнює $T_{\max}$ при будь-яких початкових значеннях, окрім всіх нулів.

Таким чином, послідовність $x_{k+3} = x_{k+1} \oplus x_k$ має максимальне можливий період, рівний 7 ($2^3 - 1$), при будь-яких початкових значеннях, окрім всіх нулів.

Послідовність $x_{k+4} = x_{k+3} \oplus x_{k+2} \oplus x_k$ може мати будь-який період, менший числа 15 ($2^4 - 1$).

Криптоаналітик, знаючи глибину ЛРП n і маючи ділянку цієї послідовності $x_{k+1}, x_{k+2}, \dots, x_{k+2n}$, може вирахувати коефіцієнти відповідного рекурентного співвідношення, розв'язавши систему відносно a_i . Наприклад, для ділянки $x_1, x_2, \dots, x_{2n}$:

$$x_{n+1} = a_1 \cdot x_n \oplus a_2 \cdot x_{n-1} \oplus \dots \oplus a_{n-1} \cdot x_2 \oplus a_n \cdot x_1$$

$$x_{n+2} = a_1 \cdot x_{n+1} \oplus a_2 \cdot x_n \oplus \dots \oplus a_{n-1} \cdot x_3 \oplus a_n \cdot x_2$$

...

$$x_{2n} = a_1 \cdot x_{2n-1} \oplus a_2 \cdot x_{2n-2} \oplus \dots \oplus a_{n-1} \cdot x_{n+1} \oplus a_n \cdot x_n$$

Якщо $n = 3$, отримуємо систему рівнянь

$$a_1 \cdot x_3 \oplus a_2 \cdot x_2 \oplus a_3 \cdot x_1 = x_4$$

$$a_1 \cdot x_4 \oplus a_2 \cdot x_3 \oplus a_3 \cdot x_2 = x_5$$

$$a_1 \cdot x_5 \oplus a_2 \cdot x_4 \oplus a_3 \cdot x_3 = x_6$$

Для ділянки ~~1,0,1,0,0,1~~ лінійної рекурентної послідовності глибини $n=3$ розв'язком системи

$$a_1 \cdot 1 \oplus a_3 \cdot 1 = 0$$

$$a_2 \cdot 1 = 0$$

$$a_3 \cdot 1 = 1$$

є $a_1 = 1, a_2 = 0, a_3 = 1$.

Таким чином, знайдено рекурентне співвідношення, що задає лінійну рекурентну послідовність глибини $n=3$: $x_{k+3} = x_{k+2} \oplus x_k$.

Для ускладнення задачі криптоаналітика нерідко декілька генераторів псевдовипадкових послідовностей на основі ЛРП об'єднують в один, синхронізуючи їхні виходи. Таки генератори називають регістрами зсуву (LFSRs – Linear Feedback Shift Registers).

Шифр А5

Шифр А5 був розроблений в 1989 році і існує у двох версіях: А5/1 – «сильна» версія шифру, дозволена до застосування тільки в деяких країнах, і А5/2 – «ослаблена» версія, дозволена до вільного застосування. Алгоритм А5 використовується в системах GSM (Group Special Mobile) для закриття зв'язку між абонентом і базовою станцією.

Алгоритм А5 реалізує потоковий шифр на основі трьох лінійних регістрів зсуву з нерівномірним рухом. У GSM-мережах використовується 64-бітний ключ, який забезпечується трьома регістрами в 19, 22 і 23 біта (що дає в сумі 64 біта).

У А5 використовується змінюване управління тактуванням. Кожен регістр тактується залежно від свого середнього біта, потім виконується XOR із зворотною пороговою функцією середніх бітів всіх трьох регістрів. Виходом є XOR трьох регістрів зсуву.

Допоміжні функції пакету MAPLE

`Divide(a, b, 'q')` – перевіряє, чи ділиться многочлен a на многочлен b без залишку. В змінну q поміщається результат ділення у випадку успіху.

```
Divide(x^7+1, x^3+x+1, 'q') mod 2;
true
q;
x^4+x^2+x+1
Divide(x^15+1, x^4+x^3+x^2+1, 'q') mod 2;
false
```

Для формування лінійної рекурентної послідовності $x_{k+3} = x_{k+2} \oplus x_k$ глибини $n=3$ з початковими значеннями $x_1=1, x_2=0, x_3=1$ можна використати цикл:

```
x[1]:=1;x[2]:=0;x[3]:=1;
for k from 1 to 20 do
x[k+3]:=x[k+2]+x[k] mod 2
end do;
```

```
x1:=1 x2:=0 x3:=1 x4:=0 x5:=0 x6:=1 x7:=1 x8:=1
x9:=0 x10:=1 x11:=0 x12:=0 x13:=1 x14:=1 x15:=1 x16:=0
x17:=1 x18:=0 x19:=0 x20:=1 x21:=1 x22:=1 x23:=0
```

Додаток Д

Інтерфейс програми «ГОСТ»

На рисунках Д.1 і Д.2 показані основні елементи інтерфейсу програми «ГОСТ».

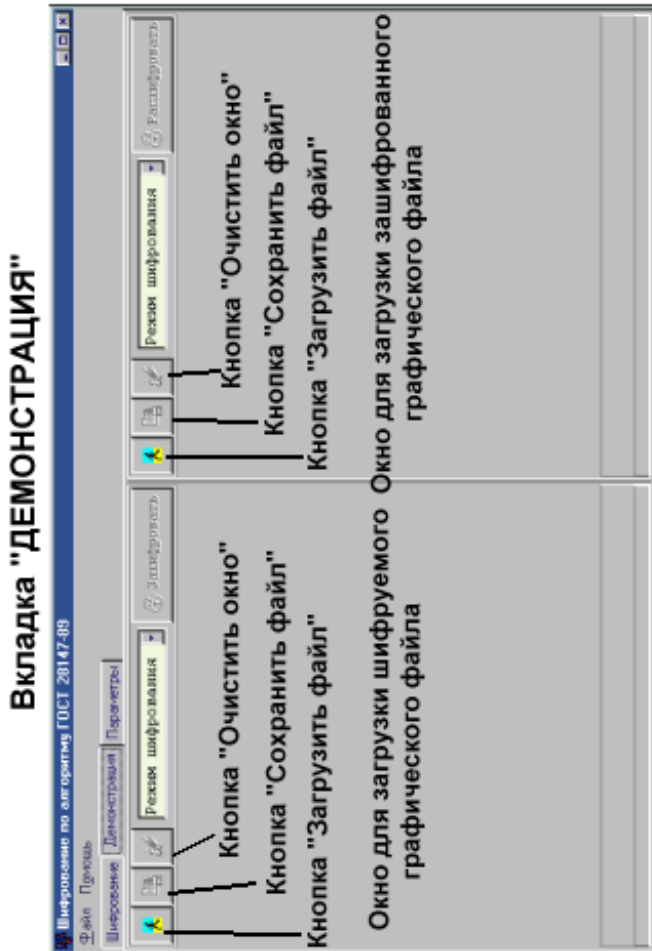


Рисунок Д.1 – Демонстрація



Рисунок Д.2 – Шифрування

Додаток Е

ДСТУ ГОСТ 28147:2009

Інформація про ДСТУ ГОСТ 28147:2009: Державний Комітет України з питань технічного регулювання та споживчої політики, Наказ від 22.12.2008 р. № 495 Про прийняття міждержавних стандартів як національні методом підтвердження та скасування відповідних міждержавних стандартів.

Повна назва — «ДСТУ ГОСТ 28147:2009 Системи обробки інформації. Захист криптографічний. Алгоритм криптографічного перетворення». ДСТУ ГОСТ 28147:2009 замінює наступні документи: ГОСТ 28147-89

ГОСТ 28147-89 — радянський і російський стандарт симетричного шифрування, введений в 1990 році, також є міждержавним стандартом СНД.

ГОСТ 28147-89 — блоковий шифр з 256-бітовим ключем і 32 циклами перетворення, що оперує 64-бітними блоками. Основа алгоритму шифру — сітка Фейстеля. Базовим режимом шифрування за ГОСТ 28147-89 є режим простої заміни (визначені також складніші режими гамування, гамування зі зворотним зв'язком і режим імітоприкладки).

Режим простої заміни

В режимі простої заміни кожен блок відкритого тексту при шифруванні і шифртексту при розшифруванні обробляється окремо і проходить 32 раунди сітки Фейстеля. Вхідний блок розбивається на дві 32-бітні частини – ліву (старші біти) L_0 і праву (молодші біти) R_0 . На кожному i -тому раунді ($i = 1, 2, \dots, 32$) відбуваються наступні обчислення:

$$R_i = L_{i-1} \oplus f(K_i, R_{i-1}), \quad L_i = R_{i-1},$$

де $f()$ – раундова функція; K_i – раундовий ключ.

Раундова функція передбачає додавання аргументів за модулем 2^{32} і застосування декількох блоків перестановок. Раундові ключі отримуються із ключа шифрування: 256-бітний ключ K розділюється

на вісімь 32-бітних частин $K_1 - K_8$, які використовуються циклічно в раундах 1-24 і в зворотному порядку в раундах 25-32.

Після виконання всіх раундових перетворень результуючі 32-бітні напівблоки переставляються місцями і об'єднуються у блок шифртексту.

Процедура розшифрування відрізняється від процедури шифрування тільки зворотнім порядком використання раундових ключів.

Режим гамування

В режимі гамування алгоритм ГОСТ працює в якості генератора псевдо випадкової послідовності і може розглядатися як потоковий шифр. При шифруванні та розшифруванні згенерована послідовність – гама шифру – накладається на відкритий або шифртекст за допомогою операції XOR.

В якості стартового значення послідовності використовується синхропосилання S (довільний 64-бітний блок), яке передається разом із шифртекстом. Гамма генерується блоками Γ_i за наступним співвідношенням

$$\Gamma_i = \text{ГОСТ}_K([C_1, C_2] \oplus \Gamma_{i-1}), \Gamma_0 = \text{ГОСТ}_K(S),$$

де ГОСТ_K – шифрування в режимі простої заміни з ключем K ; $C_1 = 1010104_{16}$, $C_2 = 1010101_{16}$ – допоміжні константи.

Режим гамування зі зворотнім зв'язком

В режимі гамування зі зворотнім зв'язком гама шифру ГОСТ залежить не тільки від синхропосилання S і ключа K , а і від попередніх блоків шифртексту Π_i .

В якості стартового значення послідовності використовується синхропосилання S , яке передається разом із шифртекстом. Гамма генерується блоками Γ_i за наступним співвідношенням

$$\Gamma_i = \text{ГОСТ}_K(\Pi_{i-1}), \Gamma_0 = \text{ГОСТ}_K(S).$$

Режим імітоприкладки

На відміну від попередніх режим імітоприкладки спрямований на забезпечення цілісності відкритого тексту, а не його конфіденційності.

Імітоприкладкою називається контрольне значення, яке обчислюється з відкритого тексту згідно з алгоритмом ГОСТ за схемою, зображеною на рис. Е.1. Відкритий текст розподіляється на 64-бітні блоки, перший блок проходить 16 раундів режиму простої заміни, результат накладається на наступний блок за допомогою операції XOR, результат накладання проходить 16 раундів режиму простої заміни і накладається на наступний блок – вихід останніх 16 раундів становитиме імітоприкладку.



Рисунок Е.1 – Вироблення імітоприкладки

Для зручності використовують не весь результуючий блок в якості імітоприкладки, а його частину з M бітів. Про значення M і правила його вибору (наприклад, молодші 32 біти блоку) абоненти, які використовують імітоприкладку при обміні повідомленнями, мають домовитись заздалегідь.

Додаток Ж

Порівняннями першого степеня

Порівняннями першого степеня називаються вирази вигляду

$$a \cdot x = 1 \bmod m,$$

де $a < m$ – задані цілі числа.

Розв'язком порівняння $a \cdot x = 1 \bmod m$ називається ціле число x , $0 < x < m$, що задовольняє йому. Позначається розв'язок порівняння через a^{-1} або $\frac{1}{a}$.

У порівняння першого степеня $a \cdot x = 1 \bmod m$ існує лише один розв'язок в діапазоні від 0 до m і тільки тоді, коли a і m взаємно прості.

В інших випадках розв'язків немає.

Для пошуку розв'язку порівняння $a \cdot x = 1 \bmod m$ можна вдатися до повного перебору, послідовно підставляючи всі можливі значення x до нього, однак більш ефективним способом є розширений алгоритм Евкліда.

Розширений алгоритм Евкліда розв'язання порівняння $a \cdot x = 1 \bmod m$

Знаходимо залишки від ділення за звичайним алгоритмом Евкліда (див. Додаток Б):

$$m = a \cdot q_0 + r_1$$

$$a = r_1 \cdot q_1 + r_2$$

$$r_1 = r_2 \cdot q_2 + r_3$$

$$r_2 = r_3 \cdot q_3 + r_4$$

.....

$$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n$$

$$r_{n-1} = r_n \cdot q_n$$

Заповнюємо таблицю Ж.1, використовуючи отримані значення.

Таблиця Ж.1 – Пошук розв’язку порівняння першого степеня

i		0	1	2	...	$n-1$	n
q_i		q_0	q_1	q_2	...	q_{n-1}	q_n
P_i	1	P_0	P_1	P_2	...	P_{n-1}	P_n

$$P_0 = q_0, \quad P_1 = q_1 \cdot P_0 + 1,$$

$$P_i = q_i \cdot P_{i-1} + P_{i-2}, \quad i \geq 2.$$

Якщо таблицю Ж.1 заповнено вірно, P_n має дорівнювати m :
 $P_n = m$.

Розв’язок порівняння отримуємо з таблиці Ж.1 за формулою
 $x = (-1)^n \cdot P_{n-1} \bmod m$.

Для розв’язання порівняння

$$a \cdot x = b \bmod m$$

спочатку знаходять розв’язок порівняння

$$a \cdot z = 1 \bmod m,$$

тобто $z = \frac{1}{a} \bmod m$, а потім формують розв’язок порівняння

$$a \cdot x = b \bmod m:$$

$$x = z \cdot b \bmod m.$$

Приклад 1.

Знайдемо розв'язок порівняння $9 \cdot x = 1 \bmod 52$.

Розв'язок порівняння існує, оскільки 9 і 52 взаємно прості.

За алгоритмом Евкліда знаходимо залишки від ділення:

$$52 = 9 \cdot 5 + 7$$

$$9 = 7 \cdot 1 + 2$$

$$7 = 2 \cdot 3 + 1$$

$$2 = 1 \cdot 2$$

Заповнимо таблицю

i		0	1	2	3
q_i		5	1	3	2
P_i	1	5	6	23	52

З отриманої таблиці обчислимо розв'язок:

$$x = (-1)^3 \cdot 23 \bmod 52 = -23 \bmod 52 = 29,$$

$$x = 29.$$

$$\text{Перевірка: } 9 \cdot 29 = 261 \bmod 52 = 1.$$

Приклад 2.

Знайдемо розв'язок порівняння $9 \cdot x = 10 \bmod 52$.

Спочатку знайдемо розв'язок порівняння $9 \cdot z = 1 \bmod 52$.

Згідно з Прикладом 1, $z = 29$.

Звідси розв'язком порівняння $9 \cdot x = 10 \bmod 52$ є

$$x = 29 \cdot 10 = 290 \bmod 52 = 30,$$

$$x = 30.$$

$$\text{Перевірка: } 9 \cdot 30 = 270 \bmod 52 = 10.$$

Додаток К

Алгоритм послідовного зведення в квадрат

В багатьох криптоалгоритмах, зокрема RSA, виникає потреба обчислювати вирази вигляду

$$b = a^k \bmod m.$$

При великих значеннях k це може бути обчислювально складно. Для оптимізації обчислень може бути використаний алгоритм послідовного зведення в квадрат.

Алгоритм послідовного зведення в квадрат

1. Розкладемо k по ступенях двійки:

$$k = k_1 + k_2 \cdot 2 + k_3 \cdot 2^2 + \dots + k_{t+1} \cdot 2^t$$

2. $b := 1$.

3. $A := a$.

4. Якщо $k_1 = 1$, тоді $b := a$.

5. Цикл по i от 2 до t

- а. $A := A^2 \bmod m$

- б. Якщо $k_i = 1$, тоді $b := b \cdot A$

6. Повертаємо b .

Приклад 1.

Обчислимо $b = 3^{42} \bmod 79$.

Розкладемо $k = 42$ по ступенях двійки:

$$k = 0 + 1 \cdot 2 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5.$$

$$b := 1.$$

$$A := 3.$$

$$k_1 = 0.$$

$$k_2 = 1, A := 3^2 \bmod 79 = 9, b := 1 \cdot 9 \bmod 79 = 9.$$

$$k_3 = 0, A := 9^2 = 81 \bmod 79 = 2.$$

$$k_4 = 1, A := 2^2 \bmod 79 = 4, b := 9 \cdot 4 \bmod 79 = 36.$$

$$k_5 = 0, A := 4^2 \bmod 79 = 16.$$

$$k_6 = 1, A := 16^2 = 256 \bmod 79 = 19, b := 36 \cdot 19 = 684 \bmod 79 = 52.$$

$$\text{Таким чином, } b = 3^{42} \bmod 79 = 52.$$

Приклад 2.

$$\text{Обчислимо } b = 2^{130} \bmod 35.$$

Обчислення b доцільно знайти за допомогою теореми Ейлера:

оскільки числа 2 і 35 взаємно прості, то $2^{\varphi(35)} = 1 \bmod 35$.

Знайдемо значення функції Ейлера:

$$\varphi(35) = \varphi(5) \cdot \varphi(7) = 4 \cdot 6 = 24.$$

$$\text{Звідси } 2^{24} = 1 \bmod 35.$$

Оскільки $130 = 24 \cdot 5 + 10$, то

$$b = 2^{130} = \underbrace{2^{24}}_{=1}^5 \cdot 2^{10} \bmod 35 = 1^5 \cdot 2^{10} \bmod 35 = 2^{10} \bmod 35.$$

Таким чином, для обчислення $b = 2^{130} \bmod 35$ достатнє знайти $b = 2^{10} \bmod 35$.

Розкладемо $k = 10$ по ступенях двійки:

$$k = 0 + 1 \cdot 2 + 0 \cdot 2^2 + 1 \cdot 2^3.$$

$$b := 1.$$

$$A := 3.$$

$$k_1 = 0.$$

$$k_2 = 1, A := 2^2 \bmod 35 = 4, b := 1 \cdot 4 \bmod 35 = 4.$$

$$k_3 = 0, A := 4^2 \bmod 35 = 16.$$

$$k_4 = 1, A := 16^2 = 256 \bmod 35 = 11, b := 4 \cdot 11 = 44 \bmod 35 = 9.$$

$$\text{Звідси, } b = 2^{10} \bmod 35 = 9 \text{ і } b = 2^{130} \bmod 35 = 9.$$

Додаток Л

Тест на простоту

Нерідко певні параметри криптографічних алгоритмів мають бути простими числами, інакше їхня криптостійкість істотно знижується. Якщо відносно малі числа можна перевірити на простоту за заздалегідь сформованою таблицею простих чисел, то для великих чисел створити таку таблицю проблематично.

Існують різні методи перевірити, чи є число простим, що базуються на властивостях простих чисел. Так, зокрема, мала теорема Ферма (див. Додаток Б) стверджує, що для простого числа p виконується співвідношення $a^{p-1} = 1 \bmod p$ для всіх $a < p$.

На цій теоремі будується тест Ферма на простоту, який передбачає пошук і підрахунок свідків простоти числа.

Свідком простоти непарного числа n називається таке число a , $a < n$, взаємно просте з ним ($\text{НСД}(a, n) = 1$), для якого виконується умова $a^{n-1} = 1 \bmod n$.

Тест Ферма

Нехай дано число n .

1. Вибираємо випадкове ціле a , $1 < a < n$.
2. Знаходимо найбільший спільний дільник чисел a і n – $\text{НСД}(a, n)$ – за допомогою алгоритму Евкліда.
3. Якщо $\text{НСД}(a, n) > 1$, повертаємо " n – складене".
4. Якщо $\text{НСД}(a, n) = 1$, обчислюємо $v = a^{n-1} \bmod n$ за допомогою алгоритму послідовного зведення в квадрат (див. Додаток К).
5. Якщо $v > 1$, повертаємо " n – складене".
6. Якщо $v = 1$, то число a є свідком простоти.
7. Повторюємо пункти 1-6 доти, поки не наберемо t свідків простоти.

Тест Ферма не дає стовідсоткової відповіді, чи є число n простим. Кількість знайдених свідків простоти визначає ймовірність того, що число n є простим (або, відповідно, складеним).

Якщо для числа n знайдено t свідків простоти, то ймовірність того, що число n складене не перевищує 2^{-t} (або, відповідно, ймовірність того, що воно просте, не менша $1 - 2^{-t}$).

Зокрема, існують складені числа K , які проходять тест Ферма, тобто всі числа a менші K і взаємно прості з ним є свідками простоти числа K . Такі числа називають числами Кармайкла⁵.

Числа Кармайкла представляють собою добуток принаймні трьох різних простих множників, відмінних від 1. Найменше число Кармайкла дорівнює $561 = 3 \cdot 11 \cdot 17$.

Числами Кармайкла є 1729, 2465, 172081, 294409, 56052361.

Приклад 1.

Знайти всіх свідків простоти числа 21.

Числа 1 і $n-1$ завжди є свідками простоти числа n . Знайдемо свідків простоти серед інших чисел, менших 21.

Взаємно простими з числом 21 є числа 2, 4, 5, 8, 10, 11, 13, 16, 17, 19. Обчислимо для цих чисел значення $a^{20} \bmod 21$:

$$2^{20} \bmod 21 = 4 \neq 1, 19^{20} \bmod 21 = 4 \neq 1$$

$$4^{20} \bmod 21 = 16 \neq 1, 17^{20} \bmod 21 = 16 \neq 1$$

$$5^{20} \bmod 21 = 4 \neq 1, 16^{20} \bmod 21 = 4 \neq 1$$

$$8^{20} \bmod 21 = 1, 13^{20} \bmod 21 = 1$$

$$10^{20} \bmod 21 = 16 \neq 1, 11^{20} \bmod 21 = 16 \neq 1$$

Свідками простоти числа 21 є 8 і 13, окрім 1 і 20.

Приклад 2.

Визначити, чи є число 33 простим.

Візьмемо $a_1 = 10$, взаємно просте з 33: $10^{32} \bmod 33 = 1$.

Число $a_1 = 10$ є свідком простоти числа 33.

Візьмемо $a_2 = 14$, взаємно просте з 33: $14^{32} \bmod 33 = 31$.

Число $a_2 = 14$ не є свідком простоти числа 33.

Звідси по тесту Ферма випливає, що число 33 – складене.

⁵ Robert Daniel Carmichael (1879 –1967)

Додаток М

Розкладання чисел на множники

Однією з важкорозв'язувальних задач, яка забезпечує стійкість криптоалгоритмів (наприклад, RSA), є факторизація чисел – розкладання їх на множники.

Будь-яке натуральне число, відмінне від 1, можна представити у вигляді добутку простих чисел, причому, існує всього один варіант такого представлення.

Канонічним розкладанням числа n на множники називають представлення його у вигляді наступного добутку з k елементів

$$n = \prod_{i=1}^k p_i^{\alpha_i} = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_k^{\alpha_k},$$

де $p_1, p_2, \dots, p_k$ – попарно різні прості числа, $\alpha_1, \alpha_2, \dots, \alpha_k$ – натуральні числа.

Для знаходження канонічного розкладання можна використовувати перебірний алгоритм, який полягає у послідовних спробах поділити число n на прості числа, менші за нього.

Для будь-якого складеного n існує хоча б один простий дільник, що не перевищує $\sqrt{n}$, тому підбирати прості числа слід в діапазоні від 2 до $\sqrt{n}$.

Однак, для факторизації великих чисел перебірний алгоритм виявляється неефективним, тому для розв'язання такої задачі використовуються більш ефективні алгоритми, наприклад, алгоритм ρ -Полларда.

Алгоритм ρ -Полларда факторизації цілих чисел

1. Визначаємо початкові значення $a := 2, b := 2$.

2. Виконуємо в циклі

$$a := (a^2 + 1) \bmod n; \quad b := (b^2 + 1) \bmod n, \quad b := (b^2 + 1) \bmod n; \\ d := \text{НСД}(a - b, n).$$

Якщо $d = 1$, продовжуємо цикл.

Якщо $1 < d < n$, повертаємо « d – дільник числа n ».

Якщо $d = n$, алгоритм закінчує роботу невдачею.

Приклад.

Знайти за допомогою алгоритму ρ -Полларда дільник числа $n = 187$.

$$a := 2, b := 2.$$

$$i = 1,$$

$$a := (2^2 + 1) \bmod 187 = 5; b := (2^2 + 1) \bmod 187 = 5,$$

$$b := (2^2 + 1) \bmod 187 = 26; d := \text{НСД}(5 - 26, 187) = 1.$$

$$i = 2,$$

$$a := (2^2 + 1) \bmod 187 = 26, b := (2^6 + 1) \bmod 187 = 116,$$

$$b := (16^2 + 1) \bmod 187 = 180;$$

$$d := \text{НСД}(26 - 180, 187) = \text{НСД}(-154, 187) = 11.$$

Звідси $d = 11$ – дільник числа 187.

Допоміжні функції пакету MAPLE

Алгоритм ρ -Полларда може бути реалізований в такій програмі:

```
n:=187; a:=2; b:=2; d:=1; i:=1;
while (d=1 and i<10) do
a:=a^2+1 mod n; b:=b^2+1 mod n; b:=b^2+1 mod n;
d:=gcd(a-b,n); i:=i+1; end do;
```

$$i := 1$$

$$a := 5 \quad b := 5 \quad b := 26 \quad d := 1$$

$$i := 2$$

$$a := 26 \quad b := 116 \quad b := 180 \quad d := 11$$