ЗАТВЕРДЖЕНО

Науково-методичною радою
Державного університету
«Житомирська політехніка»

протокол від 29 червня 2023 р.
№9

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
для проведення лабораторних занять з навчальної дисципліни «Інформаційна безпека та захист ПЗ»

для здобувачів вищої освіти освітнього ступеня «бакалавр»
спеціальності код 121 «Інженерія програмного забезпечення»
освітньо-професійна програма «Інженерія програмного забезпечення»
факультет інформаційно-комп'ютерних технологій
(назва факультету)
кафедра комп'ютерної інженерії та кібербезпеки
(назва кафедри)

Рекомендовано на засіданні
кафедри комп'ютерної
інженерії та кібербезпеки
(назва кафедри)
8 листопада 2022 р.,
протокол № 7

Розробник: старший викладач кафедри комп'ютерної інженерії та кібербезпеки ЩУР Наталія
(науковий ступінь, посада, ПРІЗВИЩЕ, власне ім'я)

Житомир
2023

ЗМІСТ

ВСТУП

Сучасні інформаційно-комунікаційні технології інтенсивно впроваджуються в усі сфери людського життя. Інформаційні ресурси стають головною цінністю наукового, економічного та технічного розвитку будь-якої галузі як в Україні, так і у світі. При цьому великого значення набуває проблема захисту даних, що полягає у забезпеченні їх конфіденційності, цілісності та достовірності при зберіганні, обробці та передачі. Постає стратегічно важливе питання якості підготовки закладами вищої освіти майбутніх ІТ-фахівців, які б у своїй діяльності ефективно використовували різноманітні методи захисту інформації, зокрема криптографічні.

Лабораторні роботи з курсу «Інформаційна безпека та захист ПЗ» мають на меті закріплення у майбутніх ІТ-фахівців теоретичних знань здобутих на лекційних заняттях, формування та набуття професійних компетенцій, практичних знань та вмінь з криптографічного захисту інформаційних ресурсів та криптографічного аналізу.

За результатами вивчення дисципліни студенти повинні вміти: застосовувати сучасне криптографічне програмне забезпечення, проектувати та програмно реалізовувати прості алгоритми шифрування, проводити найпростіший криптоаналіз класичних шифрів.

Для забезпечення кращої ефективності навчання пропонується використовувати вільно поширюване програмне забезпечення із захисту інформаційних ресурсів, що сприятиме різнобічному і змістовному вивченню відповідної предметної галузі, відкриє нові пізнавальні можливості та перспективи для підвищення рівня знань студентів, допоможе їм легко засвоїти складні принципи та технології криптографічних перетворень на практиці.

ТЕМА № 1. КЛАСИЧНИЙ ШИФР ПРОСТОЇ ЗАМІНИ ТА ЙОГО КРИПТОАНАЛІЗ. БІГРАМНИЙ ШИФР

Мета роботи: набути вміння із зашифрування та дешифрування повідомлень за допомогою шифру простої заміни, зокрема шифру Цезаря; використовуючи частотний криптоаналіз, навчитися зламувати шифротекст, зашифрований методом простої заміни; навчитися шифруванню біграмним шифром Плейфера.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням MS Excel та доступом до мережі Інтернет, текстові повідомлення для шифрування згідно варіанту.

Теоретичні відомості

ШИФР ЦЕЗАРЯ

Розглянемо один з найдавніших та найбільш поширених шифрів простої (моноалфавітної) заміни – шифр Цезаря, названий на честь римського імператора *Гая Юлія Цезаря*. У цьому шифрі кожна літера повідомлення зсувається в алфавіті на *K* позицій вперед від символу, що замінюється. При досягненні кінця алфавіту виконується циклічний перехід до його початку. При необхідності розділові знаки та пробіли ігноруються. Таким чином, наприклад, літерам алфавіту відповідатимуть числові позиції (табл. 1.1, табл. 1.2):

Таблиця. 1.1. Нумерація позицій літер англійського алфавіту

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Таблиця. 1.2. Нумерація позицій літер українського алфавіту

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Ключем шифрування є деяке фіксоване секретне число *K* – від 1 до 25 для англійського (латинського) алфавіту та *K* – від 1 до 32 для українського. При дешифруванні літера зашифрованого тексту замінюється на літеру розташовану в алфавіті на *K* позицій назад.

Приклад 1.1:

Відомо, що Цезар для шифрування використовував ключ K=3, тобто відбувався зсув символів повідомлення на три позиції вперед у латинському алфавіті (рис. 1.1). Отже, повідомлення римського імператора *ALEA JACTA EST* (Жереб кинутий) після зашифрування буде мати вигляд *DOHDMDFWDHVW*.

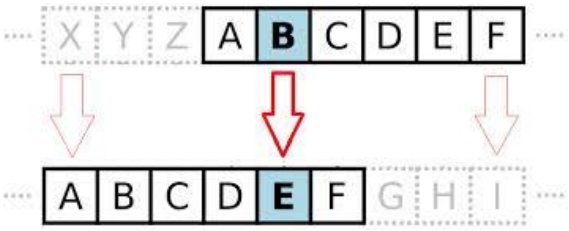


Рис. 1.1. Заміна символів повідомлення у шифрі Цезаря з ключем K=3

Зазначимо, що цей алгоритм шифрування, на сьогоднішній день, являється нестійким до зламу і не використовується на практиці, проте є важливим для вивчення. Оскільки відомо, що навіть дуже складні сучасні криптосистеми в якості типових складових використовують прості шифри заміни.

ЧАСТОТНИЙ КРИПТОАНАЛІЗ

Криптоаналіз шифру Цезаря ґрунтується на *частотному аналізі* появи окремих символів природньої мови у тексті. Частота символу у повідомленні дорівнює кількості його появи у тексті, поділеній на загальну кількість літер тексту. Для кожної мови справедливо наступне: у досить довгих текстах кожна літера зустрічається із приблизно однаковою частотою, залежно від самої літери і незалежно від конкретного тексту. Тобто імовірність появи окремих літер, а також їх порядок у словах і фразах природньої мови підпорядковуються статистичним закономірностям. Так, наприклад, відомо, що в українській та англійській мовах частоти появи літер розподілені наступним чином (табл. 1.3).

Отже, літера з найбільшою частотою в шифротексті буде замінюватися на літеру з найбільшою частотою у мові. А кількість позицій між ними буде визначати довжину ключа. Однак, якщо текст не дуже великий, то закономірності будь-якої природньої мови можуть проявлятися в ньому не обов’язково в строгій відповідності з таблицею частот. В такому випадку

розглядається відношення наступної літери за частотою появи у зашифрованому тексті та найчастішою літерою мови.

Таблиця. 1.3. Частоти появи літер в українській та англійській мовах

Українська мова						**Англійська мова**					
А	0,072	**Ї**	0,006	**У**	0,04	**A**	0,082	**J**	0,002	**S**	0,063
Б	0,017	**Й**	0,008	**Ф**	0,001	**B**	0,015	**K**	0,008	**T**	0,091
В	0,052	**К**	0,035	**Х**	0,012	**C**	0,028	**L**	0,040	**U**	0,028
Г, Ґ	0,016	**Л**	0,036	**Ц**	0,006	**D**	0,043	**M**	0,024	**V**	0,010
Д	0,035	**М**	0,031	**Ч**	0,018	**E**	**0,127**	**N**	0,067	**W**	0,023
Е	0,017	**Н**	0,065	**Ш**	0,012	**F**	0,022	**O**	0,075	**X**	0,001
Є	0,008	**О**	**0,094**	**Щ**	0,001	**G**	0,020	**P**	0,019	**Y**	0,020
Ж	0,009	**П**	0,029	**Ь**	0,029	**H**	0,061	**Q**	0,001	**Z**	0,001
З	0,023	**Р**	0,047	**Ю**	0,004	**I**	0,070	**R**	0,0060		
И	0,061	**С**	0,041	**Я**	0,029						
І	0,057	**Т**	0,055								

Приклад 1.2:

Дано текст, зашифрований за допомогою шифру моноалфавітної заміни:

ДАФИНЦШЕИЮЯЗЩШФЬИТЧИВЮЯШХСЯЯЗВИЇШЧШЮФЬСПЄСПИЇОЛ РПЧИЦРЗФЬРІИЇЇШЛСЯИФСЦРЇЧЄШЦАСЇШЧШІСХЗЧИЮДАФИНЧИЮЮ ИЦРВЧМЦИУШЙШЧСЛМЇСЛЯШЙШЕШЇШЧШЮФЬСПЄ

При зашифруванні відкритого тексту використовувався алфавіт АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯабвгґдеєжзиіїйклмнопрсту фхцчшщьюя. Припускаючи, що текст зашифрований за допомогою шифру Цезаря, складемо таблицю появи літер в даному шифротексті (табл. 1.4).

Таблиця. 1.4. Зустрічальності літер у шифротексті

А	**Б**	**В**	**Г, Ґ**	**Д**	**Е**	**Є**	**Ж**	**З**	**И**	**І**	**Ї**	**Й**	**К**	**Л**	**М**
3	0	3	0	2	2	3	0	4	14	2	8	2	0	4	2
Н	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Ю**	**Я**
2	1	4	5	10	1	1	7	2	6	10	16	1	4	7	6

З табл. 1.4 видно, що найчастіше у тексті з’являється літера «Ш» – 16 разів. А з табл. 1.3 відомо, що найчастіше в текстах українською мовою зустрічається літера «О». Тому можемо припустити, що літері «Ш» в шифротексті, ймовірно, відповідає літера «О» у відкритому тексті. Якщо послідовності літер А, Б,…, О,…, Ш,…, Я ототожнити із послідовністю їх позицій в алфавіті 0, 1,…, 18,…, 28,…, 33, то можна обчислити ключ *K*: 28–18=10.

Тепер ми можемо відновити початкове повідомлення, записавши його із розділовими знаками: *Шукаємо щастя по країнах, століттях, а воно скрізь і завжди з нами; як риба в воді, так і ми в ньому, і воно біля нас шукає нас самих. Нема його ніде від того, що воно скрізь.*

ШИФР ПЛЕЙФЕРА

Шифр Плейфера є біграмним, тобто текст повідомлення розбивається на біграми (групи з двох символів). Таким чином, шифр Плейфера є більш стійкий до зламу у порівнянні із шифром простої заміни, так як ускладнюється його частотний аналіз. Він може бути проведений, але не для 26 можливих символів (англійський алфавіт), а для 26x26 = 676 можливих біграм.

Для шифрування шифр Плейфера використовує матрицю 5x5 (для англійського алфавіту), яка містить ключове слово або фразу. Щоб скласти ключову матрицю, в першу чергу потрібно заповнити порожні клітинки матриці літерами ключового слова (виключаючи літери, що повторюються), потім заповнити клітинки, що лишилися символами алфавіту, що не зустрічаються в ключовому слові, по порядку (рис. 1.2). В англійських текстах зазвичай пропускається символ «Q», щоб зменшити алфавіт, в інших версіях «I» і «J» об’єднуються в одну клітинку.

P	L	A	Y	F
I	R	E	X	M
B	C	D	G	H
J	K	N	O	S
T	U	V	W	Z

Рис. 1.2. Матриця шифру Плейфера

Ключове слово може бути записано у верхньому рядку матриці зліва направо, або по спіралі з лівого верхнього кута до центру.

Для того щоб зашифрувати повідомлення, необхідно розбити його на біграми (групи з двох символів) та відшукати ці біграми в матриці. Два символи біграми відповідають кутам прямокутника в ключовий матриці. Визначаємо

положення кутів цього прямокутника відносно один одного. Потім, керуючись наступними 4 правилами, зашифровувати пари символів вихідного тексту.

Правила шифрування біграм

1. Якщо дві літери біграми однакові – додаємо після першого символу «X», зашифровуємо нову пару літер.
2. Якщо літери біграми знаходяться в різних стовпцях і різних рядках – замінюємо їх на літери, що знаходяться в тих самих рядках (стовпцях), але відповідно в інших кутах прямокутника.
3. Якщо літери біграми зустрічаються в одному рядку – замінюємо їх на літери, розташовані в найближчих стовпцях праворуч від відповідних літер. Якщо літера остання у рядку, то вона замінюється на перший символ цього ж рядка.
4. Якщо літери біграми зустрічаються в одному стовпці – перетворюємо їх в літери того ж стовпця, що знаходяться безпосередньо під ними. Якщо літера є нижньою в стовпці – вона замінюється на першу літеру цього ж стовпчика.

Приклад 1.3:

Зашифруємо повідомлення HIDE THE GOLD IN THE TREE STUMP із використанням ключової фрази PLAYFAIR EXAMPLE. Матрицею шифрування буде матриця описана вище (рис. 1.2).

Для шифрування розіб’ємо текст на біграми HI DE TH EG OL DI NT HE TR EX ES TU MP. Знайдемо літери першої біграми у матриці та замінимо їх на літери, що знаходяться у протилежних кутах прямокутника (рис. 1.3).

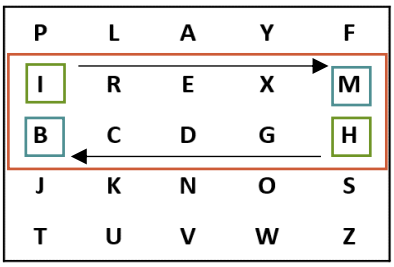


Рис. 1.3. Шифрування біграм

Далі, користуючись правилами шифрування біграм, отримаємо шифротекст: BM ND ZB XD KY BE JV DM UI XM MN UV IF.

Завдання до лабораторної роботи

Завдання 1

Завдання виконується індивідуально кожним студентом. Усі необхідні обчислення зі скріншотами описуються у звіті.

Створити програму в середовищі *MS Excel* або на будь-якій мові програмування для шифрування повідомлень із використанням шифру Цезаря (англійській алфавіт). Значення ключа шифрування визначається номером за алфавітним списком студента у журналі. Зашифрувати своє прізвище та дешифрувати отриманий шифротекст. Зразок виконання завдання наведено на рисунку нижче (рис. 1.4).

A10 =HLOOKUP(A9;$A$1:$Z$2;2;FALSE)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4																										
5																										
6	**Key**	10																								
7																										
8	**Encryption**													**Decryption**												
9	**C**	**R**	**Y**	**P**	**T**	**O**								**M**	**B**	**I**	**Z**	**D**	**Y**							
10	2	17	24	15	19	14								12	1	8	25	3	24							
11	12	27	34	25	29	24								2	-9	-2	15	-7	14							
12	12	1	8	25	3	24								2	17	24	15	19	14							
13	**M**	**B**	**I**	**Z**	**D**	**Y**								**C**	**R**	**Y**	**P**	**T**	**O**							
14																										

Рис. 1.4. Шифрування шифром Цезаря в середовищі MS Excel

Завдання 2

Студентам потрібно поділитися на ротаційні групи з трьох чоловік: СТУДЕНТ-ВІДПРАВНИК, СТУДЕНТ-ОТРИМУВАЧ, СТУДЕНТ-КРИПТОАНАЛІТИК. Обмін повідомленнями між учасниками відбуватиметься за схемою (рис. 1.5), в основі якої лежить секретна система зв'язку, описана Клодом Шеноном.

Рис. 1.5. Схема обміну повідомленнями між студентами

2.1. На сайті *CrypTool Online* – https://www.cryptool.org/en/cto/ з використанням шаблону *Caesar* виконати шифрування тексту шифром Цезаря згідно варіанту. Спочатку введіть відкритий текст до поля ***Input***, потім визначте алфавіт за допомогою опції ***Define own alphabet***, ключ шифрування оберіть самостійно.

Варіант №	**Відкритий текст**
1.	Єдино можливий порядок розташування знаків надає їм, знакам, ваги символів. Абетка є цілісною і до кінця заповненою даністю. Вона не зрадить і навіть не зміниться. Юрій АНДРУХОВИЧ
2.	Вікно відкрите дивиться у сад, де від дощу піднялись буйно трави. І день, що розпочатий так, навгад, приносить спокій тихий і ласкавий. Марта КАЛИТОВСЬКА
3.	Якщо не можна вітер змалювати, прозорий вітер на ясному тлі, змалюй дуби, могутні і кислаті, котрі од вітру гнуться до землі. Ліна КОСТЕНКО
4.	Блаженний муж, що серед ґвалту й гуку стоїть, як дуб посеред бур і грому, на згоду з підлістю не простягає руку, волить зламатися, ніж поклониться злому. Іван ФРАНКО
5.	Якщо маєш в душі бодай зернину віри в диво, воно приходить до тебе саме – рано чи пізно, в горі чи радості, в темряві чи у світлі. Бодай раз у житті воно виростає перед тобою, мов свіжий трояндовий кущ, і обдає своїм запаморочливим і справжнім ароматом. Ірен РОЗДОБУДЬКО
6.	Пори року існують для того, щоб ніколи не набриднути, тому їх так скоро забуваємо. Вже через певний час стираються риси попереднього сезону, і осінь наступного року буде такою ж вражаючою, як і минулого. Тарас ПРОХАСЬКО

Варіант №	**Відкритий текст**
7.	Найбільше і найдорожче добро в кожного народу – це його мова, ота жива схованка людського духу, його багата скарбниця, в яку народ складає і своє давнє життя, і свої сподівання, розум, досвід, почування. Панас МИРНИЙ
8.	І все то те, вся країна, повита красою, зеленіє, вмивається дрібною росою, споконвіку вмивається, сонце зустрічає... І нема тому починy, і краю немає! Тарас ШЕВЧЕНКО
9.	Боротьба захлинулася, але, хай там що, мусила мати продовження. З останніх сил, з останнього зубовного скреготу. Бо жодна катастрофа не ставить хрест на меті. Василь ШКЛЯР
10.	Я просто знаю, що все це варте зусиль і печалі. І що ми недаремно себе до цього привчали. І що всім, хто не відступиться, ще буде сходити радість тихими ранками, золотими ночами. Сергій ЖАДАН
11.	Це вже доля, а долю не обирають. Отож її приймають, яка вона вже є. А коли не приймають, тоді вона силоміць обирає нас. Василь СТУС
12.	Люди оточують нас, як повітря, що ми його вдихаємо, щоб жити. Звуки їхніх голосів лунають для нас вічною музикою життя. Рідні обличчя сяють для нас, як маленькі сонця. Павло ЗАГРЕБЕЛЬНИЙ
13.	Слово – найтонший дотик до серця; воно може стати і ніжною запашною квіткою, і живою водою, що повертає віру в добро, і гострим ножем, і розпеченим залізом, і брудом. Василь СУХОМЛИНСЬКИЙ
14.	А душа, це все на світі, що потрібно для життя. Роби свою справу чесно, з душею, – і твоє до тебе прийде. За будь-яких обставин головне – залишатися людиною. Богдан СТУПКА
15.	Мова – це не просто спосіб спілкування, а щось більш значуще. Мова – це всі глибинні пласти духовного життя народу, його історична пам’ять, найцінніше надбання віків. Олесь ГОНЧАР

2.2. Додати скріншот зашифрування до звіту (рис. 1.6).

2.3. Зберегти отриманий шифротекст до текстового документу та обмінятися файлами із шифротекстом зі студентом своєї ротаційної групи. Заздалегідь таємно узгодити довжину ключа шифрування.

2.4. Аналогічно до п.2.1 виконати дешифрування повідомлення одногрупника із використанням шифру Цезаря, увівши шифротекст до відповідного текстового поля. При цьому потрібно встановити перемикач у положення ***Decipher***.

2.5. Додати до звіту скріншот дешифрування повідомлення.

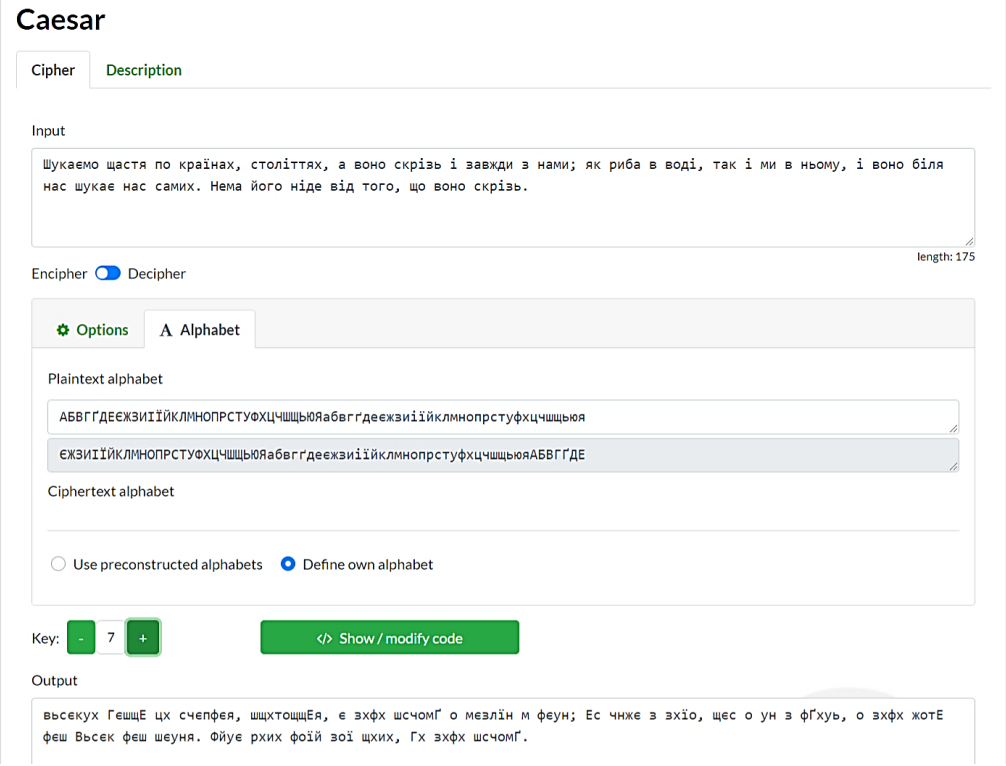


Рис. 1.6. Зашифрування шифром Цезаря

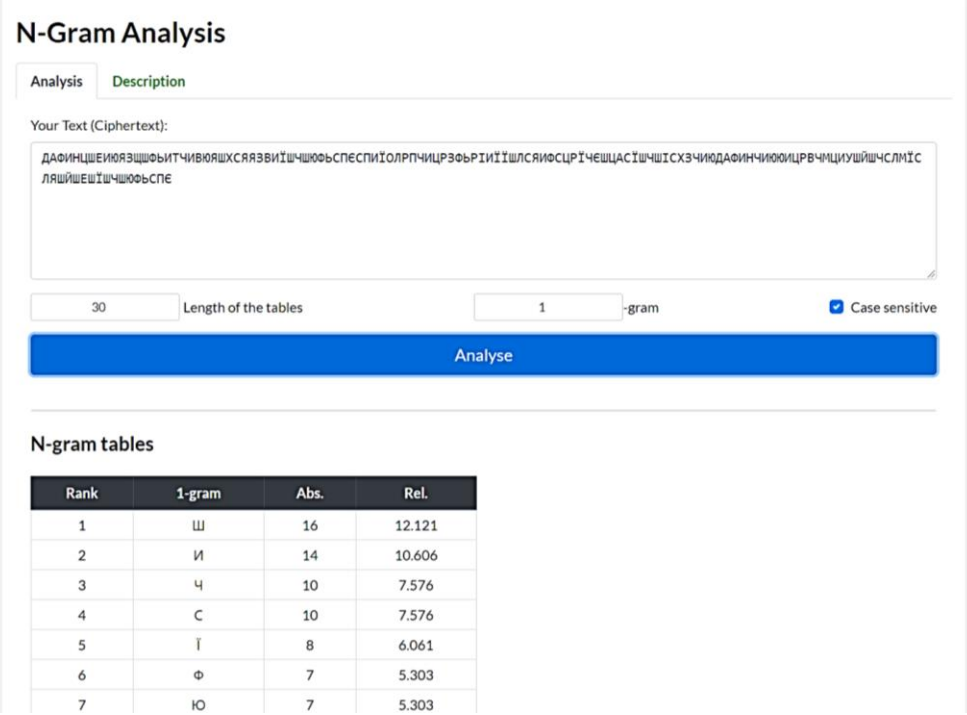
2.6. Обмінятися повідомленнями із шифротекстом з іншим студентом своєї ротаційної групи. При чому, довжина ключа шифрування повинна триматися в таємниці.

2.7. Підрахувати частоти зустрічальності літер у шифротексті одногрупника, використовуючи шаблон *N-Gram Analysis* в розділі криптоаналізу на сайті https://www.cryptool.org/en/cto/ (рис. 1.7).

2.8. Додати до звіту таблицю частоти зустрічальності літер.

2.9. На основі частоти зустрічальності літер у шифротексті підібрати значення ключа, обґрунтувавши свої дії у звіті.

2.10.Ввести шифротекст та значення підібраного ключа на сайті https://www.cryptool.org/en/cto/ з використанням шаблону *Caesar* та відновити повідомлення. Додати до звіту скріншот відновленого повідомлення.



Rank	1-gram	Abs.	Rel.
1	Ш	16	12.121
2	И	14	10.606
3	Ч	10	7.576
4	С	10	7.576
5	Ї	8	6.061
6	Ф	7	5.303
7	Ю	7	5.303

Рис. 1.7. Підрахунок частоти появи літер у тексті

Завдання 3

Виконати зашифрування повідомлення шифром Плейфера згідно варіанту (визначається номером студента у журналі: непарний – 1 варіант, парний – 2 варіант). Усі кроки алгоритму шифрування виконати вручну та описати їх у звіті.

1. Відкритий текст LITTLE STROKES FELL GREAT OAKS зашифруйте за допомогою шифру Плейфера, використовуючи ключ TRUTH.
2. Відкритий текст TILL FINAL VICTORY зашифруйте за допомогою шифру Плейфера, використовуючи ключ LIFE.

Контрольні запитання:

1. Що таке криптографічний алгоритм та шифр?
2. Що таке криптографічний ключ?
3. Назвіть складові криптографічної системи.
4. У чому полягає криптостійкість криптографічної системи?
5. Опишіть алгоритм шифрування Цезаря.
6. У чому суть методу частотного криптоаналізу?
7. Опишіть алгоритм шифру Плейфера.
8. Що є ключем у шифрі Плейфера?

ТЕМА № 2. КЛАСИЧНИЙ ШИФР ПОЛІАЛФАВІТНОЇ ЗАМІНИ ТА ЙОГО КРИПТОАНАЛІЗ. КРИПТОСИСТЕМА ХІЛЛА

Мета роботи: набути вміння із шифрування повідомлень за допомогою шифру поліалфавітної заміни, зокрема шифру Віженера; використовуючи методи Казіскі та Фрідмана, навчитися зламувати шифротекст, зашифрований методом поліалфавітної заміни; навчитися шифрувати повідомлення у криптосистемі Хілла.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням MS Excel та доступом до мережі Інтернет, текстові повідомлення згідно варіанту.

Теоретичні відомості

ШИФР ВІЖЕНЕРА

На протязі століть використання простого моноалфавітного шифру заміни було достатнім, щоб забезпечити таємність. Подальший розвиток частотного криптоаналізу, спочатку арабами, а потім і в Європі, зруйнував його стійкість. Таким чином криптографи мали придумати новий, більш стійкий шифр. Вчений епохи Відродження *Леона Батіста Альберті* вперше запропонував замість одного секретного алфавіту, використовувати два або більше, послідовно або циклічно змінюючи їх за певним правилом. Ґрунтуючись на ідеях попередника, свій шифр створив французький посол в Римі *Блез де Віженер*.

Шифр Віженера складається з послідовності декількох шифрів Цезаря з різними значеннями зсуву, що визначаються літерами ключового слова. Кожна літера відкритого тексту зсувається вперед на позицію відповідної літери ключа. Якщо ключове слово менше за повідомлення, то воно циклічно повторюється.

Приклад 2.1:

Повідомлення *ATTACK AT DAWN* зашифруємо ключем *LEMON*. В результаті чого отримаємо шифротекст *LXFOPVEFRNHR*.

	A	T	T	A	C	K	A	T	D	A	W	N
	L	E	M	O	N	L	E	M	O	N	L	E
	0	19	19	0	2	10	0	19	3	0	22	13
+	11	4	12	14	13	11	4	12	14	13	11	4
	11	23	5	14	15	21	4	5	17	13	7	17
	L	X	F	O	P	V	E	F	R	N	H	R

Для зашифровування може використовуватися й таблиця, яка отримала назву таблиця Віженера (таб.2.1). У загальному випадку таблиця Віженера складається з алфавіту, циклічно зміщеного на один символ ліворуч. Під час зашифрування кожна літера повідомлення замінюється на літеру, що знаходиться на перетині літер першого рядка (алфавіт повідомлення) і першого стовпчика (алфавіт ключа) в таблиці Віженера.

Таблиця. 2.1. Таблиця Віженера

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Приклад 2.2:

Повідомлення *PURPLE*, зашифроване ключем *SMART* за допомогою таблиці Віженера (табл. 2.2), перетвориться у шифротекст *HGRGEW*.

При дешифруванні потрібно відшукати у першому стовпчику літеру ключа і за літерами шифротексту визначити, в якому стовпчику зверху знаходиться літера відкритого тексту.

Таблиця. 2.2. Шифрування повідомлення за таблицею Віженера

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

МЕТОД КАЗІСКІ та МЕТОД ФРІДМАНА (індекс збігу)

У 1863 році офіцер прусської армії, майор *Фрідріх Казіскі* запропонував метод зламу поліалфавітного шифру на прикладі шифру Віженера. Метод Казіскі заснований на наступній ідеї: повторення літер в ключі разом з повторенням літер у відкритому тексті дає повторення літер в зашифрованому тексті. Автор прийшов до висновку, що відстань між повтореннями в шифротексту будуть рівні або кратні довжині (періоду) ключа. Щоб знайти довжину ключа виконаємо наступні дії:

1) знайдемо у шифротексті однакові відрізки довжиною не менше трьох символів (зауважмо, що такі однакові відрізки можуть з'явитися в тексті з досить малою ймовірністю);
2) визначимо відстань між стартовими позиціями відрізків у шифротексті;
3) візьмемо один із спільних дільників цих відстаней в якості довжини ключа.

Для уточнення довжини ключа будемо використовувати метод Фрідмана, що був винайдений американським криптологом *Вільямом Фрідманом* у 1920 році. Цей метод базується на обчислені індексу збігу (ІЗ), який дозволяє визначити для деякої послідовності $x = (x_1 x_2 \dots x_n)$ з літер алфавіту $A = \{a_1, a_2, \dots, a_m\}$ ймовірність того, що два випадкових елемента цієї послідовності збігаються. Значення ІЗ обчислюються за формулою:

$$I_c(x) = \frac{\sum_{i=0}^{m-1} n_i(n_i-1)}{n(n-1)}, \quad (2.1)$$

де n_i – кількість появи літери *i* в послідовності *x, n* – загальна кількість літер в x.

Довжину ключа можна визначити за формулою:

$$l \approx \frac{k_p-k_r}{I_c(x)-k_r+\frac{k_p-I_c(x)}{n}}, \quad (2.2)$$

де $k_r = \frac{1}{m}, k_p = \sum_{i=0}^{m-1} p_i^2$, де p_i – частота появи літери *i* в природній мові.

Відомо, що ІЗ рядків осмисленого тексту для різних природніх мов такий:

$I_c(x) = 0{,}058$ – українська мова;

$I_c(x) = 0{,}065$ – англійська мова

Нехай криптограма $c = (c_1 c_2 \dots c_n)$, отримана за допомогою шифру Віженера з ключем рівним *l*. Запишемо її літери в *l* стовпців.

Таблиця. 2.3. Запис шифротексту за довжиною ключа

C_1	C_2	…	C_l
c_1	c_2	…	c_l
c_{l+1}	c_{l+2}	…	c_{2l}
c_{2l+1}	c_{2l+2}	…	c_{3l}
…	…	…	…

Якщо довжину ключа визначено правильно, то кожний стовпець C_i – це відрізок відкритого тексту, зашифрованого простою заміною. Тоді ІЗ кожного стовпця буде близьким до ІЗ осмислених текстів цією мовою. Наприклад, для осмислених текстів англійською мовою ІЗ лежатиме в межах $0{,}038 < I_c(x) < 0{,}065$. Якщо довжину ключа визначено неправильно, то стовпці C_i будуть випадковими, а ІЗ таких стовпців буде близьким до 0,038.

Для текстів англійською мовою довжину ключа можна визначити за таблицею 2.4.

Таблиця. 2.4. Визначення довжини ключа за значенням ІЗ

l	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	∞
$I_c(x)$	0,0667	0,0525	0,0478	0,0445	0,0441	0,0431	0,0424	0,0414	0,0410	0,0407	0,0384

Припустимо, що на першому етапі ми знайшли довжину ключа l. Тепер для кожного стовпчика C_i визначимо літери, що найчастіше повторюються та за допомогою частотного аналізу знайдемо літери ключа.

Приклад 2.3:

Дано текст, зашифрований шифром Віженера:

MRGFNIATXZQVFFNUXFFYBTCETYXIIXGZKACJLRGKQYEIXOYYAUAPX
YIJLHPRGVTSFPAYNNYURZOPHXWYXLFRNUTZBRFKAHFWFZESYUWZ
MOLLBSBZBJHFPLXKHVIVMZTZHUIWAETIUEDFGLXDIEXIYJIUXPNNEI
XABVCINTVCIEZYYDAZGZIWTYXJIKTRZLMFFKALGZNVKZXIIMXUUNA
PGVXFUSMISKHVYVOCRVXRIWTYXZOIRFNUXZNXLDUDPZGVHVOWM
OYJERLAUGLVTUXTHRBUQZTYTXORNKBASFFXGHQVDSHUYJSYHDYU
WYXYYKHVTUCDACAHXSEVGJIEFZGLXRSBXSYKOEPPNYAKTUACEFYI
LFWEAHCIAUALLZNXMVCKLRRHGFNXMOYUESKPM

Потрібно визначити ключ та прочитати текст.

Використаємо спочатку метод Казіскі для знаходження довжини ключа. У шифротексті триграма TYX зустрічається 3 рази. Відстань між першою і другою появою становить 156 символів, між першою і третьою – 210. НСД (156, 210) = 6, тому можна припустити, що довжина ключового слова рівна 6.

Для підтвердження гіпотези скористаємося методом Фрідмана. Обчислимо ІЗ за формулою (2.1) для всього шифротексту $I_c(\text{с}) = 0{,}043$. Обчислимо довжину ключа за формулою (2.2): $l \approx 6{,}64$. За отриманими даними та за таблицею 2.4 можна зробити висновок, що довжина ключового слова обрана правильно і дорівнює 6.

Запишемо шифротекст у таблицю із 6 стовпчиків (табл. 2.5).

Таблиця. 2.5. Запис шифротексту за довжиною ключа 6

C_1	C_2	C_3	C_4	C_5	C_6
M	R	G	F	N	I
A	T	X	Z	Q	V
F	F	N	U	X	F
F	Y	B	T	C	E
T	Y	X	I	I	X
G	Z	K	A	C	J
L	R	G	K	Q	Y
E	I	X	O	Y	Y
A	U	A	P	X	Y
I	J	L	H	P	R
G	V	T	S	F	P
A	Y	N	N	Y	U
R	Z	O	P	H	X

C_1	C_2	C_3	C_4	C_5	C_6
W	Y	X	L	F	R
N	U	T	Z	B	R
F	K	A	H	F	W
F	Z	E	S	Y	U
W	Z	M	O	L	L
B	S	B	Z	B	J
H	F	P	L	X	K
H	V	I	V	M	Z
T	Z	H	U	I	W
A	E	T	I	U	E
D	F	G	L	X	D
I	E	X	I	Y	J
I	U	X	P	N	N
E	I	X	A	B	V
C	I	N	T	V	C
I	E	Z	Y	Y	D
A	Z	G	Z	I	W
T	Y	X	J	I	K
T	R	Z	L	M	F
F	K	A	L	G	Z
N	V	K	Z	X	I
I	M	X	U	U	N
A	P	G	V	X	F
U	S	M	I	S	K
H	V	Y	V	O	C
R	V	X	R	I	W
T	Y	X	Z	O	I
R	F	N	U	X	Z
N	X	L	D	U	D
P	Z	G	V	H	V
O	W	M	O	Y	J
E	R	L	A	U	G
L	V	T	U	X	T
H	R	B	U	Q	Z
T	Y	T	X	O	R
N	K	B	A	S	F
F	X	G	H	Q	V
D	S	H	U	Y	J
S	Y	H	D	Y	U
W	Y	X	Y	Y	K
H	V	T	U	C	D
A	C	A	H	X	S
E	V	G	J	I	E
F	Z	G	L	X	R
S	B	X	S	Y	K
O	E	P	P	N	Y
A	K	T	U	A	C
E	F	Y	I	L	F
W	E	A	H	C	I
A	U	A	L	L	Z
N	X	M	V	C	K
L	R	R	H	G	F
N	X	M	O	Y	U
E	S	K	P	M	

Підрахуємо кількість появи кожної літери алфавіту по стовпцях. Занесемо дані в таблицю 2.6 (комірки, що позначені кольором відповідають літерам, що зустрічаються найчастіше).

Таблиця. 2.6. Кількості появи літер по стовпцям шифротексту

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
C_1	9	1	1	2	6	7	2	5	5	0	0	3	1	6	2	1	0	3	2	6	1	0	4	0	0	0
C_2	0	1	1	0	5	5	0	0	3	1	4	0	1	0	0	1	0	6	4	1	4	8	1	4	9	8
C_3	6	4	0	0	1	0	9	3	1	0	3	3	5	4	1	2	0	1	0	7	0	0	0	13	2	2
C_4	4	0	0	2	0	1	0	6	5	2	1	7	0	1	4	5	0	1	3	2	9	5	0	1	2	6
C_5	1	3	5	0	0	3	2	2	6	0	0	3	3	3	3	1	4	0	2	0	4	1	0	10	11	0
C_6	0	0	3	4	3	6	1	0	4	5	6	1	0	2	0	1	0	5	1	1	4	4	4	2	4	5

Знайдемо тепер саме ключове слово. Так як кожен з стовпців таблиці є результатом зашифрування фрагменту відкритого тексту простою заміною, то спробуємо застосувати частотний аналіз, тобто виконаємо зсув відносно літери, що найчастіше зустрічається у кожному стовпці (табл. 2.7).

Таблиця. 2.7. Визначення літер ключового слова із застосуванням частотного аналізу

Стовпець шифротексту	Літера, що найчастіше зустрічається	Зсув відносно E	Можлива літера ключового слова
C_1	A	0 – 4 mod 26 = 22	W
	E	4 – 4 mod 26 = 0	A
	F	5 – 4 mod 26 = 1	B
	N	13 – 4 mod 26 = 9	J
	T	19 – 4 mod 26 = 15	P
C_2	Y	24 – 4 mod 26 = 20	U
	V	21 – 4 mod 26 = 17	R
	Z	25 – 4 mod 26 = 21	V
	R	17 – 4 mod 26 = 13	N
C_3	X	23 – 4 mod 26 = 19	T
	G	6 – 4 mod 26 = 2	C
	T	19 – 4 mod 26 = 15	P
	A	0 – 4 mod 26 = 22	W
C_4	U	20 – 4 mod 26 = 16	Q
	L	11 – 4 mod 26 = 7	H
	H	7 – 4 mod 26 = 3	D
	Z	25 – 4 mod 26 = 21	V

Стовпець шифротексту	Літера, що найчастіше зустрічається	Зсув відносно E	Можлива літера ключового слова
C_5	C	2 – 4 mod 26 = 24	Y
	I	8 – 4 mod 26 = 4	E
	X	23 – 4 mod 26 = 19	T
	Y	24 – 4 mod 26 = 20	U
	U	20 – 4 mod 26 = 16	Q
C_6	F	5 – 4 mod 26 = 1	B
	K	10 – 4 mod 26 = 6	G
	J	9 – 4 mod 26 = 5	F
	R	17 – 4 mod 26 = 13	N
	Z	25 – 4 mod 26 = 21	V
	V	21 – 4 mod 26 = 17	R

Отже, ключове слово: ARTHUR. Тепер можемо дешифрувати текст, розділяючи слова пропусками: Many traces we found of him in the bog girt island where he had hid his savage ally a huge driving wheel and a shaft half filled with rubbish showed the position of an abandoned mine beside it were the crumbling remains of the cottages of the miners driven away no doubt by the foul reek of the surrounding swamp in one of these a staple and chain with a quantity of gnawed bones showed where the animal had been confined a skeleton with a tangle of brown hair adhering to it lay among the debris.

КРИПТОСИСТЕМА ХІЛЛА

У 1929 році американський математик *Лестер Хілл* придумав новий поліграмний шифр заміни, в якому використовувалися як модульна арифметика, так і лінійна алгебра.

Ключем шифру є квадратна матриця $K(n \times n)$, елементи якої числа від 0 до 25, $\det K \neq 0$, $n \geq 2$. Літери алфавіту нумеруються в порядку їхнього зростання від 0 до 25. При шифруванні відкритий текст розбивається на блоки з n літер, числові значення яких розглядаються як вектор розмірності n. Кожен вектор множиться на матрицю шифрування $K(n \times n)$ по модулю 26 (для англійського алфавіту).

Приклад 2.4:

Повідомлення *HELP* зашифруємо за допомогою ключової матриці:

$$K = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix},\ det\, K = 15 - 6 = 9 \neq 0.$$

Розіб'ємо відкритий тест на вектори розмірністю 2, літерам поставимо у відповідність їх числові значення:

$$P_1 = \begin{pmatrix} H \\ E \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix} \qquad P_2 = \begin{pmatrix} L \\ P \end{pmatrix} = \begin{pmatrix} 11 \\ 15 \end{pmatrix}.$$

Помножимо ключову матрицю на кожен вектор відкритого тексту та отримаємо шифротекст *HIAT*:

$$K \cdot P_1 = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 33 \\ 34 \end{pmatrix} mod\ 26 = \begin{pmatrix} 7 \\ 8 \end{pmatrix} = HI;$$

$$K \cdot P_2 = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} 11 \\ 15 \end{pmatrix} = \begin{pmatrix} 78 \\ 97 \end{pmatrix} mod\ 26 = \begin{pmatrix} 0 \\ 19 \end{pmatrix} = AT.$$

Для того щоб дешифрувати повідомлення, кожен блок шифротексту з n літер множиться на обернену (за модулем 26) матрицю до матриці шифрування.

Шифротекст *HIAT* дешифруємо за допомогою матриці оберненої до ключової: $K^{-1} = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$ та отримаємо повідомлення *HELP*.

$$K^{-1} \cdot P_1 = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 8 \end{pmatrix} = \begin{pmatrix} 241 \\ 212 \end{pmatrix} mod\ 26 = \begin{pmatrix} 7 \\ 4 \end{pmatrix} = HE;$$

$$K^{-1} \cdot P_2 = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 19 \end{pmatrix} = \begin{pmatrix} 323 \\ 171 \end{pmatrix} mod\ 26 = \begin{pmatrix} 11 \\ 15 \end{pmatrix} = LP.$$

Завдання до лабораторної роботи

Завдання 1

Виконати зашифрування, дешифрування та криптоаналіз повідомлення, зашифрованого шифром Віженера згідно варіанту (визначається номером студента у журналі). Усі кроки алгоритму шифрування описати у звіті.

1. Виконати зашифрування тексту шифром Віженера на сайті *CrypTool Online* – https://www.cryptool.org/en/cto/ з використанням шаблону *Vigenère* згідно варіанту (ключове слово обрати самостійно):

Варіант №	Відкритий текст
1.	Two things are infinite: the universe and human stupidity; and I am not sure about the universe. Albert Einstein
2.	Live as if you were to die tomorrow. Learn as if you were to live forever. Mahatma Gandhi
3.	Darkness cannot drive out darkness: only light can do that. Hate cannot drive out hate: only love can do that. Martin Luther King
4.	Happiness is when what you think, what you say, and what you do are in harmony. Mahatma Gandhi
5.	Peace cannot be achieved through violence, it can only be attained through understanding. Ralph Waldo Emerson
6.	It has become appallingly obvious that our technology has exceeded our humanity. Albert Einstein
7.	Far and away the best prize that life has to offer is the chance to work hard at work worth doing. Theodore Roosevelt
8.	When you are enthusiastic about what you do, you feel this positive energy. It's very simple. Paulo Coelho
9.	It is fine to celebrate success, but it is more important to heed the lessons of failure. Bill Gates
10.	Between the great things we cannot do and the small things we will not do, the danger is that we shall do nothing. Adolph Monod
11.	Courage is what it takes to stand up and speak; courage is also what it takes to sit down and listen. Winston S. Churchill
12.	A cat has absolute emotional honesty: human beings, for one reason or another, may hide their feelings, but a cat does not. Ernest Hemingway
13.	Tell me and I forget. Teach me and I remember. Involve me and I learn. Benjamin Franklin
14.	When something is important enough, you do it even if the odds are not in your favor. Elon Musk
15.	Even if I knew that tomorrow the world would go to pieces, I would still plant my apple tree. Martin Luther King

2. Додати скріншот зашифрування до звіту. Описати алгоритм шифрування у звіті на прикладі перших 2-3 слів відкритого тексту.
3. Обмінятися шифротекстом та ключами із іншим студентом своєї групи.
4. Виконати дешифрування шифротексту одногрупника на сайті на сайті *CrypTool Online* – *https://www.cryptool.org/en/cto/* з використанням шаблону *Vigenère*.

5. Додати скріншот дешифрування до звіту. Описати алгоритм дешифрування у звіті на прикладі перших 2-3 слів шифротексту.
6. Виконати криптоаналіз шифротексту, зашифрованого шифром Віженера згідно варіанту:
    a) Обчислити довжину ключа за методом Казіскі. Здійснити пошук триграм, що повторюються можна, використовуючи шаблон *N-Gram Analysis* в розділі криптоаналізу на сайті https://www.cryptool.org/en/cto/;
    b) Обґрунтувати довжину ключа, використовуючи метод Фрідмана. Обчислити індекс збігу можна за допомогою MS Excel. Додати скріншот обчислення індексу збігу та описати хід обчислень у звіті;
    c) Знайти літери ключового слова, використовуючи частотний аналіз;
    d) Відновити початкове повідомлення із знайденим ключем та додати скріншот до звіту.

Варіант №	**Шифротекст**
1.	TJAVPQCZCYPGWPZIHHISXHKJKWSOYEPBZQFDBERKWSIIMTFUFFSMDEJPZEXKASXMKDTIEE LCIEEPBZEEXAWPZCHDIDSONOJPBZPZVVPWZCHDIDTJAVPDBAMJEONXFUOYSECSYXVSQKW DXQOCJISITRGHAEIIVWRUHSWAZCRWRUUWNINXHDWFBSOECIOJHUJGPWNXAIMEVWJXYTA SIYPJAELCWRIIHSSMKWWJSLGGAPMTGPLRIAEQZRGPLVJBEZVGGASLIGEHVCCJIFUIOEITXQ WKQZWGBDFSLZISKVETJAVNGCJKRCRWPNPMOVZVVPRFDBARFDBAIOXGPWNXHDSLIDKPR GWPMVHSRIINPKHPWOOKFDRWRUQOZJFGQAWNDFGMEVKEXYIVAQRVOERJIHDIDPBZAZIVE RKWSI
2.	PVVCHWGFYLPVLCWOGFXXYVKZVADRCDPSZDEAQRFVACLCVKICDDNSTZQJSTEHZARJEAH YPBWZNLBOVRGHXSVYDJRNTOHPVXDUPVHHRSCTYARREKKIJLQZZZGHOPVQRNSKSLOCEP DJRZYHWQYZIPVVXZAJVQRQBUPDYVFEKAFRYGIOPMHAOTSWEAVHHRSSPHJTFCFARRADN HWZUPVVDDISIPDOCEDWDOKXHWBJEKWHKSLOUFZGXMVTVXCKSDCCFOEUSWZUPVVADO HKPQPVFFVWBUJHWFJLQZOGCHHIUPWKKYLWSWCWFKAV
3.	BTESSAFISTWDTGRBFHOROEAXIBTENEZPEYXBASGCBTEEEZQTNIBTITKAKOAKMFAYLIYEJS NNEIECEECSZPSJMUUNOWPFHKQEARJWATRORSFHOROETNEBEEKQMPLOQQFLKWAIHKRBT EEAMDEORGAUXLMMDZSVAMUVMFHGRTUVOROEIFIETETXPQYXIJDOAKPFOAXJGTOXAYOX IBTATXPMTOWVFIZXPQMUWBUMVSZFATXBTITKAXIKXWACRSAQTUAPQRKZMDYUYZEEIV MFHKEZFIYFCDIKHTUKKPIZDSEZWSZSIFRKEAGRKCWGRKRMYIKWEAURHTAVKXWETKETM WGC
4.	VOEZUWVTEEIMHBLGNMPNIAZLGDOERHKAHVTOKZTUKMPHBVRMVFOSZLGOUZGROPNQZS EVREKPCAENRPKASPURVLNGYAGSIIKSPHPYGGKKIFREPKOSOKPVRNTGGPNGNIOPDFZSHILN IOULAFUJKUFVTMVFAAJMVDAFTSVTENTXVOAGCIUOOHRHXVYNMIHHRGNIUJIRTGGZENILU ARNORKUGVTMVZOJTHKYEPZMQUHNBIJPTUKVVVHNXQGKUFRMVALRHYVZOZKHCFTUKTK LCVTKVVGRZLGYOSJMUZOPOEVLDXTSYSEQMIYPLYUTGUUCYYEOTRXVKMYVTKXPSGGWQ MRRGPKAYNTHQMOHXJTPGUZJWSPBYMVPOAZLGYEVTXJHTJKWJHLYKMVOEEMSOHDSXSO AHRXIXLLNZMQUOELPGLFEUQVOEYOKJAIAZSVOECKEELAAJWCMEGESHHNRCHCYKNMI

Варіант №	Шифротекст
5.	WEVGCDYEXIGLECVSXHGADRDGVUYLUMAINEIYYSVFTJOTIILEFELROTXWIQWMTXAPHREL ONJWTCWPWTQVBINFWDVPGVWEPUCDYKPSLXHGWXPEPGTAHGRNSUJEKINVXWIAPWLIRV SPUUGWIMOPCDYVGFTINCWZIDQVILEJYHLOHERSUPXGCRQESETPMVLTQVILEGBEICVECXP CYHIOHEGSOOJJPLQJEIORPTAHGRHSMGSCIIUNJWTCFDYTVSHTECODVMQWIFECYIMFWPDJA NPILEOSBINVEUXETXWIDQSGGLQWTWAPHNSUTIPPOPIXRTJILLONIWSUUITECJSCIIUHXJFGV TRTASJONQAPRDCPAZETCQIAWXXJUNMUCOWPXWTGRRERGJJPLA
6.	ZJYGIGYMNCCACCOXMBLVSWZSZJYGISFVGWHGYIRAUUKWAFRJVPLTSVEKOXCGYIRKCENS RKSFKMLDWSPSUWLCCSILCGYIRKIYKMMSHRCCYGIJZPLDCFVCOMKVCPHSJRESFJWREHSOV BNMLDGUFACGBPVVNXCEPSUJKRCPBWWAXQOJSBMIRATNKVAYSQPFEXOYCWYGIVEWTW OQFJFWSYZRGUCATIRFSQFXHWFFDMGZHRMINKSPIITDMEVNOAQRSCCGBGIESLWSRRIFRVM MDMOYZWCGACRWSACARXESBQRPTJIVJXIUOAULAKHUVMNLSEVWTKCSFXHWFFZRMABQ KLEFWEIISHSPKMVWCSNLELVRIXHSHCVVSGBXESWKOYFXOXDRFTLWKUVVENSEKLALDRI WOFABMISZSBIWHWKVCPIEARUMALSYPQACSSIMEFRFRRDOVREXHSHCVVSGBSRGEKOGIEG WRLKLEJSJZPLTSCCINLMBWTEGDYVAHGKVCPCGARKSHWZC
7.	ZMWSWZPPWPSMZWOXBPAZLKSIETDRKMXMVRTSNIKVOEKNLJFQAVHLZXKEDDMPKEFBCG NPWDRMJLUOCEAGGXFEHMWSMQUMKDBPSPWQOSWQFWVQUWHHBFDMKXBIETDUWEAPW DFUPAKDTQDMJHWFOESUATAIJWVQZIAVWQOOMDFPUWMIFAIMNHFKDIJPVQWZQRIDZZWD AEWZWVKQABSQRFKUGUFASJJLBSOBZHAFNCWKSMNQKWVQLTSFSIDMJHWXKDWBCGZMW SWZPPWPSMZWOKWPZMFIODWESBOOHWUNCRHMSYSEWUGRBNAIEUOKBWJJSFUWMUKAA ASQRXABQRIDPZGXPXAADDMMJLOKSZWOSLBUPAERFZEVYWVQUTDZOEDIODMTAIJLHEOI XHVQWZKLHESIJPVQWZLKSPWQKHGSQIJGMAQNJRAQRMJBVMNUZHODUWMURDAIEVODAA OHSFWVVWCYKZJRKNNQFJHTAULUIQDMSUWEPPWSZMYMOKSDAQDRJQUWM
8.	FPKCEKUAWWFQRETLCCGHTNPINRPCKLJAKGHYVCDLRMASTHUWHDWYEJEOMBZRTOYJZR XPXUTLOLHMJSZSTHUHBGPIFHQAIJCMAGPCKLJHBNVWGAKMKLRDTLBVVVXLJANMIUEHSA KIWJUIEFJEYRXUUCGHXJCSJIGANEFNXUIPLFBHZWUEWUEHGHXVCSBDRACSEYHZKLRSNHJI GCHFFVRDMBVJEELBTYGFNLISJSHZIIQGXIIKVAVFRCGOXPVREEWXVVUUXMKLRMHCJXUU GAICRAKNYANIMCEKHPMOIRRDYIIXUEVIKXBNLYVHFSAINIQPBHBMFHHHKLRSTHUCGOIMF JSUKLFAFVXLDMYIHHRRQSVUIPRTTHUQNRHIEAUEKYJLNDHQJPNYTFFRTTAYJMQELIWXUE MLVRPHXM
9.	MFWVVOLDIYRTZYTOLPKRIIGPTJEYEYWKHZGWHLDNHCBWTZHLPRYNSEARTOLPLBYREDL TAOXPKVDJRNXDOMIHBKHOLPBECMIOBSLZPLFGSJJDMRRNXSXETCIMKVEUIOBVDOSZTEDO LPKVWVWYHEODWPLRVZXSXURDTLGUTDGVEVOAALMVROLLMIAISFMFFXPPYKSVROLGIG PPWUORRWXRFWCWXRFOSEAVBMSHGVAMXSHWTCITLCAIHEAVADVHTJCJSWFFINXLGUCGI LKRNYTCXJEIXWRVVZREAVSJYYWFFOLPPRTZVHTJSOMWEKHZFPTJTGEJALDYPPWFNOLPIR LZFPTTHVROMYENXLBESNTCXRDDRNASYDRNA
10.	WHMCCZGULRTQQYKGZBSIXMKPSXFPPISDBSLSEIDXWQTAPSFMTRZUASISPJFXARGZYDUM OSOZVZASZLELWZIYHYQBDXAXTLWQACNEFUBHMDXAWMVQWAIFXQVIWXMGELAZOSGD AWILFQYKAZIWPLTMDIUDMEXZACRLLEBSELTIGITQMYWLMVOMFSICSMZLDSHMBTIFFTJTM EPTRYFPPFMFBZRXAZLVAPMESLTMESHFPPVWMTEVGGJWIKUVWMXQPLTHQVHLWZBSSKQP THVQVOSGDADXSKKWSKQLQSJFWZPGZO
11.	NJYRVNIWHULTCHBLDCGKMFUVXNZLXHECNALBEZHYFRNBVGPINUXVXYVKDUDPGXUWL VZMBVGSOMKHENMYRKITUTCSSLMYYBUYFLFHMZIGDBKBRVNIWHULTCHBLDCGKBEMML TUAHVGYIEPWRSPOBCYRVNIOGJHEMVPHLMFPGUXBNXJNLPMNBTAXMYKFHLLBUMLCMPH ENALGKYESLPINPLRFFVLKWXYMRCGSRXIBUZKIULMYYULLKWAVBTY
12.	PRAGPGOIHJYFLWKZCFASMZQBYMZDRTVFJZGGZENYJRLPIIKFOEDWIRUARDXGLRBPXGOIS TWGLQDZQNUHEYWBTYCCZVNMLVRPLENYQBYELNXELRGOLGOETHEAFWUXGHTFEYXBAL ENTRSPOAEAPQABMAHVYMINSMTTSALMNQIAAIDWCGOIMNIYCISRLVJLWVWYLWSKVNJXI XEYMSROLRTFUOQBYICJQSVVTDRT
13.	KAYQWWNSGVOMZBFMOJDOZMZDYHQUZQYIYXZPYAZBTLKLGMYAJSUVZSWLUDPQBERWC KTUDNTPXTDMWYYIAVDFNPEETRMOGZAYWEZBDYSDIMOCUEZLZLYHQUEMURABPAYUEIY BURADTBJFAZZSWNQMOQTUDNLKNLUMDENVQCDSSCAVOGYIAVLJQOHMLLISGXAMWTMV OKFKQCDPJAXQKCNTEBCSJIYXZPYAZKPMZRBICCSTEICCTUDNTPXTFMLAMEDARSNDQALLI RATPKTDQTDDFMUTTCXSFMPPTUDTTTJSQDPLBHQVHCFRQIOSQTEIDYQLACCTFLGMDYSDN MWGJFEICCROEBWWNNRTFCSCQLMWTUDXLPJNFADGGLUVRQFNPKWMXEDMWYYIHMDUM

Варіант №	Шифротекст
	EZMGCWWQMYATUZBPPFDQDLQYAFQYEXEFJLAPFQMWBTWZCYQFFQWCGSSQKFPJWQBFPS TAWFPKAYQWGJSRWCCROFQZLFLMVOKJNFIWQZPBWCR
14.	LOEUSWJLTZCXSZIFAKLPVDHEFNFRFYWZFHONSUDOCRWDHHBSWHRHOBJHIGKAHBLOPWU RFUCIDPFHKDWUWHONWPNOCRWDERDAFAODZHLOAWZEXLHDGPGVFISNOPTKDWKZIRBAP JIWSIWUTDBZSJCHDPSUCHKAFLEGHKDLAUBPGSOYSKMYSHZRWZFLFOLPNDZHGBRJZKJFAQ RKMYIPDAJMEFHEGUSLTSWJAQBKLSOYSKMYSHZRWZWHQWFUOWTQDSYRDAFAORINSIIO WPQAOOCRWVTKSNKVRRINHVTHBPAHLWCYJLAWSANVLXHEGUAQRWDSHRDAKMOUOXW ATHFSGYLGFAKAIQHDWMEDFHWZSQSOKHNGCLWUHHONLLDYWOAVNRTLWVPOSSZVEPPN SJEOWBW
15.	TFECVPZEKULXWVYPEREOJTRRPEMCVJSITYRCMTDRLRSKBWIAIAESDRANIBPCCECKVNMNX QLRCZARSRKBWIAIAESLZIPFYGELGTZPTRGCVGMNXGSIPIVYGIGYPWAIRELEJNXIIERLGHZG TWTYRAIRWBCQAEPPSFRQPHITNEIDGEPGIJRDITFSLGTJCSCSZPLPOIVYXECYPGTLNWJRFZHL ITUNSMVFDLAGRZJATUTIVVZPRTRFPRSVBQSNVFMIIETLWAKVDJATGTSNFSDTIIVESNVOPGO DRDMNJBXIAIRLENRGSPEKRZJGFQAVATGTGEDRLRSKBAIRWBCQOMRCENUBGIRRTLMNZAE LEWNNIOWNWPOSFEECCRDWODRLGTFSGMSZBYSFWNTXHFSOISZEPTRRPEMCVVDEMVNYW OWVYZIKVYKTYRAIRWRNXIFAOISZEPH

Завдання 2

Виконати зашифрування повідомлення згідно варіанту (визначається номером студента у журналі: непарний – 1 варіант, парний – 2 варіант). Усі кроки алгоритму шифрування описати у звіті. Обчислення можна виконувати в MS Excel.

1. У криптосистемі Хілла з матрицею $\begin{pmatrix} 11 & 2 & 19 \\ 5 & 23 & 25 \\ 20 & 7 & 1 \end{pmatrix}$ зашифруйте текст LOT TRY CAT.
2. У криптосистемі Хілла з матрицею $\begin{pmatrix} 11 & 14 & 19 \\ 19 & 17 & 24 \\ 2 & 0 & 18 \end{pmatrix}$ зашифруйте текст OUT OF DATE.

Контрольні запитання:

1. Поясніть відмінність між шифрами моноалфавітної та поліалфавітної заміни.
2. Опишіть алгоритм шифрування Віженера.
3. У чому полягає метод Казіскі?
4. Як уточнити довжину ключа методом Фрідмана?
5. Що таке індекс збігу?
6. Що являє собою ключ у криптосистемі Хілла?
7. Опишіть алгоритм шифрування криптосистемою Хілла.

ТЕМА № 3. МОДЕЛЮВАННЯ ПРОЦЕСІВ ШИФРУВАННЯ ЗА ДОПОМОГОЮ ОПЕРАЦІЇ XOR. АЛГОРИТМ DES

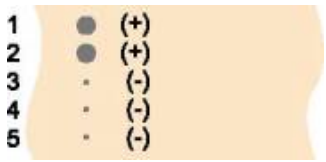
Мета роботи: набути вміння шифрування повідомлень із використанням операції побітового додавання за модулем 2, дослідити процеси шифрування за допомогою алгоритму DES на основі навчальної програми CrypTool 2.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням MS Excel та CrypTool 2, текстові повідомлення згідно варіанту.

Теоретичні відомості

ШИФР ОДНОРАЗОВОГО БЛОКНОТУ (ШИФР ВЕРНАМА)

Шифр одноразового блокноту, або шифр Вернама, було запропоновано у 1917 році співробітниками телеграфної компанії AT&T *Мейджором Джозефом Моборном* та *Гільбертом Вернамом*. Відкритий текст представлявся у вигляді п’ятизначних імпульсних комбінацій – коді Бодо. Наприклад, літера «А» на паперовій стрічці мала вигляд (рис. 3.1):



«+» – отвір;
«-» – його відсутність.

Рис. 3.1. Літера «А» на паперовій стрічці

У класичному розумінні одноразовий блокнот є унікальною послідовністю символів ключа, що згенерована випадковим чином. Заздалегідь готувалася «гама» – перфострічка з випадковими знаками. Потім електромеханічно складалися її імпульси з імпульсами знаків відкритого тексту. Отримана сума представляла собою шифротекст. На приймальному кінці імпульси, отримані по каналу зв'язку, складалися з імпульсами тієї ж самої «гами», в результаті чого відновлювалися вихідні імпульси повідомлення.

Ідея шифру Вернама легко поширюється на двійкові дані. Ключем виступає послідовність випадкових символів. При цьому ключ повинен володіти трьома критично важливими властивостями:

1) бути дійсно випадковим;

2) за розміром збігатися з заданим відкритим текстом (ключ ні в якому разі не зациклюється);

3) застосовуватися тільки один раз.

Для шифрування бінарних даних (потоку бітів) виконується додавання бітів за модулем 2 (операція XOR, eXclusive OR – виключне або), що позначається ⊕ (табл. 3.1).

Таблиця. 3.1. Операція XOR над бітами

⊕	**0**	**1**
0	0	1
1	1	0

На практиці використовують довгі випадкові або псевдовипадкові ключі, згенеровані за допомогою спеціальних технічних пристроїв або програмно-апаратних комплексів. Можна один раз фізично передати носій інформації з довгим дійсно випадковим ключем, а потім по мірі необхідності пересилати повідомлення. При дешифруванні одержувач, використовуючи точно такий самий ключ, виконує додавання за модулем 2 кожного символу ключа та шифротексту.

Приклад 3.1:

Шифрування за допомогою шифру одноразового блокнота повідомлення *SUN* із використанням випадкової ключової послідовності 00001011 00010010 00001111:

Відкритий текст	01010011 01010101 01001110
Ключова гама	00001011 00010010 00001111
Результат додавання за модулем 2	01011000 01000111 01000001
Шифротекст	*XGA*

У 1949 році Клод Шеннон опублікував роботу, в якій довів абсолютну стійкість шифру Вернама. Інших шифрів з цією властивістю не існує. При цьому умови, яким повинен задовольняти ключ, настільки сильні, що практичне використання шифру Вернама є важко здійсненним. Тому він використовується тільки для передачі повідомлень найвищої секретності.

АЛГОРИТМ DES

Американський стандарт шифрування даних (Data Encryption Standard), оснований на мережі Фейстеля та прийнятий у 1977 році, є типовим представником сімейства блокових шифрів.

Ключ шифрування складається з 56 випадкових бітів; додається ще 8 біт в позиціях 8, 16,…, 64, таким чином, щоб кожен байт містив непарну кількість одиниць. (використовується при знаходженні помилок при обміні та зберіганні ключів).

Процес шифрування полягає в початковій перестановці 64 бітів вхідного блоку, шістнадцяти циклах шифрування та кінцевій перестановці бітів (рис. 3.2).

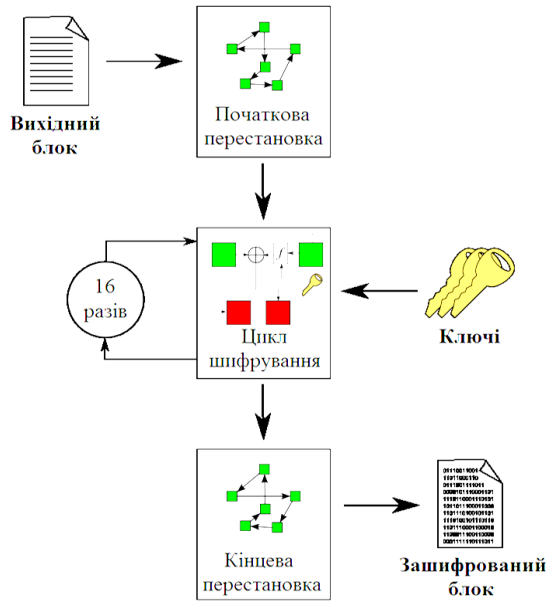


Рис. 3.2. Загальна схема алгоритму DES

Розглянемо алгоритм докладніше:

Початкова перестановка

Початковий текст, що являє собою 64-бітний блок $X = (x_1, x_2, x_3, \ldots, x_{64})$ перетворюється в 64-бітний блок $X_0 = IP(X)$ за допомогою початкової перестановки *IP* (Initial Permutation), що визначається таблицею 3.2.

Таблиця 3.2. Матриця початкової перестановки IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Раунди шифрування

Після *IP*-перестановки 16 разів повторюється процедура шифрування блоку X_0 за допомогою функції f та раундових ключів K_i, де $i = 1, 2, \ldots, 16$ (рис. 3.3).

Кожен раунд шифрування містить такі етапи:

1. $X_0 = IP(X)$ розбивається на дві половини L_0, R_0, *де* L_0 – перші (старші) 32 біти блоку X_0, а R_0 – останні (молодші) 32 біти блоку X_0.
2. Права половина R_i – це бітове додавання L_{i-1} та $f(R_{i-1}, K_i)$ по модулю 2: $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$.
3. Ліва половина L_i дорівнює правій половині попереднього блоку R_{i-1} без змін: $L_i = R_{i-1}$.

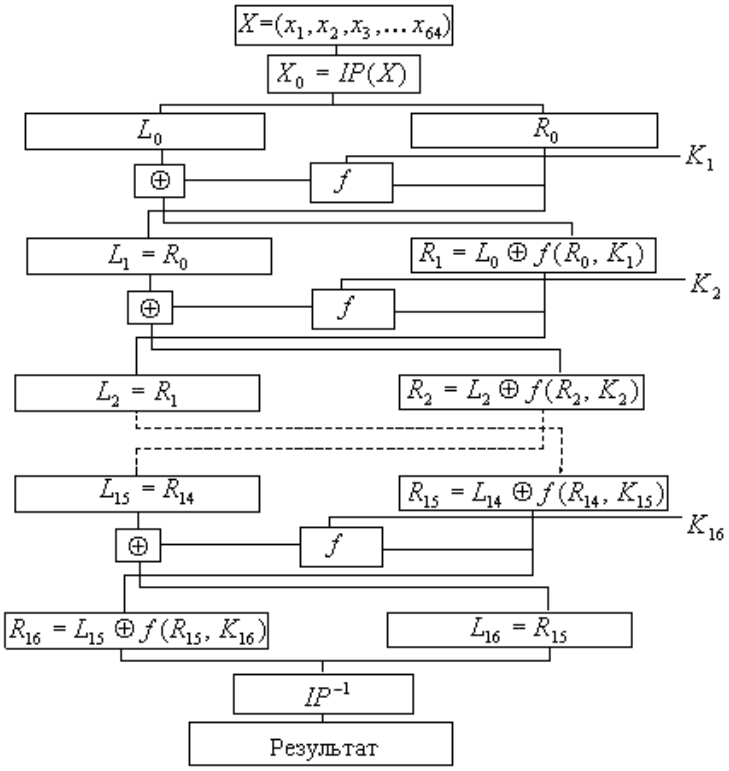


Рис. 3.3. Схема шифрування алгоритму DES

Після 16-ї ітерації ліва і права половини блока не міняються місцями.

Основна функція шифрування (функція Фейстеля)

Аргументи функції f – 32-бітовий вектор R_{i-1} та 48-бітовий підключ K_i. Для обчислення функції f використовуються:

1) функція розширення E;
2) перетворення S, яке складається з 8 перетворень *S*-блоків;
3) перестановка P.

Функція E розширює 32-бітовий вектор R_{i-1} до 48-бітового вектора $E(R_{i-1})$ шляхом дублювання деяких бітів R_{i-1}. Порядок бітів вектора $E(R_{i-1})$ зазначений у таблиці 3.3.

Таблиця 3.3. Перестановка з розширенням

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Отриманий після розширення блок $E(R_{i-1})$ додається по модулю 2 із раундовими ключами K_i. Потім представляється у вигляді восьми послідовних блоків $B_1, B_2, \ldots, B_8$, тобто $E(R_{i-1}) \oplus K_i = B_1 B_2 \ldots B_8$.

Кожен B_j являється 6-бітовим блоком. Далі кожен з блоків B_j перетворюється у 4-бітовий блок B'_j за допомогою перетворень S_j. Перетворення S_j визначаються таблицею 3.4. Індекс j вказує, який з масивів *S*-боксу використовувати. Застосувавши операцію вибору до кожного із блоків B_j, одержимо 32-бітний блок $B'_1, B'_2, \ldots, B'_8$.

Таблиця 3.4. S-бокси алгоритму DES

		Номер стовпця															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Номер	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_1

	Номер стовпця																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	**S_2**
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	**S_3**
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	**S_4**
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	**S_5**
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	**S_6**
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	**S_7**
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	**S_8**
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Приклад 3.2:

Припустимо, що $B_3 = 101111$. Знайдемо B'_3 – ?

Перший і останній розряди B_3 – двійковий запис числа a, $0 \le a \le 3$.

Середні чотири розряди B_3 – двійковий запис числа b, $0 \le b \le 15$.

Пара чисел (a,b) визначає число, що знаходиться в перетині рядка a та стовпця b. Двійкове представлення цього числа дає B'_3.

У нашому випадку $a = 11_2 = 3$, $b = 0111_2 = 7$, а число обумовлене парою (3,7), дорівнює 7. Його двійкове представлення $B'_3 = 0111$.

Отриманий блок $B'_1, B'_2, \ldots, B'_8$ перетворюється за допомогою матриці перестановки P (табл. 3.5).

Таблиця 3.5 Матриця перестановки P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Таким чином, $f(R_{i-1}, K_i) = P(B'_1, B'_2, \ldots, B'_8)$.

Генерація ключів

Ключ K – 64-бітний блок з вісьмома бітами контролю парності, що розміщені в позиціях 8, 16, 24, 32, 40, 48, 56, 64. Ще раз відзначимо, що на кожній ітерації використовується нове значення ключа $K_1, K_2, \ldots, K_{16}$, яке обчислюється із початкового значення ключа K.

Для видалення контрольних бітів і підготовки ключа до роботи використовується перестановка ключа (табл. 3.6).

Таблиця 3.6. Матриця перестановки ключа

57	49	41	33	25	17	9	C_0
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	D_0
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

Ця перестановка визначається двома блоками C_0 та D_0 по 28 біт кожний. Тобто 56-бітовий ключ ділиться на 2 половини, які потім циклічно зсуваються на один чи два біти ліворуч в залежності від етапу.

Тобто C_i, D_i, де i = 1,2,3,...,16 визначаються з C_{i-1}, D_{i-1}, одним або двома лівими циклічними зсувами згідно таблиці. 3.7.

Таблиця 3.7. Матриця зсуву для обчислення ключів

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Число зсуву	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Після зсуву C_iD_i знову вибирається 48 бітів з 56 бітів та міняється їх порядок за наступною таблицею (табл. 3.8):

Таблиця 3.8. Матриця перестановки зі стисненням

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Наприкінці шифрування виконується відновлення позицій бітів за допомогою матриці перестановок IP^{-1} (табл. 3.9).

Таблиця 3.9. Матриця кінцевої перестановки IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

При дешифруванні даних всі дії відбуваються в зворотному порядку. Ключі застосовуються в зворотному порядку. Функція *f*, перестановки *IP* і IP^{-1} такі самі як і в процесі шифрування.

ЗНАЙОМСТВО ІЗ СЕРЕДОВИЩЕМ CRYPTOOL 2

CrypTool 2 – безкоштовне програмне забезпечення з відкритим вихідним кодом, що реалізує концепцію візуального програмування та виконання каскадів криптографічних процедур. CrypTool 2 є однією із складових великого проекту CrypTool, призначеного в першу чергу для електронного навчання криптографії та криптоаналізу.

Програмний засіб CrypTool 2 (рис. 3.4) на даний час доступний німецькою та англійською мовами, має інтуїтивно зрозумілий сучасний графічний інтерфейс та зручне меню, за допомогою якого користувач у робочій області програми може перетворювати повідомлення з використанням найвідоміших криптографічних алгоритмів.

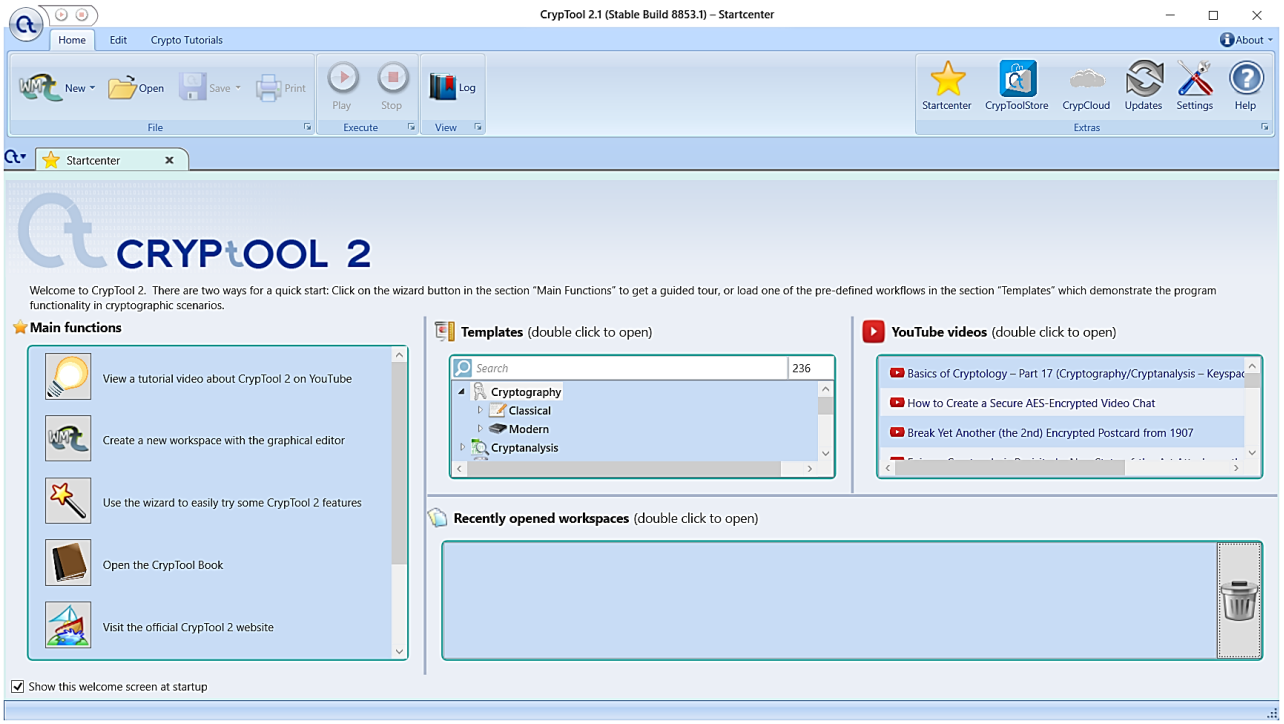


Рис. 3.4. Вікно завантаження CrypTool 2

Розділ «Templates» (рис. 3.5) містить як готові шаблони проектів, що реалізують криптографічні та криптоаналітичні алгоритми, математичні та інші функції, протоколи тощо, так і набір інструментів, котрі можуть бути використані для модифікації готових або створення нових проектів. Крім того CrypTool 2 пропонує стеганографічні способи перетворення даних, тобто такі, при яких повідомлення не шифрується, а приховується сам факт його передачі чи існування.

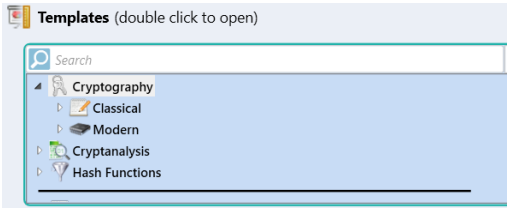


Рис. 3.5. Шаблони CrypTool 2

Для використання готового шаблону у розділі «Templates» із меню, необхідно обрати потрібний алгоритм. Після чого відкриється нова вкладка, що складатиметься із окремих модульних компонентів, пов'язаних між собою (рис. 3.6). Вони мають властивості подібні до діалогового вікна операційної системи Windows. Активізація компоненту відбувається шляхом натискання по ньому лівою клавішею миші. Кожен компонент у лівому верхньому кутку містить меню (наприклад, ), що дає змогу налаштування дій, відкриття довідки тощо.

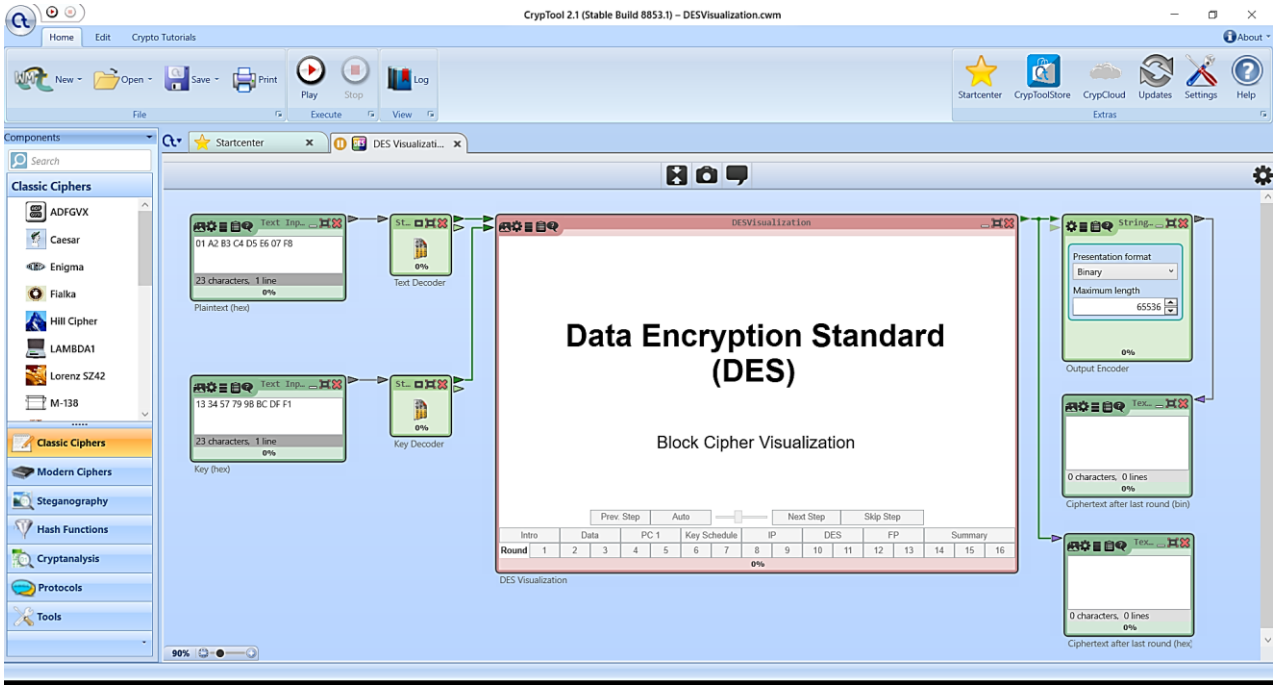


Рис. 3.6. Вкладка шаблону, що візуалізує шифр DES

Налаштування параметрів роботи компонентів (наприклад, визначення дії криптографічного перетворення, алфавіт, ключ тощо) відбувається з використанням панелі «Parameter», що знаходиться праворуч робочого вікна, або за допомогою опції «Settings» у меню компоненту. Панель модульних компонентів «Components», що розміщена ліворуч робочого вікна, дає змогу додавати до проекту нові складові, таким чином удосконалюючи його.

Завдання до лабораторної роботи

Завдання 1

Реалізувати в середовищі MS Excel або на будь-якій мові програмування перетворення текстового повідомлення у двійкову послідовність, що додається за модулем 2 (XOR) із ключовою гамою. Перевірити роботу програми для вхідних даних згідно варіанту (англійській алфавіт). Кроки алгоритму шифрування зі скріншотами описати у звіті.

Варіант №	Відкритий текст	Ключова гама
1.	JOB	00000010 01111110 00010110
2.	LAW	00000001 01100101 01111000
3.	CAT	00010011 01101001 00011100
4.	AGE	01100000 00000010 00001010
5.	TEA	00011111 00011100 01110101
6.	SKY	01111000 01100100 00011101

Варіант №	Відкритий текст	Ключова гама
7.	WIN	01100000 01101000 00011101
8.	ICE	01111111 00001011 01110010
9.	DAY	01100010 00011000 00100010
10.	ART	00001001 00010110 00011110
11.	JOY	00000011 01100100 01100011
12.	SEA	01111001 00011100 01101010
13.	BAG	01101110 01101110 00011100
14.	JAM	00011101 01100100 00011011
15.	OWL	01111000 00100001 00110010

Завдання 2

Виконати зашифрування блоку повідомлення за допомогою алгоритму DES на основі навчальної програми CrypTool 2 (Templates⇒Cryptography⇒ Modern⇒Symmetric⇒DES Visualization). Ключ ***обрати самостійно*** *(64 бітова послідовність символів), не використовувати ключ за замовчуванням. У звіті описати зі скріншотами нижчезазначені кроки алгоритму.*

Варіант №	Відкритий текст
1.	EVERYONE
2.	TOMORROW
3.	DOWNLOAD
4.	KINDNESS
5.	BIRTHDAY
6.	INFINITY
7.	ORIGINAL
8.	POSITIVE
9.	DAUGHTER
10.	GRATEFUL
11.	PROPERTY
12.	EXCHANGE
13.	CHAMPION
14.	PROGRESS
15.	MAGAZINE

Генерація ключів

- ✓ Ключ (64 біти) у 16-ій системі числення.
- ✓ Ключ (64 біти) у 2-ій системі числення (з позначенням бітів контролю парності).
- ✓ Ключ (56 біт) у 2-ій системі числення.
- ✓ Перестановка початкового 56-бітного ключа.

- ✓ Поділ ключа на дві частини C_0 та D_0.
- ✓ Зсув C_0 та D_0.
- ✓ Перестановка зі стисненням.
- ✓ Таблиця усіх раундових ключів.

Зашифрування блоку

- ✓ Початковий блок повідомлення у вигляді тексту.
- ✓ Початковий блок повідомлення (64 біти) у 16-ій системі числення.
- ✓ Початковий блок повідомлення (64 біти) у 2-ій системі числення.
- ✓ Початкова перестановка.
- ✓ Поділ блоку на дві половини L_0 та R_0 (1 раунд).
- ✓ Функція Фейстеля та додавання з ключем:
  - Перестановка з розширенням R_0;
  - Додавання R_0 з раундовим ключем K_1;
  - Перетворення з використанням S-боксів (з поясненням);
  - Перестановка P.
- ✓ Додавання L_0 та $f(R_0, K_1)$.
- ✓ Таблиця лівих та правих половин кожного раунду.
- ✓ Кінцева перестановка.
- ✓ Результат шифрування блоку.

Контрольні запитання:

1. У чому полягає алгоритм одноразового блокноту?
2. Що являє собою операція XOR?
3. Які переваги і недоліки шифрування методом одноразового блокноту?
4. До яких шифрів належить стандарт шифрування даних DES?
5. Якою повинна бути довжина ключа у шифрі DES?
6. З яких кроків складається алгоритм шифрування DES.
7. Скільки разів виконується перетворення Фейстеля над блоком у DES?

ТЕМА № 4. ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ БЛОКОВОГО СИМЕТРИЧНОГО ШИФРУ AES

Мета роботи: дослідити процеси шифрування за допомогою алгоритму AES на основі навчальної програми CrypTool 2, розглянути та порівняти результати зашифрування даних у режимах ECB та CBC.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням CrypTool 2.

Теоретичні відомості

УДОСКОНАЛЕНИЙ СТАНДАРТ ШИФРУВАННЯ AES

У 1997 році Американський інститут стандартизації NIST (National Institute of Standards & Technology) оголосив конкурс на новий стандарт симетричного криптоалгоритму.

Згідно з вимогами конкурсу, алгоритм мав обов'язково:

- ✓ бути симетричним;
- ✓ бути блокових шифром;
- ✓ мати довжину блока 128 біт і підтримувати три довжини ключа: 128, 192 і 256 біт.

2 жовтня 2000 року NIST оголосив переможця. Ним став бельгійський алгоритм RIJNDAEL. У 2001 році алгоритм був затверджений як стандарт шифрування та отримав назву AES – Advanced Encryption Standard (удосконалений стандарт шифрування).

Математична база

Скінченне поле GF(2^8) складається з многочленів вигляду

$$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0, \text{ де } a_i \in \{0,1\}.$$

У вигляді многочлена $a(x)$ скінченого поля GF(2^8) можна подати будь-який байт, що складається з бітів $a_7a_6a_5a_4a_3a_2a_1a_0$.

Приклад 4.1:

Байт: 01011010.

Многочлен:

$$0 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 = x^6 + x^4 + x^3 + x.$$

Операції над елементами скінченого поля GF(2^8) вводяться наступним чином.

Додавання

$$\forall a(x), b(x) \in GF(2^8)$$

$$a(x) + b(x) = c(x) = c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0,$$

де $c_i = a_i \oplus b_i, i = 0,1, \dots ,7.$

Приклад 4.2: У двійковій формі:

$$\begin{array}{r} 10110001 \\ 10001111 \\ \hline 00111110. \end{array}$$

Те саме у вигляді многочленів:

$$(x^7 + x^5 + x^4 + 1) + (x^7 + x^3 + x^2 + x + 1) = x^5 + x^4 + x^3 + x^2 + x.$$

Множення

Щоб задати множення у полі GF(2^8), потрібно спочатку зафіксувати нерозкладний многочлен степеня 8 з коефіцієнтами із множини {0,1} (нерозкладність означає, що він ділиться лише на себе і на одиницю). Таких многочленів є декілька, автори AES вибрали такий:

$$m(x) = x^8 + x^4 + x^3 + x + 1 = 11B_{16}$$

Два елементи поля GF(2^8) множать за модулем $m(x)$ так:

1) Множать як звичайні многочлени.
2) Проміжний результат ділять на $m(x)$ і за остаточний результат приймають остачу від ділення.

Приклад 4.3:

1) $(x^6 + x^5 + x^4 + x^2) \cdot (x^7 + x^5 + x^4 + x) = x^{13} + x^{11} + x^{10} + x^7 + x^{12} + x^{10} + x^9 +$

$$+x^6 + x^{11} + x^9 + x^8 + x^5 + x^9 + x^7 + x^6 + x^3$$

$$= x^{13} + x^{12} + x^9 + x^8 + x^5 + x^3.$$

2)

$$\begin{array}{l|l} x^{13}+x^{12}+x^9+x^8+x^5+x^3 & x^8+x^4+x^3+x+1 \\ \hline x^{13}+x^9+x^8+x^6+x^5 & x^5+x^4+1 \\ \hline x^{12}+x^6+x^3 & \\ \hline x^{12}+x^8+x^7+x^5+x^4 & \\ \hline x^8+x^7+x^6+x^5+x^4+x^3 & \\ \hline x^8+x^4+x^3+x+1 & \\ \hline x^7+x^6+x^5+x+1 & \end{array}.$$

Звідси

$$(x^{13}+x^{12}+x^9+x^8+x^5+x^3)\ mod\ (x^8+x^4+x^3+x+1) = x^7+x^6+x^5+x+1.$$

Алгоритм AES

AES є симетричним ітеративним блоковим алгоритмом шифрування зі 128 довжиною блока та зі змінною довжиною ключа. Довжина ключа може дорівнювати 128, 192 або 256 бітів. На відміну від DES, алгоритм AES не використовує збалансовану мережу Фейстеля. AES базується на архітектурі SQUARE (КВАДРАТ), для якої характерно:

1) представлення блоку у вигляді масиву байтів;
2) шифрування за один раунд всього блоку даних;
3) виконання криптографічних перетворень, як над окремими байтами масиву, так і над його рядками і стовпцями.

Блок проміжного результату називають **станом**. Матриця стану має 4 рядки та 4 стовпці (Nb).

Матриця стану при Nb=4:

$$\begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{pmatrix}.$$

Основним елементом, яким оперує алгоритм AES, є байт – послідовність 8 біт, що обробляються як єдине ціле.

Задавати значення байта зручно в шістнадцятковій системі числення. Для цього байт ділиться на дві групи з 4-х біт: група старших біт в байті представляється першим шістнадцятковим символом, а група молодших біт – другим. Наприклад, для байта 10101100 отримаємо: 10101100 = 1010 1100 = AC.

Приклад 4.4:

Розглянемо перетворення тексту у матрицю:

Відкритий текст: A SECRET MESSAGE

У шістнадцятковому вигляді: 41 20 53 45 43 52 45 54 20 4D 45 53 53 41 47 45.

Отримаємо: $\begin{pmatrix} 41 & 43 & 20 & 53 \\ 20 & 52 & 4D & 41 \\ 53 & 45 & 45 & 47 \\ 45 & 54 & 53 & 45 \end{pmatrix}$.

Ключ шифру розглядають як матрицю байтів, яка має 4 рядки і кількість стовпців (*Nk*), що дорівнює довжині ключа, поділеній на 32.

Матриця ключа шифру при *Nk*=4:

$$\begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix}.$$

Вхідні та вихідні дані розглядають як одновимірні масиви з індексами 0,...,*Nb*-1. Елементами масиву є байти. Ці блоки мають довжину 16, 24 або 32 байти.

Кількість циклів шифрування *Nr* залежить від значень *Nk*:

	Nk (Довжина ключа)	*Nb (Довжина блоку)*	*Nr (Кількість раундів)*
AES-128	4 (128)	4 (128)	10
AES-192	6 (192)		12
AES-256	8 (256)		14

Шифрування за алгоритмом AES складається з:

I. Початкового додавання раундового ключа.

II. *Nr-1* раундів, кожен з яких складається з чотирьох етапів:

1. Підстановка байтів;
2. Зсув рядків;
3. Перемішування стовпців;
4. Додавання раундового ключа.

III. Завершального раунду *Nr*, в якому пропускається перемішування стовпців. Розглянемо кожен з чотирьох етапів детальніше.

Підстановка байтів

Виконується окремо для кожного байта і складається з двох послідовних перетворень.

1. Байт розглядають як елемент поля GF(2^8). Якщо він ненульовий, до нього шукають обернений відносно множення в полі GF(2^8). Якщо ж байт нульовий, оберненого не існує. Тому нульовому байту 00000000 відповідає він сам.

2. Над утвореним байтом виконують таке перетворення:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

На основі цих двох перетворень створено спеціальну таблицю замін байтів в шістнадцятковій системі, що називається *S*-боксом (табл. 4.1):

Таблиця 4.1. S-бокс алгоритму AES

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Приклад 4.5:

Отриману матрицю із прикладу 4.4 перетворимо за допомогою *S*-боксу:

$$\begin{pmatrix} 41 & 43 & 20 & 53 \\ 20 & 52 & 4D & 41 \\ 53 & 45 & 45 & 47 \\ 45 & 54 & 53 & 45 \end{pmatrix} \Rightarrow \begin{pmatrix} 83 & 1A & B7 & ED \\ B7 & 00 & E3 & 83 \\ ED & 6E & 6E & A0 \\ 6E & 20 & ED & 6E \end{pmatrix}.$$

Зсув рядків

Рядки стану циклічно зсувають на різні кількості байтів:

Nb	Кількість зсувів...			
	0-го рядка (-)	1-го рядка (*C1*)	2-го рядка (*C2*)	3-го рядка (*C3*)
4	0	1	2	3
6	0	1	2	3
8	0	1	3	4

Обернення етапу зсуву рядків полягає у циклічному зсуві праворуч трьох нижніх рядків на *Nb-C1*, *Nb-C2*, *Nb-C3* байтів відповідно.

Перемішування стовпців

Стовпці стану розглядають як многочлен над полем GF(2^8) та множать за модулем $x^4 + 1$ на фіксований многочлен $c(x)$:

$$c(x) = 03_{16} \cdot x^3 + 01_{16} \cdot x^2 + 01_{16} \cdot x + 02_{16}.$$

Якщо $a(x)$ – стовпець до застосування до нього перемішування, а $b(x)$ – після, то перетворення можна записати так:

$$b(x) = c(x) \otimes a(x),$$

або у матричному вигляді:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

Додавання раундового ключа

Побітове додавання за модулем 2 раундового ключа до відповідних бітів, отриманих у попередньому циклі. Раундовий ключ отримують з розширеного ключа шифру. Довжина раундового ключа дорівнює довжині блока *Nb*.

Генерація ключів. *Розширений ключ* – одновимірний масив 4-байтових слів – позначають $W[Nb \cdot (Nr + 1)]$.

Алгоритм розширення ключа при $Nk \leq 6$

1. Перші Nk 4-байтових слів $W[i]$ послідовно вибираються з ключа шифру: 0-е слово – перші чотири байти, 1-е слово – другі чотири байти і т.д.

2. У слові $W[i-1]$ виконують циклічний зсув байтів за схемою: $(a,b,c,d) \Rightarrow (b,c,d,a)$, де a,b,c,d – байти.

3. Потім до кожного з 4-х байтів одержаного слова застосовують S-блок. До результату додають раундову сталу за модулем 2 (табл. 4.2).

Таблиця 4.2. Масив раундових констант Rcon

01	02	04	08	10	20	40	80	1B	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

4. Решту слів $W[i]$ визначають за формулою: $W[i] = W[i-Nk] \oplus W[i-1]$

При $Nk > 6$ виконується те саме, за винятком одного: якщо $i-4$ кратне Nk, то перед кроком 4 до кожного байта слова ще раз застосовують S-блок.

Дешифрування:

I. Перед першим раундом дешифрування виконується операція додавання з ключем.

II. Потім виконується 9 раундів дешифрування, кожен з яких здійснює такі операції:

1. Зсув рядків в зворотному порядку. Байти в останніх трьох рядках матриці зсуваються циклічно вліво на різне число байт.

2. Обернена операція до операції підстановки байтів. Байти матриці замінюються новими значеннями за таблицею зворотної заміни, що є інвертованим S-боксом (табл. 4.2).

Таблиця 4.2. Інвертований S-бокс алгоритму AES

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

3. Процедура, зворотна процедурі перемішування стовпців. Кожен стовпець матриці розглядається як 4-бітовий многочлен над полем GF(2^8) і множиться на фіксований многочлен:

$c^{-1}(x) = 0b_{16} \cdot x^3 + 0d_{16} \cdot x^2 + 09_{16} \cdot x + 0e_{16}$ по модулю многочлена $x^4 + 1$.

Таку операцію можна записати в матричному вигляді:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 01 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 01 & 0e \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

4. Операція додавання з ключем по модулю 2.

III. Завершальний раунд не містить операцію перемішування стовпців.

Завдання до лабораторної роботи

Завдання 1

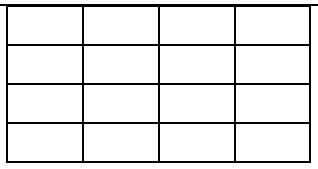
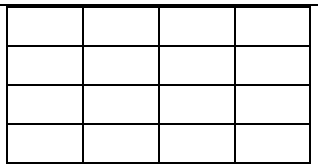
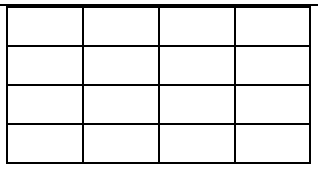
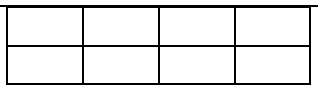
Виконати зашифрування блоку даних відкритого тексту за допомогою алгоритму AES на основі навчальної програми CrypTool 2 (Templates⇒Cryptography⇒Modern⇒Symmetric⇒AES Visualization) згідно варіанту.

Варіант №	Блок відкритого тексту	Ключ
1.	0102030405060708091011121314151 6	0102030405060708090A0B0C0D0E0F00
2.	1020304050607080910111213141516 0	020406080A0C0E10121416181A1C1E00
3.	0203040506070809101112131415160 1	04080C0014181C2024282C3034383C00
4.	2030405060708091011121314151601 0	0810180028303840485058606870780 0

Варіант №	Блок відкритого тексту	Ключ
5.	0304050607080910111212131415160102	10203000506070809 0A0B0C0D0E0F000
6.	3040506070809101112131415160102 0	20406000A0C0E10121416181A1C1E000
7.	0405060708091011121314151601020 3	4080C1014181C2024282C3034383C000
8.	4050607080910111213141516010203 0	810182028303840485058606870780 00
9.	0506070809101112131415160102030 4	0203040506070809 0A0B0C0D0E0F0001
10.	5060708091011121314151601020304 0	0406080A0C0E10121416181A1C1E0002
11.	0607080910111213141516010203040 5	080C1004181C2024282C3034383C0004
12.	6070809101112131415160102030405 0	101820083038404850586068707800 08
13.	0708091011121314151601020304050 6	2030400060708090A0B0C0D0E0F00010
14.	7080910111213141516010203040506 0	406080A0C0E10121416181A1C1E00020
15.	0809101112131415160102030405060 7	810182028303840485058606870780 00

1.1. Сформувати раундові ключі для зашифрування даних. У звіті описати зі скріншотами кроки алгоритму генерації ключів згідно схеми:

Генерація ключів

Ключ (128 бітів) у 16-ій системі числення:	
Початковий ключ (128 бітів) у вигляді матриці байтів:	
Циклічний зсув **w3**:	
Результат SubBytes(**w3**):	
w4 = SubBytes(**w3**) ⊕ Rcon(1) ⊕ **w0**:	
w5 = **w1** ⊕ **w4**:	
w6 = **w2** ⊕ **w5**:	
w7 = **w3** ⊕ **w6**:	
Ключ 1-го раунду (128 бітів) у вигляді матриці байтів:	
Ключ 2-го раунду (128 бітів) у вигляді матриці байтів:	
• • •	
Ключ 10-го раунду (128 бітів) у вигляді матриці байтів:	

	w40 **w41** **w42** **w43**

1.2. Обчислити вручну значення слів ключа **Wn** та **Wm** згідно варіанту та описати усі дії у звіті:

Варіант №	**n**	**m**
1.	32	33
2.	12	13
3.	16	17
4.	8	9
5.	24	25
6.	28	29
7.	32	33
8.	20	21
9.	36	37
10.	8	9
11.	24	25
12.	28	29
13.	40	41
14.	12	13
15.	20	21

1.3. Виконати зашифрування блоку повідомлення згідно схеми:

Зашифрування блоку

Блок повідомлення (128 бітів) у 16-ій системі числення:	
Блок повідомлення (128 бітів) у вигляді матриці стану:	
Додавання матриці стану з початковим ключем (AddRoundKey):	
Раунд 1	
Підстановка байтів з використанням S-боксу (SubBytes):	
Зсув рядків (ShiftRows):	
Перемішування стовпців (MixColumns):	
Додавання з ключем 1-го раунду (AddRoundKey):	
Результуюча матриця стану 1-го раунду:	
Раунд 2	

Результуюча матриця стану 2-го раунду:	
Раунд 3	
Результуюча матриця стану 3-го раунду:	
…	
Раунд 10	
Підстановка байтів з використанням S-боксу (SubBytes):	
Зсув рядків (ShiftRows):	
Додавання з ключем 10-го раунду (AddRoundKey):	
Результуюча матриця стану 10-го раунду:	
Результат шифрування блоку:	

1.4. Описати з усіма обчисленнями операцію *Перемішування стовпців* (*MixColumns*) для байта $\mathbf{s_{i,j}}$ раунду **r** згідно варіанту (усі операції над байтом виковувати як над многочленом в полі $GF(2^8)$):

Варіант №	$\mathbf{s_{i,j}}$	**r**
1.	$s_{0,0}$	2
2.	$s_{0,1}$	9
3.	$s_{0,2}$	3
4.	$s_{0,3}$	4
5.	$s_{1,0}$	7
6.	$s_{1,1}$	3
7.	$s_{1,2}$	9
8.	$s_{1,3}$	5
9.	$s_{2,0}$	2
10.	$s_{2,1}$	6
11.	$s_{2,2}$	8
12.	$s_{2,3}$	7
13.	$s_{3,1}$	9
14.	$s_{3,2}$	8
15.	$s_{3,3}$	4

1.5. Порівняти результати власних обчислень із результатами роботи програми.

Зауваження: *у візуалізації AES елементи результуючої матриці відображені дзеркально відносно головної діагоналі. Тому, якщо ви обчислили елемент* $s_{i,j}$, *то порівнюйте його з елементом матриці за індексом* $s_{j,i}$.

Завдання 2

Порівняння результатів зашифрування блоків даних у різних режимах за допомогою алгоритму AES (рис. 4.1) на основі навчальної програми CrypTool 2 (Templates⇒Cryptography⇒ Modern⇒Symmetric⇒AES Cipher (Text Input)). Результати шифрування зі скріншотами описати у звіті.

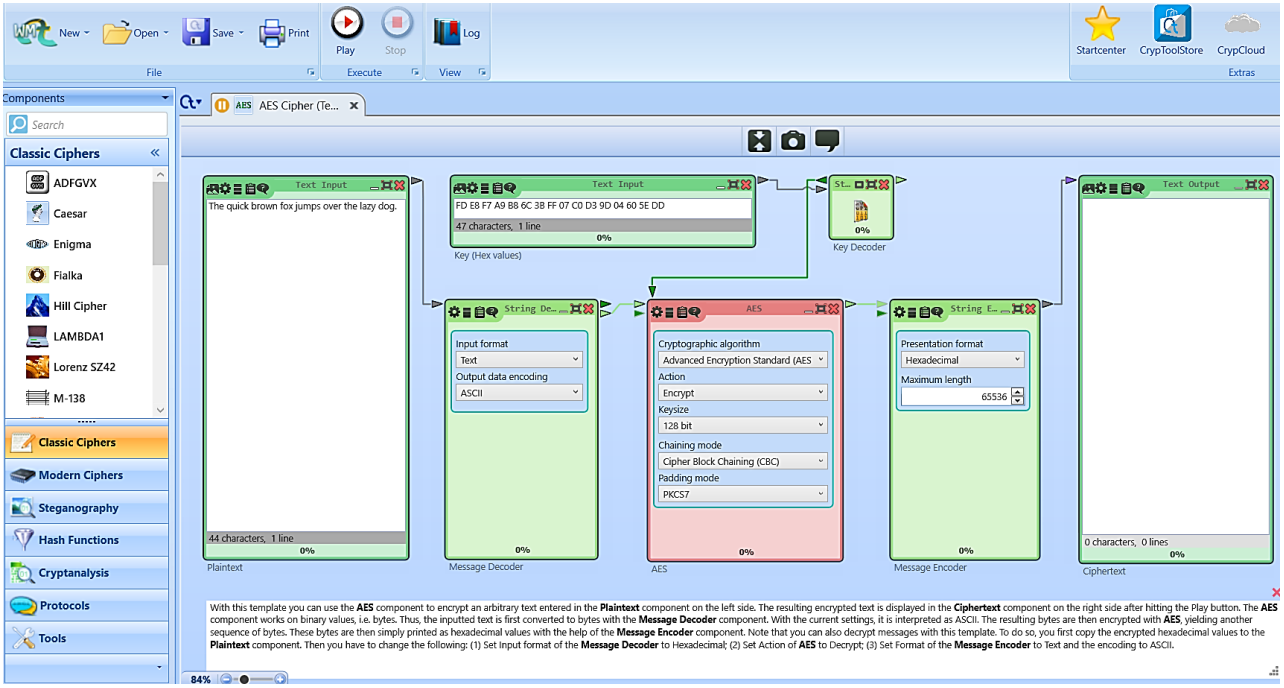


Рис. 4.1. Шаблон проекту, що реалізує шифрування тексту за алгоритмом AES

2.1. Видозмінити шаблон проекту, додавши до нього наступні компоненти «Text Input» та «Text Decoder» з панелі «Components», розділ «Tools».

2.2. Встановити зв’язки між компонентами за допомогою стрілок (перетягуванням). Наприклад, вектор ініціалізації 000102030405060708090A0B0C0D0E0F, що вводитиметься до текстового поля, потім передаватиметься до компоненту, що відповідає за декодування символів, після чого отриманий результат передається на виконання алгоритму шифрування і т.д. (рис. 4.2).

2.3. Переглянути та за необхідності змінити параметри компонентів.

Налаштування параметрів роботи компонентів (наприклад, визначення дії криптографічного перетворення, розмір ключа тощо) відбувається з використанням панелі «Parameter», що знаходиться праворуч робочого вікна, або за допомогою опції «Settings» у меню компоненту.

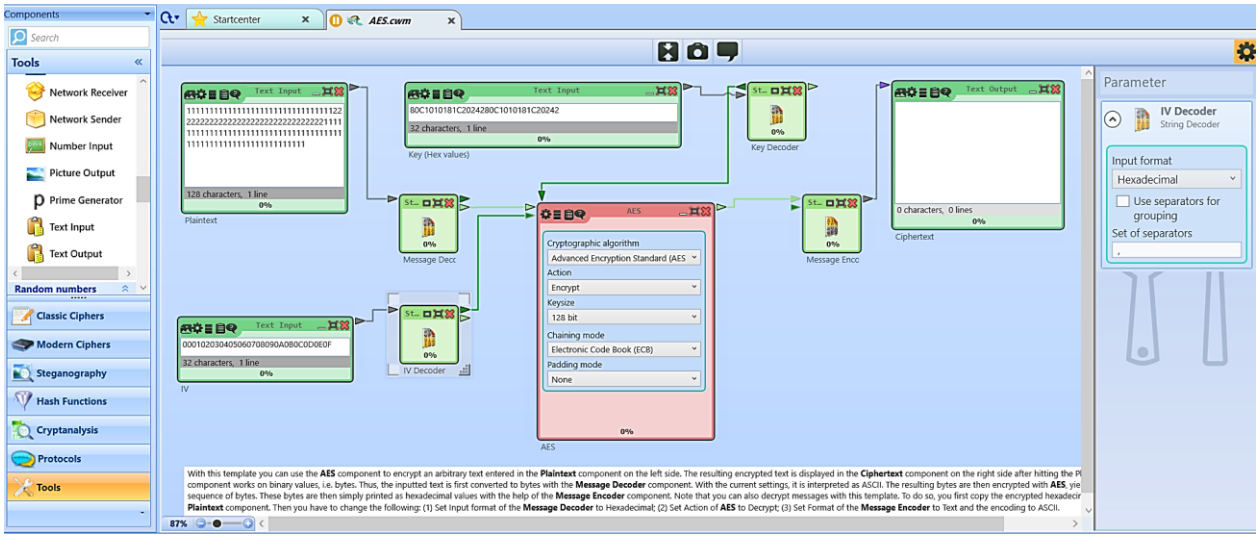


Рис. 4.2. Встановлення зв'язків між компонентами

2.4. Виконати зашифрування відкритого тексту, що містить символи, які часто повторюються у режимі простої заміни (ECB) з ключем 80C1010181C2024280C1010181C20242.

2.5. Виконати зашифрування відкритого тексту із п. 2.4 у режимі зв'язування блоків (CBC), порівняти результати шифрування та дати відповідь на питання «***Який режим роботи алгоритму виявився кращим? Чому?***».

Контрольні запитання:

1. Опишіть основні кроки зашифрування за алгоритмом AES.
2. Яка довжина блоку в AES?
3. Яка довжина ключа в AES?
4. Від чого залежить кількість раундів шифрування за алгоритмом AES?
5. Яким чином генеруються ключі в AES?
6. Які особливості дешифрування за алгоритмом AES?
7. Назвіть основні режими роботи блокових симетричних алгоритмів шифрування.

ТЕМА № 5. ДОСЛІДЖЕННЯ ОСНОВНИХ ОПЕРАЦІЙ ШИФРУ «КАЛИНА» У ПРОЦЕСІ ФОРМУВАННЯ ДОПОМІЖНОГО КЛЮЧА

Мета роботи: дослідити процес формування допоміжного ключа у шифрі «Калина», порівняти алгоритми шифрування AES та «Калина».

Матеріально-технічне забезпечення: ПК з доступом до мережі Інтернет.

Теоретичні відомості

НАЦІОНАЛЬНИЙ СТАНДАРТ ШИФРУВАННЯ ДСТУ 7624:2014 («КАЛИНА»)

«Калина» – блоковий симетричний шифр, описаний у національному стандарті України ***ДСТУ 7624:2014*** *«Інформаційні технології. Криптографічний захист інформації* (введений в дію з 1 липня 2015 р.).

Основні характеристики

1) Спроектований на основі SP-мережі (AES);

2) Забезпечує захист від відомих методів криптоаналізу;

3) Має високу швидкодію на сучасних і перспективних програмних та програмно-апаратних платформах;

4) Визначає 10 режимів роботи.

Розміри блока даних можуть бути такими: 128, 256 або 512 бітів. **Матриця стану** має 8 рядків та Nb стовпців, що являють собою елементи поля $GF(2^8)$.

Матриця стану при Nb=2:

$$\begin{pmatrix} s_{0,0} & s_{0,1} \\ s_{1,0} & s_{1,1} \\ s_{2,0} & s_{2,1} \\ s_{3,0} & s_{3,1} \\ s_{4,0} & s_{4,1} \\ s_{5,0} & s_{5,1} \\ s_{6,0} & s_{6,1} \\ s_{7,0} & s_{7,1} \end{pmatrix}.$$

Довжина ключа може також бути 128, 256 або 512 бітів. **Ключ** шифру розглядають як матрицю байтів, яка має матриця байтів, яка має 8 рядків та Nk стовпців.

Матриця ключа шифру при Nk=4:

$$\begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \\ k_{4,0} & k_{4,1} & k_{4,2} & k_{4,3} \\ k_{5,0} & k_{5,1} & k_{5,2} & k_{5,3} \\ k_{6,0} & k_{6,1} & k_{6,2} & k_{6,3} \\ k_{7,0} & k_{7,1} & k_{7,2} & k_{7,3} \end{pmatrix}.$$

Кількість раундів шифрування алгоритму «Калина» (*Nr*) залежить від значень *Nb* і *Nk*:

Розмір блоку	***Кількість раундів шифрування для різних довжин ключа***		
	*Довжина ключа **128** бітів (Nk = 2)*	*Довжина ключа **256** бітів (Nk = 4)*	*Довжина ключа **512** бітів (Nk = 8)*
128 *(Nb = 2)*	10	14	–
256 *(Nb = 4)*	–	14	18
512 *(Nb = 8)*	–	–	18

Шифрування за алгоритмом «Калина» складається з:

I. Додавання з нульовим ключем по модулю 2^{64}.

II. *Nr-1* раундів, кожен з яких складається з чотирьох етапів:

1. Підстановка байтів;
2. Зсув рядків;
3. Перемішування стовпців;
4. Додавання раундового ключа по модулю 2

III. Завершальний раунд *Nr*, в якому замість $\oplus$ виконується додавання по модулю 2^{64}.

Розглянемо кожен з чотирьох етапів детальніше.

Додавання з нульовим ключем по модулю 2^{64}

Операція $\boxplus$ забезпечує побітове додавання раундового ключа до матриці стану за модулем 2^{64}.

Підстановка байтів

Кожен байт матриці стану замінюється відповідно до заданої таблиці підстановки (табл. 5.1). Задано (рекомендовано) чотири таблиці підстановок

«байт-у-байт». Причому для байтів одного рядка поточного стану шифру застосовано одну й ту саму підстановку.

Заміна одного байту полягає у виборі з таблиці підстановки нового значення за адресою, яку задає поточне значення байту.

Таблиця 5.1. Підстановки алгоритму «Калина»

Підстановка π_0:

A8	43	5F	06	6B	75	6C	59	71	DF	87	95	17	F0	D8	09
6D	F3	1D	CB	C9	4D	2C	AF	79	E0	97	FD	6F	4B	45	39
3E	DD	A3	4F	B4	B6	9A	0E	1F	BF	15	E1	49	D2	93	C6
92	72	9E	61	D1	63	FA	EE	F4	19	D5	AD	58	A4	BB	A1
DC	F2	83	37	42	E4	7A	32	9C	CC	AB	4A	8F	6E	04	27
2E	E7	E2	5A	96	16	23	2B	C2	65	66	0F	BC	A9	47	41
34	48	FC	B7	6A	88	A5	53	86	F9	5B	DB	38	7B	C3	1E
22	33	24	28	36	C7	B2	3B	8E	77	BA	F5	14	9F	08	55
9B	4C	FE	60	5C	DA	18	46	CD	7D	21	B0	3F	1B	89	FF
EB	84	69	3A	9D	D7	D3	70	67	40	B5	DE	5D	30	91	B1
78	11	01	E5	00	68	98	A0	C5	02	A6	74	2D	0B	A2	76
B3	BE	CE	BD	AE	E9	8A	31	1C	EC	F1	99	94	AA	F6	26
2F	EF	E8	8C	35	03	D4	7F	FB	05	C1	5E	90	20	3D	82
F7	EA	0A	0D	7E	F8	50	1A	C4	07	57	B8	3C	62	E3	C8
AC	52	64	10	D0	D9	13	0C	12	29	51	B9	CF	D6	73	8D
81	54	C0	ED	4E	44	A7	2A	85	25	E6	CA	7C	8B	56	80

Підстановка π_1:

CE	BB	EB	92	EA	CB	13	C1	E9	3A	D6	B2	D2	90	17	F8
42	15	56	B4	65	1C	88	43	C5	5C	36	BA	F5	57	67	8D
31	F6	64	58	9E	F4	22	AA	75	0F	02	B1	DF	6D	73	4D
7C	26	2E	F7	08	5D	44	3E	9F	14	C8	AE	54	10	D8	BC
1A	6B	69	F3	BD	33	AB	FA	D1	9B	68	4E	16	95	91	EE
4C	63	8E	5B	CC	3C	19	A1	81	49	7B	D9	6F	37	60	CA
E7	2B	48	FD	96	45	FC	41	12	0D	79	E5	89	8C	E3	20
30	DC	B7	6C	4A	B5	3F	97	D4	62	2D	06	A4	A5	83	5F
2A	DA	C9	00	7E	A2	55	BF	11	D5	9C	CF	0E	0A	3D	51
7D	93	1B	FE	C4	47	09	86	0B	8F	9D	6A	07	B9	B0	98
18	32	71	4B	EF	3B	70	A0	E4	40	FF	C3	A9	E6	78	F9
8B	46	80	1E	38	E1	B8	A8	E0	0C	23	76	1D	25	24	05
F1	6E	94	28	9A	84	E8	A3	4F	77	D3	85	E2	52	F2	82
50	7A	2F	74	53	B3	61	AF	39	35	DE	CD	1F	99	AC	AD
72	2C	DD	D0	87	BE	5E	A6	EC	04	C6	03	34	FB	DB	59
B6	C2	01	F0	5A	ED	A7	66	21	7F	8A	27	C7	C0	29	D7

Підстановка π_2:

93	D9	9A	B5	98	22	45	FC	BA	6A	DF	02	9F	DC	51	59
4A	17	2B	C2	94	F4	BB	A3	62	E4	71	D4	CD	70	16	E1
49	3C	C0	D8	5C	9B	AD	85	53	A1	7A	C8	2D	E0	D1	72
A6	2C	C4	E3	76	78	B7	B4	09	3B	0E	41	4C	DE	B2	90
25	A5	D7	03	11	00	C3	2E	92	EF	4E	12	9D	7D	CB	35
10	D5	4F	9E	4D	A9	55	C6	D0	7B	18	97	D3	36	E6	48
56	81	8F	77	CC	9C	B9	E2	AC	B8	2F	15	A4	7C	DA	38
1E	0B	05	D6	14	6E	6C	7E	66	FD	B1	E5	60	AF	5E	33
87	C9	F0	5D	6D	3F	88	8D	C7	F7	1D	E9	EC	ED	80	29
27	CF	99	A8	50	0F	37	24	28	30	95	D2	3E	5B	40	83
B3	69	57	1F	07	1C	8A	BC	20	EB	CE	8E	AB	EE	31	A2
73	F9	CA	3A	1A	FB	0D	C1	FE	FA	F2	6F	BD	96	DD	43
52	B6	08	F3	AE	BE	19	89	32	26	B0	EA	4B	64	84	82
6B	F5	79	BF	01	5F	75	63	1B	23	3D	68	2A	65	E8	91
F6	FF	13	58	F1	47	0A	7F	C5	A7	E7	61	5A	06	46	44
42	04	A0	DB	39	86	54	AA	8C	34	21	8B	F8	0C	74	67

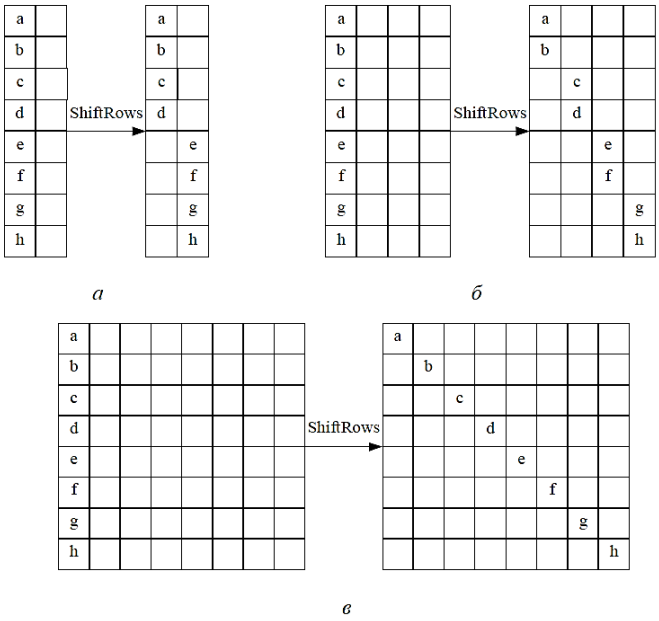
Підстановка π_3:

68	8D	CA	4D	73	4B	4E	2A	D4	52	26	B3	54	1E	19	1F
22	03	46	3D	2D	4A	53	83	13	8A	B7	D5	25	79	F5	BD
58	2F	0D	02	ED	51	9E	11	F2	3E	55	5E	D1	16	3C	66
70	5D	F3	45	40	CC	E8	94	56	08	CE	1A	3A	D2	E1	DF
B5	38	6E	0E	E5	F4	F9	86	E9	4F	D6	85	23	CF	32	99
31	14	AE	EE	C8	48	D3	30	A1	92	41	B1	18	C4	2C	71
72	44	15	FD	37	BE	5F	AA	9B	88	D8	AB	89	9C	FA	60
EA	BC	62	0C	24	A6	A8	EC	67	20	DB	7C	28	DD	AC	5B
34	7E	10	F1	7B	8F	63	A0	05	9A	43	77	21	BF	27	09
C3	9F	B6	D7	29	C2	EB	C0	A4	8B	8C	1D	FB	FF	C1	B2
97	2E	F8	65	F6	75	07	04	49	33	E4	D9	B9	D0	42	C7
6C	90	00	8E	6F	50	01	C5	DA	47	3F	CD	69	A2	E2	7A
A7	C6	93	0F	0A	06	E6	2B	96	A3	1C	AF	6A	12	84	39
E7	B0	82	F7	FE	9D	87	5C	81	35	DE	B4	A5	FC	80	EF
CB	BB	6B	76	BA	5A	7D	78	0B	95	E3	AD	74	98	3B	36
64	6D	DC	F0	59	A9	4C	17	7F	91	B8	C9	57	1B	E0	61

Зсув рядків

Рядки стану циклічно зсувають праворуч на різну кількість байтів, залежно від розміру блока (рис. 5.1):

Номер рядка	Значення зсуву, байтів		
	Довжина блоку 128 бітів	*Довжина блоку 256 бітів*	*Довжина блоку 512 бітів*
0	0	0	0
1	0	0	1
2	0	1	2
3	0	1	3
4	1	2	4
5	1	2	5
6	1	3	6
7	1	3	7



а *б* *в*

Рис 5.1. Зсув рядків: а – 128-бітовий блок; б – 256-бітовий блок; в – 512-бітовий блок

Перемішування стовпців

Стовпці стану розглядають як многочлен над полем $GF(2^8)$ та множать за модулем $x^8 + 1$ на фіксований многочлен $c(x)$:

$$c(x) = 01_{16} \cdot x^7 + 05_{16} \cdot x^6 + 01_{16} \cdot x^5 + 08_{16} \cdot x^4 + 06_{16} \cdot x^3 + 07_{16} \cdot x^2 + 04_{16} \cdot x + 01_{16}.$$

Або у матричному вигляді:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 01 & 01 & 05 & 01 & 08 & 06 & 07 & 04 \\ 04 & 01 & 01 & 05 & 01 & 08 & 06 & 07 \\ 07 & 04 & 01 & 01 & 05 & 01 & 08 & 06 \\ 06 & 07 & 04 & 01 & 01 & 05 & 01 & 08 \\ 08 & 06 & 07 & 04 & 01 & 01 & 05 & 01 \\ 01 & 08 & 06 & 07 & 04 & 01 & 01 & 05 \\ 05 & 01 & 08 & 06 & 07 & 04 & 01 & 01 \\ 01 & 05 & 01 & 08 & 06 & 07 & 04 & 01 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}.$$

Для множення у полі $GF(2^8)$ алгоритм «Калина» використовує нерозкладний многочлен $m(x) = x^8 + x^4 + x^3 + x^2 + 1$.

Додавання раундового ключа

Побітове додавання за модулем 2 раундового ключа до відповідних бітів, отриманих у попередньому раунді.

Розгортання ключів:

1. З ключа шифрування K формується допоміжний ключ K_t з довжиною, що дорівнює розміру блока ($64 \times Nb$ біт) з використанням трьох раундів

зашифрування. Вхіднимим даними для перетворення є число $Nb + Nk + 1$ (у двійковому вигляді), інші байти заповнюються нулями. У якості раундових ключів використовується ключ шифрування K (якщо ключ довше блоку, використовується його молодша і старша половини).

2. На основі ключа K та допоміжного ключа K_t формуються раундові ключі K_{2i} (з парними індексами) довжиною, що дорівнює розміру блока ($64 \times Nb$ біт), з використанням двох раундів зашифрування для кожного раундового ключа. У якості раундових ключів використовується результат додавання по модулю 2^{64} допоміжного ключа K_t та змінної tmv_i – двійкове значення, яке залежить від індексу раундового ключа, який формується.

3. З раундових ключів K_{2i} з парними індексами формуються раундові ключі K_{2i+1} (з непарними індексами) шляхом циклічного зсуву попереднього ключа з парним індексом вліво на $2 \times Nb + 3$ байт.

Загалом використовується $Nr+1$ раундових ключів K_i ($i = 0,1 .., Nr$), кожен довжиною $64 \times Nb$ біт.

Дешифрування:

I. Виконуються операції з п. II, але на початку замість $\oplus$ виконується віднімання по модулю 2^{64} з ключем останнього раунду.

II. $Nr-1$ раундів, кожен з яких складається з чотирьох етапів:

1. Додавання раундового ключа за модулем 2;
2. Зворотна операція до перемішування стовпців;
3. Зсув рядків в зворотному порядку;
4. Обернена операція до підстановки байтів.

III. Віднімання з ключем нульового раунду по модулю 2^{64}.

Розглянемо кожен з чотирьох етапів детальніше.

Операція, зворотна операції перемішування стовпців

Стовпці стану множать на фіксований многочлен $c^{-1}(x)$ обернений до $c(x)$:

$$c^{-1}(x) = 95_{16} \cdot x^7 + 76_{16} \cdot x^6 + A8_{16} \cdot x^5 + 2F_{16} \cdot x^4 + 49_{16} \cdot x^3 + D7_{16} \cdot x^2 + CA_{16} \cdot x + AD_{16}.$$

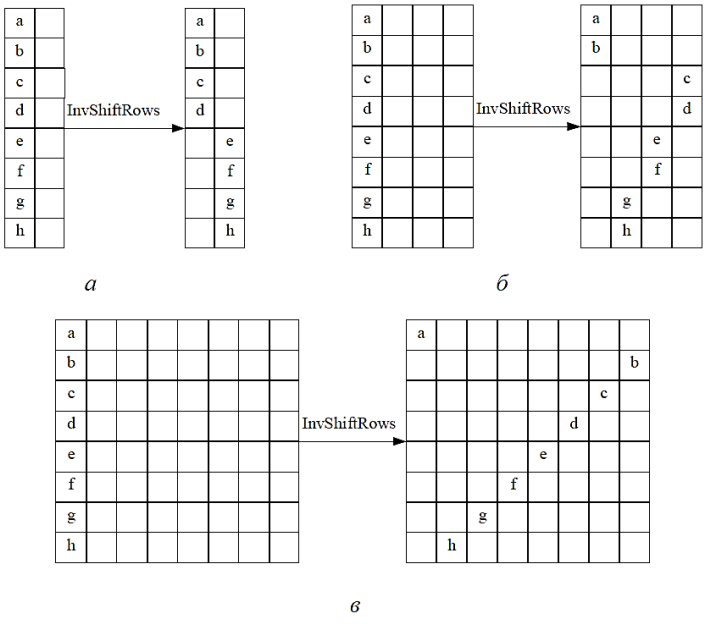
Або у матричному вигляді:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} AD & 95 & 76 & A8 & 2F & 49 & D7 & CA \\ CA & AD & 95 & 76 & A8 & 2F & 49 & D7 \\ D7 & CA & AD & 95 & 76 & A8 & 2F & 49 \\ 49 & D7 & CA & AD & 95 & 76 & A8 & 2F \\ 2F & 49 & D7 & CA & AD & 95 & 76 & A8 \\ A8 & 2F & 49 & D7 & CA & AD & 95 & 76 \\ 76 & A8 & 2F & 49 & D7 & CA & AD & 95 \\ 95 & 76 & A8 & 2F & 49 & D7 & CA & AD \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}.$$

Зсув рядків в зворотному порядку

Рядки стану циклічно зсувають ліворуч на різну кількість байтів, залежно від розміру блока (рис. 5.2):

Номер рядка	Значення зсуву, байтів		
	Довжина блоку 128 бітів	*Довжина блоку 256 бітів*	*Довжина блоку 512 бітів*
0	0	0	0
1	0	0	1
2	0	1	2
3	0	1	3
4	1	2	4
5	1	2	5
6	1	3	6
7	1	3	7



а *б* *в*

Рис 5.2. Зсув рядків в обереному порядку: а – 128-бітовий блок; б – 256-бітовий блок; в – 512-бітовий блок

Обернена операція до операції підстановки байтів

Кожен байт матриці стану замінюється відповідно до заданої таблиці зворотної заміни (табл. 5.2).

Таблиця 5.2. Обернені підстановки алгоритму «Калина»

Підстановка $_{-1}\pi_0$:

A4	A2	A9	C5	4E	C9	03	D9	7E	0F	D2	AD	E7	D3	27
E3	A1	E8	E6	7C	2A	55	0C	86	39	D7	8D	B8	12	6F
CD	8A	70	56	72	F9	BF	4F	73	E9	F7	57	16	AC	50
9D	B7	47	71	60	C4	74	43	6C	1F	93	77	DC	CE	20
99	5F	44	01	F5	1E	87	5E	61	2C	4B	1D	81	15	F4
D6	EA	E1	67	F1	7F	FE	DA	3C	07	53	6A	84	9C	CB
83	33	DD	35	E2	59	5A	98	A5	92	64	04	06	10	4D
97	08	31	EE	AB	05	AF	79	A0	18	46	6D	FC	89	D4
FF	F0	CF	42	91	F8	68	0A	65	8E	B6	FD	C3	EF	78
CC	9E	30	2E	BC	0B	54	1A	A6	BB	26	80	48	94	32
A7	3F	AE	22	3D	66	AA	F6	00	5D	BD	4A	E0	3B	B4
8B	9F	76	B0	24	9A	25	63	DB	EB	7A	3E	5C	B3	B1
F2	CA	58	6E	D8	A8	2F	75	DF	14	FB	13	49	88	B2
E4	34	2D	96	C6	3A	ED	95	0E	E5	85	6B	40	21	9B
19	2B	52	DE	45	A3	FA	51	C2	B5	D1	90	B9	F3	37
0D	BA	41	11	38	7B	BE	D0	D5	69	36	C8	62	1B	82

Підстановка $_{-1}\pi_1$:

83	F2	2A	EB	E9	BF	7B	9C	34	96	8D	98	B9	69	8C
3D	88	68	06	39	11	4C	0E	A0	56	40	92	15	BC	B3
6F	F8	26	BA	BE	BD	31	FB	C3	FE	80	61	E1	7A	32
70	20	A1	45	EC	D9	1A	5D	B4	D8	09	A5	55	8E	37
A9	67	10	17	36	65	B1	95	62	59	74	A3	50	2F	4B
D0	8F	CD	D4	3C	86	12	1D	23	EF	F4	53	19	35	E6
5E	D6	79	51	22	14	F7	1E	4A	42	9B	41	73	2D	C1
A6	A2	E0	2E	D3	28	BB	C9	AE	6A	D1	5A	30	90	84
B2	58	CF	7E	C5	CB	97	E4	16	6C	FA	B0	6D	1F	52
0D	4E	03	91	C2	4D	64	77	9F	DD	C4	49	8A	9A	24
A7	57	85	C7	7C	7D	E7	F6	B7	AC	27	46	DE	DF	3B
9E	2B	0B	D5	13	75	F0	72	B6	9D	1B	01	3F	44	E5
FD	07	F1	AB	94	18	EA	FC	3A	82	5F	05	54	DB	00
E3	48	0C	CA	78	89	0A	FF	3E	5B	81	EE	71	E2	DA
B8	B5	CC	6E	A8	6B	AD	60	C6	08	04	02	E8	F5	4F
F3	C0	CE	43	25	1C	21	33	0F	AF	47	ED	66	63	93

Підстановка $_{-1}\pi_2$:

45	D4	0B	43	F1	72	ED	A4	C2	38	E6	71	FD	B6	3A
50	44	4B	E2	74	6B	1E	11	5A	C6	B4	D8	A5	8A	70
A8	FA	05	D9	97	40	C9	90	98	8F	DC	12	31	2C	47
99	AE	C8	7F	F9	4F	5D	96	6F	F4	B3	39	21	DA	9C
9E	3B	F0	BF	EF	06	EE	E5	5F	20	10	CC	3C	54	4A
94	0E	C0	28	F6	56	60	A2	E3	0F	EC	9D	24	83	7E
7C	EB	18	D7	CD	DD	78	FF	DB	A1	09	D0	76	84	75
1D	1A	2F	B0	FE	D6	34	63	35	D2	2A	59	6D	4D	77
8E	61	CF	9F	CE	27	F5	80	86	C7	A6	FB	F8	87	AB
3F	DF	48	00	14	9A	BD	5B	04	92	02	25	65	4C	53
F2	29	AF	17	6C	41	30	E9	93	55	F7	AC	68	26	C4
CA	7A	3E	A0	37	03	C1	36	69	66	08	16	A7	BC	C5
22	B7	13	46	32	E8	57	88	2B	81	B2	4E	64	1C	AA
58	2E	9B	5C	1B	51	73	42	23	01	6E	F3	0D	BE	3D
2D	1F	67	33	19	7B	5E	EA	DE	8B	CB	A9	8C	8D	AD
82	E4	BA	C3	15	D1	E0	89	FC	B1	B9	B5	07	79	B8

Підстановка $_{-1}\pi_3$:

B2	B6	23	11	A7	88	C5	A6	39	8F	C4	E8	73	22	43	C3
82	27	CD	18	51	62	2D	F7	5C	0E	3B	FD	CA	9B	0D	0F
79	8C	10	4C	74	1C	0A	8E	7C	94	07	C7	5E	14	A1	21
57	50	4E	A9	80	D9	EF	64	41	CF	3C	EE	2E	13	29	BA
34	5A	AE	8A	61	33	12	B9	55	A8	15	05	F6	03	06	49
B5	25	09	16	0C	2A	38	FC	20	F4	E5	7F	D7	31	2B	66
6F	FF	72	86	F0	A3	2F	78	00	BC	CC	E2	B0	F1	42	B4
30	5F	60	04	EC	A5	E3	8B	E7	1D	BF	84	7B	E6	81	F8
DE	D8	D2	17	CE	4B	47	D6	69	6C	19	99	9A	01	B3	85
B1	F9	59	C2	37	E9	C8	A0	ED	4F	89	68	6D	D5	26	91
87	58	BD	C9	98	DC	75	C0	76	F5	67	6B	7E	EB	52	CB
D1	5B	9F	0B	DB	40	92	1A	FA	AC	E4	E1	71	1F	65	8D
97	9E	95	90	5D	B7	C1	AF	54	FB	02	E0	35	BB	3A	4D
AD	2C	3D	56	08	1B	4A	93	6A	AB	B8	7A	F2	7D	DA	3F
FE	3E	BE	EA	AA	44	C6	D0	36	48	70	96	77	24	53	DF
F3	83	28	32	45	1E	A4	D3	A2	46	6E	9C	DD	63	D4	9D

Завдання до лабораторної роботи

Завдання 1

Дослідити процес формування допоміжного ключа $\boldsymbol{K_t}$ *на основі ключа* $\boldsymbol{K}$ *довжиною 128 бітів згідно варіанту:*

Варіант №	Ключ
1.	20406000A0C0E10121416181A1C1E000
2.	4080C1014181C2024282C3034383C000
3.	810182028303840485058606870780000
4.	02030405060708090A0B0C0D0E0F0001
5.	0406080A0C0E10121416181A1C1E0002
6.	080C1004181C2024282C3034383C0004
7.	1018200830384048505860687078000 8
8.	2030400060708090A0B0C0D0E0F00010
9.	406080A0C0E10121416181A1C1E00020

Варіант №	Ключ
10.	80C1010181C2024282C3034383C00040
11.	018202030384048505860687078000810
12.	030405060708090A0B0C0D0E0F000102
13.	06080A0C0E10121416181A1C1E000204
14.	0C1014081C2024282C3034383C000408
15.	182028003840485058606870780008l0

У схемі формування допоміжного ключа використовується **три раунди** зашифрування.

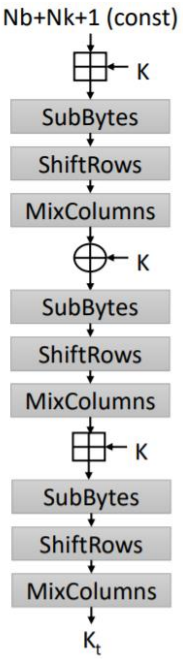


Рис 5.3. Формування допоміжного ключа у шифрі «Калина»

Вхідні дані: Початкова матриця стану ***Nb+Nk+1=2+2+1=5*** (128 бітів) у 16-ій системі числення:

05	00
00	00
00	00
00	00
00	00
00	00
00	00
00	00

Ключ шифрування: початковий ключ ***K*** (128 бітів) у 16-ій системі числення:

00	08
01	09
02	0A
03	0B
04	0C
05	0D
06	0E
07	0F

Додавання матриці стану з ключем ***K*** за модулем 2^{64}:

05	00	⊞	00	08	=	05	08
00	00		01	09		01	09
00	00		02	0A		02	0A
00	00		03	0B		03	0B
00	00		04	0C		04	0C
00	00		05	0D		05	0D
00	00		06	0E		06	0E
00	00		07	0F		07	0F

Раунд 1

Підстановка байтів:

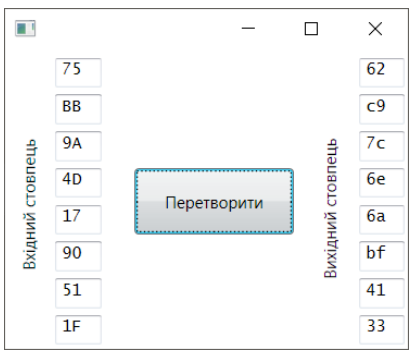
05	08	→	75	71
01	09		BB	3A
02	0A		9A	DF
03	0B		4D	B3
04	0C		6B	17
05	0D		CB	90
06	0E		45	51
07	0F		2A	1F

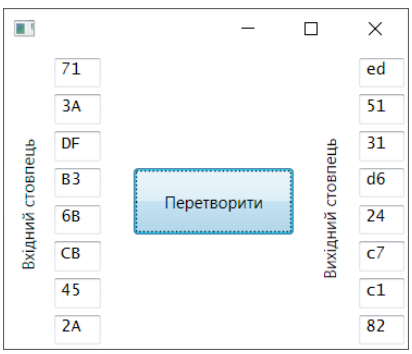
Зсув рядків:

75	71
BB	3A
9A	DF
4D	B3
17	6B
90	CB
51	45
1F	2A

Перемішування стовпців:

01	01	05	01	08	06	07	04	·	75	71	=	62	ED
04	01	01	05	01	08	06	07		BB	3A		C9	51
07	04	01	01	05	01	08	06		9A	DF		7C	31
06	07	04	01	01	05	01	08		4D	B3		6E	D6
08	06	07	04	01	01	05	01		17	6B		6A	24
01	08	06	07	04	01	01	05		90	CB		BF	C7
05	01	08	06	07	04	01	01		51	45		41	C1
01	05	01	08	06	07	04	01		1F	2A		33	82





Додавання ключа ***K*** по модулю 2:

62	ED		00	08		62	E5
C9	51		01	09		C8	58
7C	31		02	0A		7E	3B
6E	D6	⊕	03	0B	=	6D	DD
6A	24		04	0C		6E	28
BF	C7		05	0D		BA	CA
41	C1		06	0E		47	CF
33	82		07	0F		34	8D

Раунд 2

Підстановка байтів:

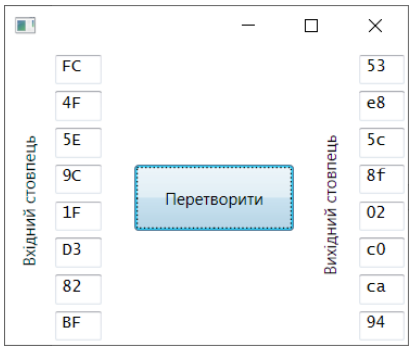
62	E5		FC	D9
C8	58		4F	81
7E	3B		5E	41
6D	DD	→	9C	FC
6E	28		C3	1F
BA	CA		23	D3
47	CF		2E	82
34	8D		40	BF

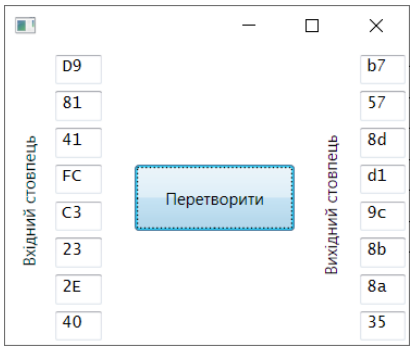
Зсув рядків:

FC	D9
4F	81
5E	41
9C	FC
1F	C3
D3	23
82	2E
BF	40

Перемішування стовпців:

01	01	05	01	08	06	07	04		FC	D9		53	B7
04	01	01	05	01	08	06	07		4F	81		E8	57
07	04	01	01	05	01	08	06		5E	41		5C	8D
06	07	04	01	01	05	01	08	·	9C	FC	=	8F	D1
08	06	07	04	01	01	05	01		1F	C3		02	9C
01	08	06	07	04	01	01	05		D3	23		C0	8B
05	01	08	06	07	04	01	01		82	2E		CA	8A
01	05	01	08	06	07	04	01		BF	40		94	35





Додавання матриці стану з ключем ***K*** за модулем 2^{64}:

53	B7	⊞	00	08	=	53	BF
E8	57		01	09		E9	60
5C	8D		02	0A		5E	97
8F	D1		03	0B		92	DC
02	9C		04	0C		06	A8
C0	8B		05	0D		C5	98
CA	8A		06	0E		D0	98
94	35		07	0F		9B	44

Раунд 3

Підстановка байтів:

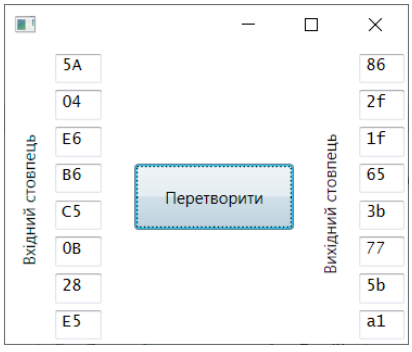
53	BF	→	5A	26
E9	60		04	E7
5E	97		E6	24
92	DC		B6	A5
06	A8		6C	C5
C5	98		84	0B
D0	98		6B	28
9B	44		1D	E5

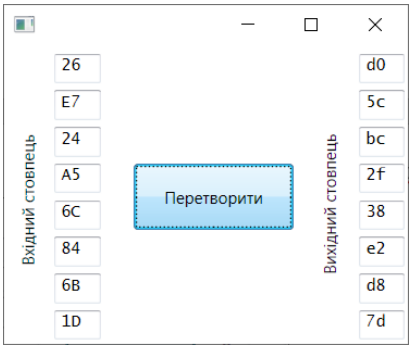
Зсув рядків:

5A	26
04	E7
E6	24
B6	A5
C5	6C
0B	84
28	6B
E5	1D

Перемішування стовпців:

01	01	05	01	08	06	07	04	·	5A	26	=	86	D0
04	01	01	05	01	08	06	07		04	E7		2F	5C
07	04	01	01	05	01	08	06		E6	24		1F	BC
06	07	04	01	01	05	01	08		B6	A5		65	2F
08	06	07	04	01	01	05	01		C5	6C		3B	38
01	08	06	07	04	01	01	05		0B	84		77	E2
05	01	08	06	07	04	01	01		28	6B		5B	D8
01	05	01	08	06	07	04	01		E5	1D		A1	7D





Допоміжний ключ K_t (128 бітів) у 16-ій системі числення:

86	D0
2F	5C
1F	BC
65	2F
3B	38
77	E2
5B	D8
A1	7D

Завдання 2

Провести порівняльну характеристику алгоритмів AES та «Калина». Додати до звіту таблицю та заповнити її:

	AES	***«Калина»***
Розмір ключа		
Розмір блока		
Кількість раундів		
Математичні операції		
Кількість таблиць підстановки		
Нерозкладний многочлен		
Генерація ключів (основні операції)		

Контрольні запитання:

1. Опишіть основні кроки зашифрування за алгоритмом «Калина».
2. Яка довжина блоку в алгоритмі «Калина»?
3. Яка довжина ключа в алгоритмі «Калина»?
4. Від чого залежить кількість раундів шифрування за алгоритмом «Калина»?
5. Яким чином генеруються ключі в «Калина»?
6. Які особливості дешифрування за алгоритмом «Калина»?
7. Назвіть основні режими роботи алгоритму шифрування «Калина».

ТЕМА № 6. АСИМЕТРИЧНІ ШИФРИ RSA ТА ЕЛЬ-ГАМАЛЯ. АЛГОРИТМ ОБМІНУ КЛЮЧАМИ ДІФФІ-ХЕЛМАНА

Мета роботи: набути уміння із генерації ключів, зашифрування і дешифрування повідомлення за допомогою алгоритмів RSA та Ель-Гамаля, дослідити алгоритм обміну ключами Діффі-Хеллмана, на практиці здійснити формування спільного ключа між двома абонентами.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням MS Excel та доступом до мережі Інтернет.

Теоретичні відомості

КРИПТОСИСТЕМИ З ВІДКРИТИМ КЛЮЧЕМ

Якими б не були надійними та швидкими симетричні криптографічні системи – їх слабким місцем, під час практичної реалізації, є проблема обміну ключами. Для вирішення цієї і ряду інших проблем були запропоновані криптосистеми з відкритим ключем, які називають також асиметричними криптосистемами.

Концепція криптографії з відкритим ключем була висунута Вітфілдом Діффі (Whitfield Diffie) та Мартіном Хелманом (Martin Hellman), і окремо Ральфом Мерклом (Ralph Merkle). У *асиметричних криптосистемах* для шифрування використовується один, відкритий (публічний, загальнодоступний) ключ, а для дешифрування – інший, закритий (секретний, приватний). Закритий ключ та відкритий ключ – це два великі числа, обчислені на основі деякого асиметричного алгоритму. Відкритий може бути доступним будь-якому учаснику процесу інформаційного обміну. При чому, знання відкритого ключа не дозволяє обчислити відповідний закритий ключ.

Ідея криптографії з відкритим ключем дуже тісно пов'язана з ідеєю *однобічних функцій*, тобто таких функцій $f(x)$, що по відомому x досить просто знайти значення $f(x)$, тоді як визначити x з $f(x)$складно (рис. 6.1).

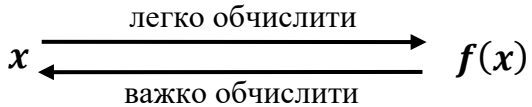


Рис. 6.1. Схема роботи однобічних функцій

Також використовуються *однобічні функції з лазівкою*. Лазівка – це певний секрет, що допомагає розшифрувати. Тобто існує такий y, що знаючи $f(x)$, можна обчислити x.

АЛГОРИТМ ШИФРУВАННЯ RSA

Найбільш простим для розуміння та реалізації є алгоритм з відкритим ключем RSA, названий на честь трьох авторів – Рона Рівеста (Ron Rivest), Аді Шаміра (Adi Shamir) і Леонарда Едлмана (Leonard Adleman).

Безпека RSA заснована на складності розкладання на множники великих чисел. Відкритий і закритий ключі є функціями двох великих простих чисел розрядністю 100…200 десяткових цифр і навіть більше. Відновлення відкритого тексту за шифртекстом та відкритим ключем є рівнозначне до розкладання числа на два великі прості множники.

Генерація ключів

1. Вибираються два великих випадкових простих числа, p і q (для максимальної безпеки p і q варто обирати рівної довжини).

2. Обчислюється добуток (модуль системи): $n = p \cdot q$.

3. Обчислюється функція Ейлера $\varphi(n) = \varphi(pq) = (p-1)(q-1)$. Результат розрахунку даної функції дорівнює кількості додатних чисел, які не більше n і взаємно прості з n.

4. Випадковим чином вибирається число e (ключ шифрування), таке що $1 < e < \varphi(n)$ та взаємно просте з $\varphi(n)$.

5. За допомогою розширеного алгоритму Евкліда знаходиться число d (ключ дешифрування), таке що $ed \equiv 1(mod\ \varphi(n))$.

6. Пара $(e,\ n)$ публікується у якості відкритого ключа.

7. Пара $(d,\ n)$ виконує роль секретного ключа і тримається таємниці.

Два простих числа p і q більше не потрібні. Проте вони не повинні бути розкриті.

Зашифрування

Для шифрування повідомлення *M* воно спочатку розбивається на цифрові блоки, менші n (для двійкових даних вибирається найбільший степінь числа 2, менший n). Зашифроване повідомлення *C* буде складатися із блоків c_i. Формула шифрування виглядає наступним чином: $c_i = m_i^e \, mod \, n$.

Дешифрування

Для дешифрування повідомлення візьмемо кожний зашифрований блок c_i і обчислимо: $m_i = c_i^d \, mod \, n$.

Приклад 6.1:

Зашифруємо повідомлення КНИГА, що складається із символів українського алфавіту та представляється як послідовність цілих чисел *M* = 14 17 10 3 0.

Для простоти обчислень будемо використовувати невеликі числа, проте пам'ятаємо, що на практиці застосовують дуже великі прості числа. Оберемо $p = 3$ і $q = 11$, тоді $n = p \cdot q = 3 \cdot 11 = 33$.

Обчислимо $\varphi(33) = 2 \cdot 10 = 20$.

Виберемо (випадково) $e = 3$ та перевіримо виконання умов: $1 < 3 < 20$, НСД(3, 20) =1.

Визначимо d – ключ дешифрування з рівняння $3d \equiv 1(mod\ 20)$.

Для розв'язання рівняння використаємо *розширений алгоритм Евкліда*:

1) послідовно виконуємо ділення з остачею попереднього значення r_{i-1} на наступне r_i, у відповідністю з рівністю $r_{i-1} = r_i q_{i+1} + r_{i+i}$ (якщо $r_i = 1$, тоді зупиняєио процес);

2) використовуємо рекурентне співвідношення $u_{i+1} = u_{i-1} - q_{i+1} u_i$;

3) використовуємо рекурентне співвідношення $v_{i+1} = v_{i-1} - q_{i+1} v_i$;

4) щоб почати процес виконання алгоритму, використовуємо значення $r_0 = 20$, $r_1 = 3$, $u_0 = 1$, $u_1 = 0$, $v_0 = 0$, $v_1 = 1$.

i	r_i	q_i	u_i	v_i
0	20		1	0
1	3		0	1
2	20 *mod* 3 =2	20 *div* 3 = 6	1-6·0=1	0-6·1=-6
3	3 *mod* 2 =1	3 *div* 2 = 1	0-1·1=-1	1-1·(-6)=7

Виконуємо перевірку $3 \cdot 7 \; mod \; 20 = 1$. Таким чином, $d = 7$.

Опублікуємо відкритий ключ $(e, \; n) \; = \; (3, 33)$.

Зберігаємо в таємниці секретний ключ $(d, \; n) \; = \; (7, 33)$.

Зашифруємо повідомлення *M* = 14 17 10 3 0, що складається із п’яти блоків m_i:

$c_1 = 14^3 \; mod \; 33 = \big((14^2 \; mod \; 33) \cdot (14^1 \; mod \; 33)\big) \; mod \; 33 = (31 \cdot 14) \; mod \; 33 =$
$= 434 \; mod \; 33 = 5;$
$c_2 = 17^3 \; mod \; 33 = \big((17^2 \; mod \; 33) \cdot (17^1 \; mod \; 33)\big) \; mod \; 33 = (25 \cdot 17) \; mod \; 33 =$
$= 425 \; mod \; 33 = 29;$
$c_3 = 10^3 \; mod \; 33 = 1000 \; mod \; 33 = 10;$
$c_4 = 3^3 \; mod \; 33 = 27 \; mod \; 33 = 27;$
$c_5 = 0^3 \; mod \; 33 = 0 \; mod \; 33 = 0.$

Шифротекст: C = 5 29 10 27 0.

Для дешифрування потрібно також виконати піднесення до степеня, використовуючи ключ дешифрування 7:

$m_1 = 5^7 \; mod \; 33 = ((5^4 \; mod \; 33) \cdot (5^3 \; mod \; 33)) \; mod \; 33 = (31 \cdot 26) \; mod \; 33 =$
$= 806 \; mod \; 33 = 14;$
$m_2 = 29^7 \; mod \; 33 = ((29^4 \; mod \; 33) \cdot (29^3 \; mod \; 33)) \; mod \; 33 =$
$= (((29^2))^2 \; mod \; 33) \cdot (29^2 \; mod \; 33) \cdot (29 \; mod \; 33)) \; mod \; 33 = (25 \cdot 16 \cdot 29) \; mod \; 33$
$= 11600 \; mod \; 33 = 17;$
$m_3 = 10^7 \; mod \; 33 = ((10^4 \; mod \; 33) \cdot (10^3 \; mod \; 33)) \; mod \; 33 =$
$= (((10^2))^2 \; mod \; 33) \cdot (10^2 \; mod \; 33) \cdot (10 \; mod \; 33)) \; mod \; 33 = (1 \cdot 1 \cdot 10) \; mod \; 33 =$
$= 10 \; mod \; 33 = 10;$
$m_4 = 27^7 \; mod \; 33 = ((27^4 \; mod \; 33) \cdot (27^3 \; mod \; 33)) \; mod \; 33 =$
$= (((27^2))^2 \; mod \; 33) \cdot (27^2 \; mod \; 33) \cdot (27 \; mod \; 33)) \; mod \; 33 = (9 \cdot 3 \cdot 27) \; mod \; 33 =$
$= 729 \; mod \; 33 = 3;$
$m_5 = 0^7 \; mod \; 33 = 0 \; mod \; 33 = 0.$

Відкритий текст: M = 14 17 10 3 0 $\Rightarrow$ КНИГА.

АЛГОРИТМ ШИФРУВАННЯ ЕЛЬ-ГАМАЛЯ

Алгоритм шифрування Ель-Гамаля (ElGamal) – криптосистема з відкритим ключем, заснована на складності обчислення дискретних логарифмів в скінченному полі. Шифр була запропонована американським вченим єгипетського походження Тахером Ель-Гамалем у 1984.

Генерація ключів

1. Генерується просте випадкове число p.

2. Вибирається генератор g, таке що $1 < g < p - 1$ та $g^{p-1} mod\, p = 1$.

3. Вибирається випадкове число x, таке що $1 < x < p - 1$.

4. Обчислюється $y = g^x \ mod\, p$.

5. Відкритими даними є p, g, y.

6. Закритим ключем є x.

Зашифрування

Повідомлення M шифрується таким чином:

Вибирається сесійний ключ – випадкове число k, таке що $1 < k < p - 1$.

Потім обчислюються $a = g^k \ mod\, p$ та $b = y^k M \ mod\, p$.

Пара чисел $(a,\ b)$ є шифротекстом.

Дешифрування

Для дешифрування $(a,\ b)$ обчислюється:

$M = b(a^x)^{-1} \ mod\, p$ або $M = b(a^x)^{-1} \ mod\, p = b \cdot a^{(p-1-x)} \ mod\, p$.

Приклад 6.2:

Зашифруємо повідомлення M = 5.

Спершу згенеруємо ключі шифрування. Нехай $p = 11$, $g = 2$.

Виберемо $x = 8$ – випадкове ціле число x таке, що таке що $1 < x < p - 1$.

Обчислимо $y = g^x \ mod\, p = 2^8 \ mod\, 11 = 3$.

Отже, відкритим даними є трійка є 11, 2 та 3, закритим ключем є число $x = 8$.

Для шифрування вибираємо випадкове ціле число $k = 9$ таке, що $1 < k < p - 1$.

Обчислюємо $a = g^k mod\, p = 2^9 mod\, 11 = 512 \ mod\, 11 = 6$.

Обчислюємо $b = y^k M \ mod\, p = 3^9 \cdot 5 \ mod\, 11 = 19683 \cdot 5 \ mod\, 11 = 9$.

Пара $(6,\ 9)$ є шифротекстом.

Шифротекст $(6,\ 9)$, закритий ключ $x = 8$.

Для дешифрування обчислюємо M за формулою:

$$M = b(a^x)^{-1} mod\ p == b \cdot a^{(p-1-x)}\ mod\ p = 9 \cdot\ 6^{(11-1-8)} mod\ 11 = 5.$$

Отримали початкове повідомлення $M = 5$.

АЛГОРИТМ ОБМІНУ КЛЮЧАМИ ДІФФІ-ХЕЛМАНА

Протокол обміну ключами Діффі-Хелмана дозволяє двом сторонам отримати спільний секретний ключ, використовуючи незахищений від прослуховування, але захищений від модифікації канал зв'язку. Отриманий ключ можна використовувати для симетричного шифрування повідомлень. Алгоритм заснований на складності обчислень дискретних логарифмів.

Припустимо, користувачі A і B мають намір обмінятися ключами за алгоритмом Діффі-Хелмана, суть якого полягає в наступному (рис. 6.2):

1. A і B спільно обирають просте число p і ціле число g таке, що $1 < g < p - 1$ і g є первісним коренем p.

Первісним коренем за модулем p називається таке число g, що при піднесення до степеню $g^i mod\ p$ всі його степені $i \in \{1, \dots, p-1\}$ за модулем p пробігають по всім числам взаємно простим із p.

Нехай $p = 5$. Усі взаємно прості числа з p: 1, 2, 3, 4.

*Елементи **2** та **3** є первісними коренями 5.*

1
$1^1 mod\ 5 = 1$
$1^2 mod\ 5 = 1$
$1^3 mod\ 5 = 1$
$1^4 mod\ 5 = 1$
2
$2^1 mod\ 5 = 2\ mod\ 5 = \mathbf{2}$
$2^2 mod\ 5 = 4\ mod\ 5 = \mathbf{4}$
$2^3 mod\ 5 = 8\ mod\ 5 = \mathbf{3}$
$2^4 mod\ 5 = 16\ mod\ 5 = \mathbf{1}$
3
$3^1 mod\ 5 = 3 \bmod 5 = \mathbf{3}$
$3^2 mod\ 5 = 9 \bmod 5 = \mathbf{4}$
$3^3 mod\ 5 = 27 \bmod 5 = \mathbf{2}$
$3^4 mod\ 5 = 81 \bmod 5 = \mathbf{1}$
4
$4^1 mod\ 5 = 4 \bmod 5 = 4$
$4^2 mod\ 5 = 16 \bmod 5 = 1$
$4^3 mod\ 5 = 64 \bmod 5 = 4$
$4^4 mod\ 5 = 256 \bmod 5 = 1$

2. Користувач A вибирає випадкове ціле число $x < p$, обчислює $x_A = g^x \, mod \, p$ та відправляє його користувачеві B.
3. Користувач B вибирає випадкове ціле число $y < p$, обчислює $y_B = g^y \, mod \, p$ та відправляє його користувачеві A.
4. Користувач A обчислює закритий ключ за формулою $k_A = y_B^x \, mod \, p$.
5. Користувач B обчислює закритий ключ за формулою $k_B = x_A^y \, mod \, p$.

Ці дві формули обчислення дають однакові результати. Відкритими параметрами є: p, g, x_A та y_B. Закриті параметри: x, y.

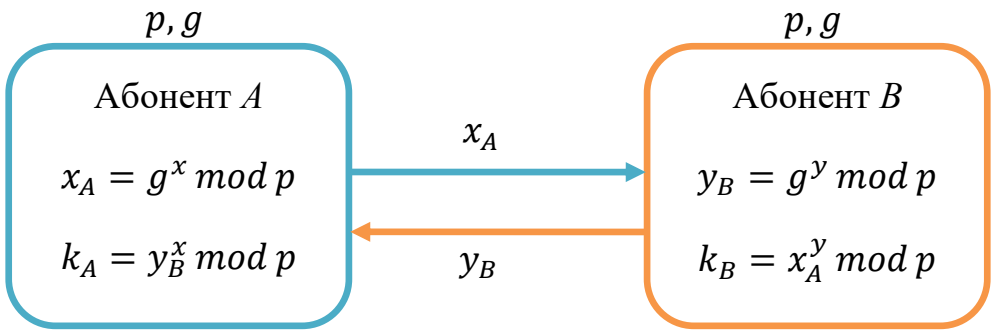


Рис. 6.2. Схема обміну ключами Діффі-Хелмана

Приклад 6.3:

1. Нехай p =11, g =2.
2. x = 4, обчислимо $x_A = 2^4 \, mod \, 11 = 16 \, mod \, 11 = 5$.
3. y = 6, обчислимо $y_B = 2^6 \, mod \, 11 = 64 \, mod \, 11 = 9$.
4. $k_A = 9^4 \, mod \, 11 = (9^2)^2 \, mod \, 11 = 4^2 \, mod \, 11 = 16 \, mod \, 11 = 5$.
5. $k_B = 5^6 \, mod \, 11 = (5^3)^2 \, mod \, 11 = 4^2 \, mod \, 11 = 16 \, mod \, 11 = 5$.

Секретний ключ, обчислений обома сторонами – 5.

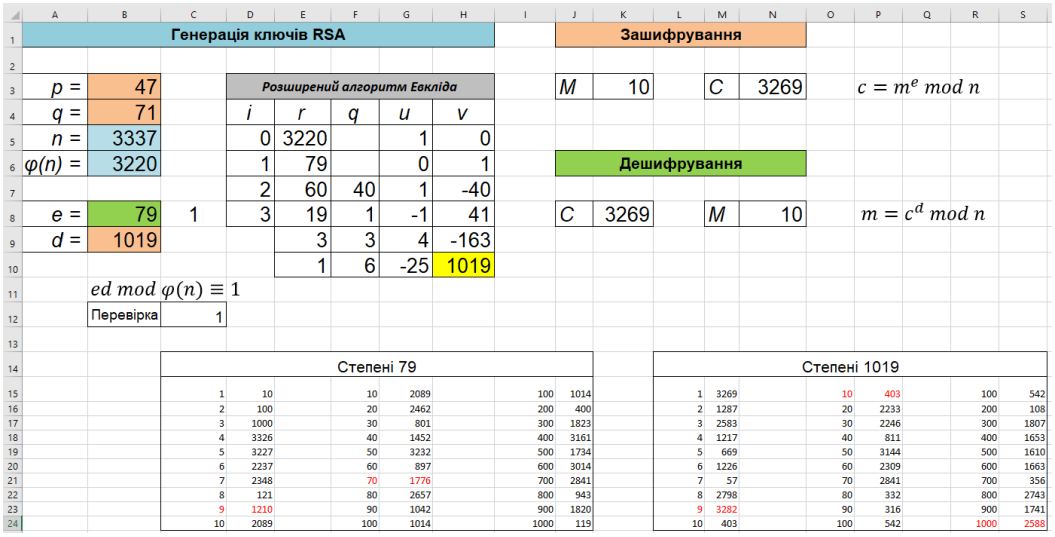
Завдання до лабораторної роботи

Завдання 1

Реалізувати в середовищі MS Excel або на будь-якій мові програмування роботу асиметричного криптографічного алгоритму RSA. Кроки алгоритму шифрування зі скріншотами описати у звіті.

Значення параметрів p і q та відкритого ключа e визначається згідно варіанту. Зашифрувати число, що відповідає кількості літер у вашому прізвищі та дешифрувати отриманий шифротекст. Зразок виконання завдання наведено на

рисунку нижче (рис. 6.3).



Генерація ключів RSA

p =	47
q =	71
n =	3337
$\varphi(n)$ =	3220
e =	79
d =	1019

$ed \bmod \varphi(n) \equiv 1$

Перевірка 1

Розширений алгоритм Евкліда

i	r	q	u	v
0	3220		1	0
1	79		0	1
2	60	40	1	-40
3	19	1	-1	41
	3	3	4	-163
	1	6	-25	1019

Зашифрування

M	10	C	3269

$c = m^e \bmod n$

Дешифрування

C	3269	M	10

$m = c^d \bmod n$

Степені 79

1	10	10	2089	100	1014
2	100	20	2462	200	400
3	1000	30	801	300	1823
4	3326	40	1452	400	3161
5	3227	50	3232	500	1734
6	2237	60	897	600	3014
7	2348	70	1776	700	2841
8	121	80	2657	800	943
9	1210	90	1042	900	1820
10	2089	100	1014	1000	119

Степені 1019

1	3269	10	403	100	542
2	1287	20	2233	200	108
3	2583	30	2246	300	1807
4	1217	40	811	400	1653
5	669	50	3144	500	1610
6	1226	60	2309	600	1663
7	57	70	2841	700	356
8	2798	80	332	800	2743
9	3282	90	316	900	1741
10	403	100	542	1000	2588

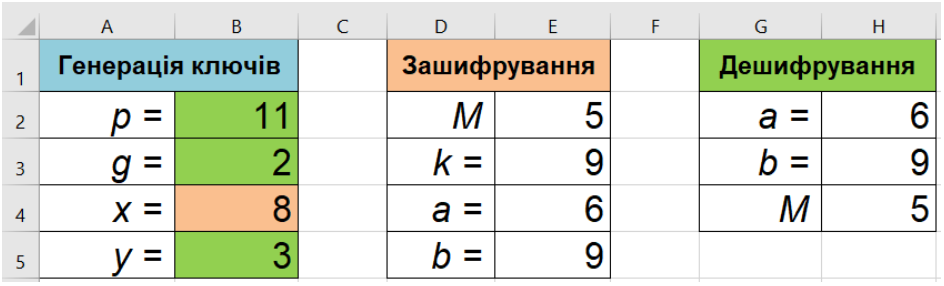
Рис. 6.3. Реалізація шифру RSA в MS Excel

Варіант №	p	q	e
1.	41	43	23
2.	53	61	11
3.	29	37	31
4.	37	53	17
5.	67	79	23
6.	19	41	29
7.	23	83	21
8.	31	61	13
9.	17	97	13
10.	59	83	19
11.	103	107	11
12.	73	89	23
13.	53	61	23
14.	29	59	25
15.	37	47	19

Завдання 2

Реалізувати в середовищі MS Excel або на будь-якій мові програмування роботу асиметричного криптографічного алгоритму Ель-Гамаля. Кроки алгоритму шифрування зі скріншотами описати у звіті.

Значення параметрів p і g та закритого ключа x визначається згідно варіанту. Зашифрувати число, що відповідає кількості літер у вашому прізвищі та дешифрувати отриманий шифротекст. Зразок виконання завдання наведено на рисунку нижче (рис. 6.4).



	A	B	C	D	E	F	G	H
1	Генерація ключів			Зашифрування			Дешифрування	
2	p =	11		M	5		a =	6
3	g =	2		k =	9		b =	9
4	x =	8		a =	6		M	5
5	y =	3		b =	9			

Рис. 6.4. Реалізація шифру Ель-Гамаля в MS Excel

Варіант №	***p***	***g***	***x***	***k***
1.	13	6	10	4
2.	17	3	7	11
3.	11	7	5	6
4.	19	10	6	5
5.	23	5	16	11
6.	13	6	7	10
7.	11	7	9	7
8.	29	8	13	10
9.	23	5	16	9
10.	17	8	13	12
11.	19	13	11	9
12.	29	12	10	8
13.	17	6	8	15
14.	23	11	7	8
15.	13	7	10	11

Завдання 3

Реалізувати в середовищі MS Excel або на будь-якій мові програмування алгоритм обміну ключами Діффі-Хеллмана. Кроки алгоритму шифрування зі скріншотами описати у звіті.

Значення параметрів p і g та x і y визначається згідно варіанту. Обчислити значення відкритих параметрів x_A та y_B та здійснити обмін ними між абонентами. Визначити секретні ключі K_A та K_B, якими абоненти не обмінюються. Зразок виконання завдання наведено на рисунку нижче (рис. 6.5).

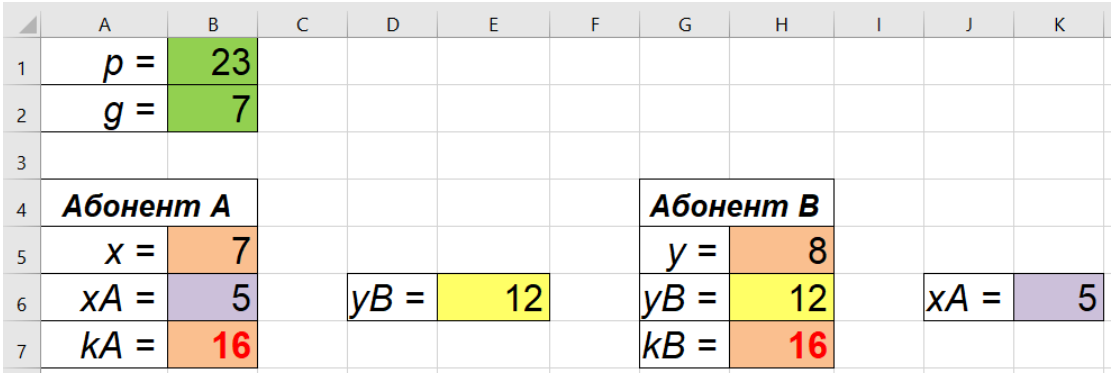


Рис. 6.5. Реалізація алгоритму обміну ключами Діффі-Хелмана

Варіант №	***p***	***g***	Абонент А ***x***	Абонент В ***y***
1.	23	5	7	8
2.	13	6	4	5
3.	11	7	6	8
4.	17	6	4	7
5.	23	7	6	4
6.	19	3	7	5
7.	11	6	8	6
8.	17	3	4	9
9.	13	7	5	8
10.	19	2	8	6
11.	11	8	7	5
12.	23	10	5	4
13.	17	5	6	7
14.	19	10	7	4
15.	13	11	9	6

ДЗ: Знайти первісні корені 7?

Контрольні запитання:

1. У чому полягає ідея криптосистеми з відкритим ключем?
2. Поняття односторонньої функції.
3. Дайте характеристику алгоритму шифрування RSA.
4. На основі яких операцій відбувається створення закритого ключа із відкритого у RSA?
5. На чому заснована складність зламу алгоритму RSA?
6. Опишіть алгоритм шифрування Ель-Гамаля.
7. Опишіть алгоритм обміну ключами Діффі-Хелмана.
8. На чому базується криптостійкість протоколу обміну ключами Діффі-Хелмана?

ТЕМА № 7. ЦИФРОВИЙ ПІДПИС

Мета роботи: набути уміння із створення та перевірки підпису повідомлення за допомогою алгоритмів RSA та Ель-Гамаля; навчитись створювати власні ключі криптографічного захисту даних, обмінюватися ними з іншими користувачами, шифрувати та підписувати повідомлення за допомогою системи GNU Privacy Guard.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням GNU Privacy Guard, інструкції до лабораторної роботи, текстові повідомлення для шифрування та підписування згідно варіанту.

Теоретичні відомості

ПОНЯТТЯ ЦИФРОВОГО ПІДПИСУ

Із широким розповсюдженням у сучасному світі електронних форм документів, у тому числі і конфіденційних, та засобів їхньої обробки, особливо актуальним є питання автентифікації, ідентифікації та неспростовності електронної документації. Для захисту від підробки, перевірки цілісності даних та достовірності джерела повідомлення використовують цифровий підпис.

Електронний підпис – електронні дані, які додаються підписувачем до інших електронних даних або логічно з ними пов’язуються і використовуються ним як підпис.

(Електронний) цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача.

Існує декілька алгоритмів побудови цифрового підпису (ЦП). Найбільш ефективним та найпоширенішим у застосуванні на даний момент є алгоритм ЦП на основі асиметричних криптосистем з використанням хеш-функцій. *Хеш-функція* являє собою функцію, математичну або іншу, що отримує на вхід рядок змінної довжини і перетворює його в рядок фіксованої, зазвичай меншої, довжини. Такі перетворення ще називають *функціями згортки*, а їх результати – *хешем*, *хеш-значенням* або *дайджестом повідомлення*.

Хеш-функція *H*, яка використовується у алгоритмі ЦП, призначена для того, щоб стиснути повідомлення *M* довільної довжини до двійкового хеш-значення *h(M)* фіксованої довжини.

Основні властивості криптографічної хеш-функції:

1) *Детермінованість* – для однакових повідомлень *M* функція має повертати однакові хеш-значення *h*;
2) *Односторонність* – за значенням *h* неможливо відновити *M*;
3) *Наявність лавинного ефекту* – будь-які, навіть незначні, зміни у повідомленні *M* призводять до значних змін у хеш-значенні *h*;
4) *Відсутність колізій (унікальність хеша)* – ймовірність співпадіння хеш-значень двох різних повідомлень повинна бути надзвичайно малою;
5) Висока швидкість роботи.

ЕТАПИ ЦИФРОВОГО ПІДПИСУ

1. *Генерація пари ключів.* За допомогою алгоритму генерації ключів створюється пара ключів – закритий (для створення підпису) та відкритий (для перевірки підпису).
2. *Формування підпису.* Для заданого електронного документа за допомогою деякої хеш-функції обчислюється хеш-значення, після чого воно зашифровується із використанням закритого ключа підписувача. Зашифрований дайджест і є ЦП для даного документу.
3. *Перевірка (верифікація) підпису.* Для отриманого документу одержувач знову обчислює його хеш-значення, після чого за допомогою відкритого ключа підписувача дешифрує ЦП. Якщо хеші рівні – підпис справжній.

Управлянням ключами займаються центри сертифікації ключів (ЦСК), що забезпечують:

- доступ користувача до справжнього відкритого ключа іншого користувача;
- захист ключів від підміни зловмисником;
- організацію відкликання ключа у випадку його компрометації.

Сертифікат відкритого ключа – електронний документ, який засвідчує належність відкритого ключа фізичній або юридичній особі.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ RSA

Для створення підпису повідомлення M спочатку необхідно за допомогою деякої хеш-функції обчислити хеш-значення $h(M)$.

Далі за алгоритмом RSA генеруються ключі (e, n) і (d, n).

ЦП повідомлення $h(M)$ буде мати вигляд: $S = h(M)^d mod\, n.$

Тепер кожний, хто має відкритий ключ підписувача повідомлення, може перевірити дійсність підпису. Для цього необхідно знайти результат хешування прийнятого повідомлення M за допомогою тієї самої хеш-функції $h'(M)$ та порівняти його із $s^e mod\, n = h(M)$. Якщо дайджести рівні – підпис дійсний.

Приклад 7.1:

З використанням алгоритму RSA підписати та перевірити підпис повідомлення M хеш-значення, якого $h(M) = 88$.

Оберемо $p = 17$ і $q = 11$, тоді $n = p \cdot q = 17 \cdot 11 = 187$.

Обчислимо $\varphi(187) = 16 \cdot 10 = 160$.

Виберемо відкритий ключ $e = 7$ та перевіримо виконання умов: $1 < 7 < 160$, НСД(7, 160) =1.

Знайдемо закритий ключ $d = 23$ за розширеним алгоритмом Евкліда з рівняння $7d \equiv 1(mod\ 160)$.

Обчислимо підпис за допомогою закритого ключа підписувача:

$$s = h(M)^d mod\, n = 88^{23} mod\, 187 = 11.$$

Для перевірки підпису повідомлення M одержувачу потрібно знову обчислити його хеш-значення $h(M) = 88$ та порівняти із значенням, отриманим за допомогою відкритого ключа підписувача:

$$s^e mod\, n = 11^7 mod\, 187 = 88.$$

В даному випадку будемо вважати, що підпис справжній.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ ЕЛЬ-ГАМАЛЯ

Як правило, спочатку потрібно за допомогою деякої хеш-функції знайти дайджест $h(M)$ для повідомлення M.

Для генерації пари ключів спочатку вибирається просте число p та числа g (первісний корінь за модулем p) й x (закритий ключ). Обидва ці числа повинні бути менше p. Після чого обчислюється $y = g^x \ mod \ p$ (відкрити ключ).

Виберемо сесійний ключ – випадкове число k, таке що $1 < k < p - 1$ та обчислимо $r = g^k mod \ p$. Після чого обчислимо $s = k^{-1}(h(M) - xr) mod \ p - 1$.

Отже, підписом повідомлення M являється пара $(r, \ s)$.

Випадкове значення k повинне зберігатися в секреті і не повинно дублюватися для різних підписів. Для перевірки підпису потрібно використати відкриті параметри $(p, \ g, \ y)$ та переконатися, що $g^{h(M)} \equiv y^r r^s (mod \ p)$.

Приклад 7.2:

З використанням алгоритму Ель-Гамаля підписати та перевірити підпис повідомлення M хеш-значення, якого $h(M) = 14$.

Виберемо $p = 19$ і $g = 10$. Нехай $x = 16$ – закритий ключ. Обчислимо відповідний відкритий ключ $y = g^x mod \ p = 10^{16} mod \ 19 = 4$.

Виберемо $k = 5$ (сесійний ключ), такий що $1 < 5 < 18$.

Визначимо, що $d = 23$ (закритий ключ) за розширеним алгоритмом Евкліда з рівняння $7d \equiv 1 (mod \ 160)$.

Обчислимо підпис:

$$r = 10^5 mod \ 19 = 3;$$

$$s = 5^{-1}(14 - 16 \cdot 3) mod \ 18 == -374 \ mod \ 18 = 4;$$

$5 \cdot ? \ \equiv 1 \ mod \ 18 \rightarrow 5^{-1} mod \ 18 = 11$ *(за розширеним алгоритмом Евкліда)*.

Приймається $(M, 3, \ 4)$. Обчислимо ліву та праву частину рівняння $g^{h(M)} \equiv y^r r^s (mod \ p)$ за модулем p:

$$g^{h(M)} mod \ p = 10^{14} mod \ 19 = 16;$$

$$y^r r^s \ mod \ p \ = 4^3 \cdot 3^4 mod \ 19 = 16.$$

Можна зробити висновок, що підпис дійсний.

РОБОТА ІЗ СИСТЕМОЮ GNU PRIVACY GUARD ІЗ ВИКОРИСТАННЯМ ОБОЛОНКИ KLEOPATRA

GNU Privacy Guard, **GnuPG** – вільно поширюване програмне забезпечення, що використовує криптографію з відкритим ключем. Перша версія проекту, створена Вернером Кохом (Werner Koch) та профінансована німецьким урядом, вийшла в світ у 1999 році під ліцензією GNU General Public. Функції GnuPG дозволяють шифрувати та підписувати повідомлення за допомогою цифрового підпису, а також керувати списками відкритих ключів респондентів.

Звичним інтерфейсом для GnuPG є командний рядок, проте на сьогоднішній день існують різні зовнішні оболонки, які роблять доступною функціональність цієї програми через графічний інтерфейс користувача, наприклад *Kleopatra* для Windows або *GNU Privacy Assistant* (GPA) для Linux.

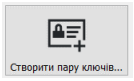
В GnuPG використовуються різні криптографічні алгоритми: симетричні шифри, шифрування з відкритим ключем і змішані (гібридні) алгоритми.

Гібридна (змішана, комбінована) криптосистема – це криптосистема, в якій розподіл ключів здійснюється за допомогою асиметричних криптоалгоритмів, а процес шифрування даних – за допомогою симетричних. Тобто симетричний ключ використовується для шифрування даних, а асиметричний для шифрування самого симетричного ключа. Гібридні криптосистеми поєднують в собі зручність розподілу секретних ключів та високу швидкість шифрування.

Як правило, при гібридному шифруванні створюється *одноразовий секретний сеансовий ключ* – це псевдовипадкове число, яке генерується на основі випадкових рухів миші, натискань клавіш клавіатури тощо. Такий ключ використовується лише один раз для шифрування повідомлення з використанням деякого надійного та швидкого симетричного алгоритму. Сеансовий ключ зашифровується відкритим ключем одержувача та додається до шифротексту. Під час дешифрування процедури виконуються у зворотному порядку.

Створення пари ключів

При першому запуску *Kleopatra* (рис. 7.1) потрібно створити власну зв'язку ключів. Для цього необхідно виконати наступні дії:

1) натиснути кнопку 

або скористатися меню *Файл⇒Створити пару ключів*;

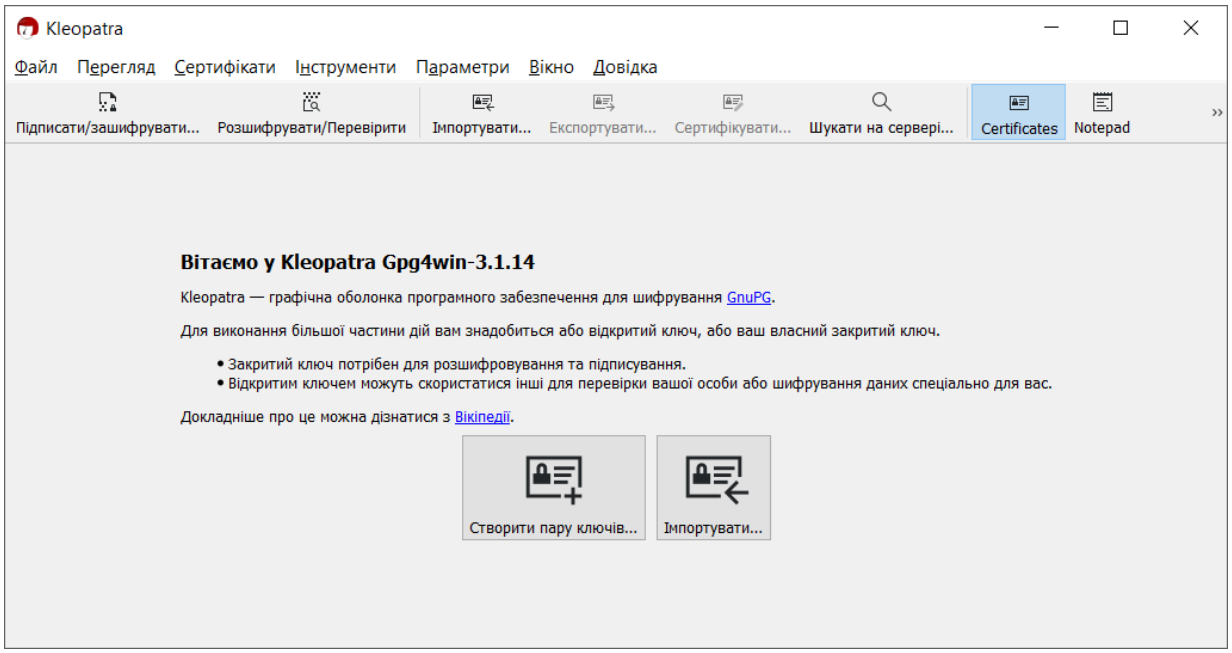
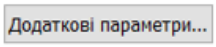


Рис. 7.1. Стартове вікно оболонки Kleopatra

2) у вікні *Майстра створення ключів* (рис. 7.2) потрібно ввести відомості про себе у відповідні поля (ім'я, електронну адресу); кнопка 

дозволяє вибрати тип ключа його довжину, строк дії тощо.

Основною особливістю GnuPG є система ключів. В GnuPG користувач створює декілька ключів, причому кожен служить для окремої дії (і використовує різні алгоритми). Один із ключів, що створюється першим, є *головним ключем*, решта ключів йому підпорядковані – це *підключі* (субключі).

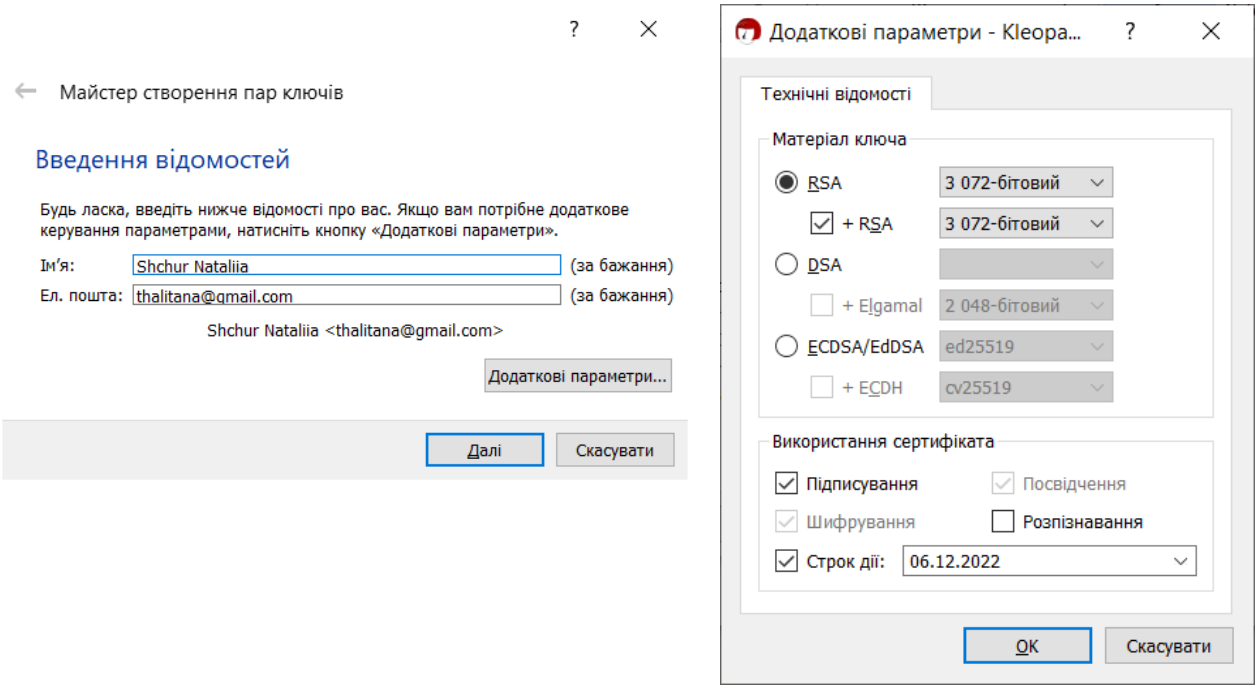


Рис. 7.2. Створення пари ключів за допомогою майстра

3) у наступному вікні необхідно натиснути *Створити* та ввести пароль для захисту нового ключа (рис. 7.3);

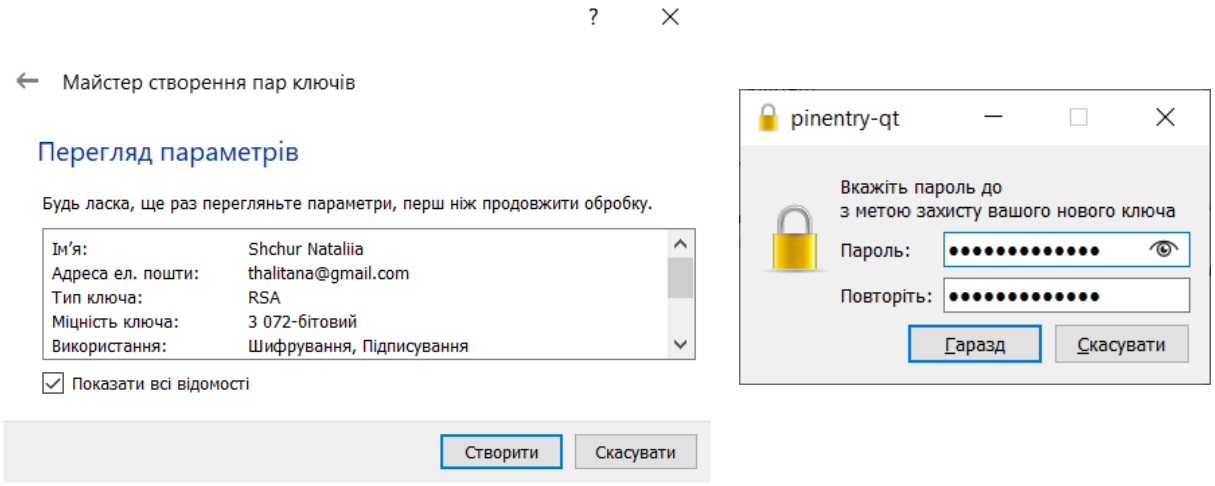


Рис. 7.3. Введення паролю для захисту нового ключа

4) у наступному вікні майстер має повідомити про успішне створення ключів (рис. 7.4), після чого потрібно натиснути кнопку *Завершити*.

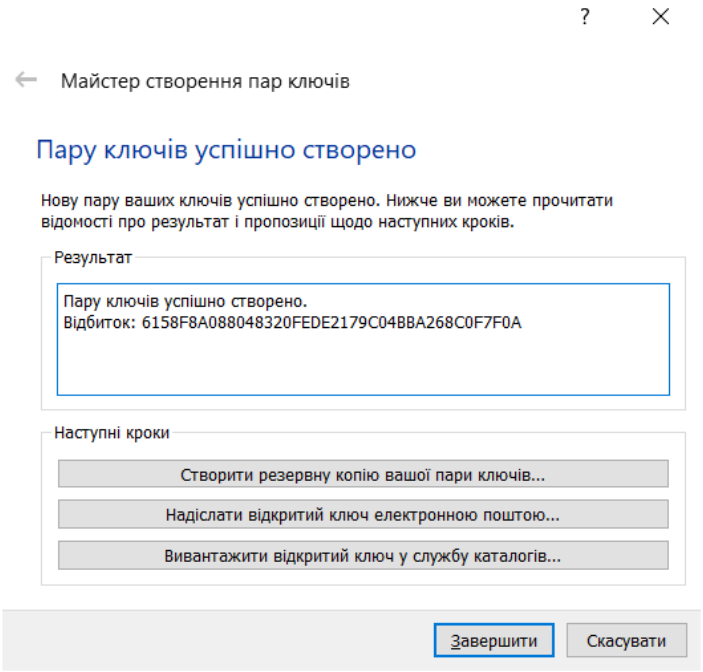


Рис. 7.4. Повідомлення про успішне створення ключів

Усі функції управління ключами здійснюються у вікні ***Kleopatra*** (рис. 7.5), в якому висвітлюються всі ключі, створені користувачем для власного користування, а також усі імпортовані публічні ключі його кореспондентів.

Ключі зберігаються у зашифрованій формі у вигляді двох файлів, які називаються *зв'язками ключів* (keyrings). Ці файли записуються у папках на диску відповідно до поточних налаштувань.

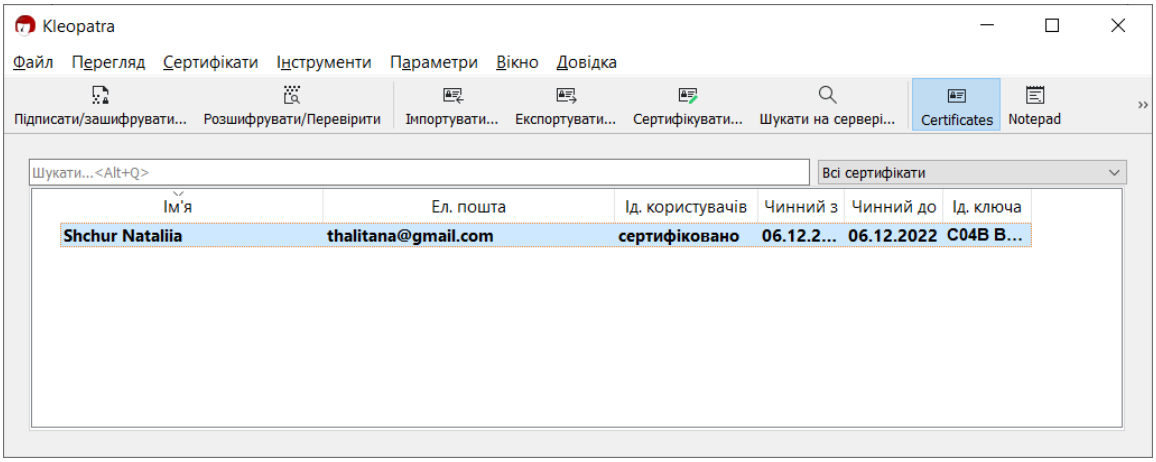
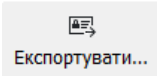


Рис. 7.5. Список наявних ключів у вікні оболонки Kleopatra

Експорт ключів

До початку обміну повідомленнями з іншими користувачами GPG варто обмінятись з ними публічними ключами.

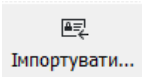
Для експорту ключа потрібно:

1) у вікні ***Kleopatra*** натиснути кнопку 

або у контекстному меню електронного ключа вибрати пункт *Експортувати*, або використати меню *Файл⇒ Експортувати*;
2) обрати папку для збереження ключа, ввести його ім’я та натиснути *Зберегти*.

Імпорт ключів

Імпортувати відкриті ключі інших користувачів можна, виконавши такі дії:

3) у вікні ***Kleopatra*** натиснути кнопку 

або у контекстному меню електронного ключа вибрати пункт *Імпортувати*, або використати меню *Файл⇒ Імпортувати*;
4) обрати ключ на диску, ввести його ім’я та натиснути *Відкрити*;
5) також варто погодитися із перевіркою сертифіката ключа (рис. 7.5).

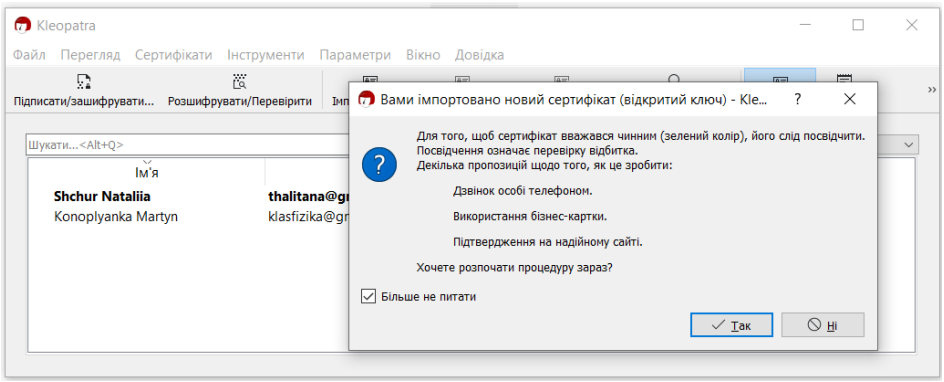


Рис. 7.5. Перевірка сертифіката ключа, що імпортується

Шифрування та (або) підписування файлів

Для шифрування та (або) підписування файлу необхідно натиснути кнопку 

або використати меню *Файл⇒ Підписати/зашифрувати*;

Відкриється діалогове вікно *Підписати/зашифрувати файли* (рис. 7.6), у якому потрібно обрати необхідну дію та обрати відкриті ключі одержувача(-ів) повідомлення, натиснувши по піктограмі .

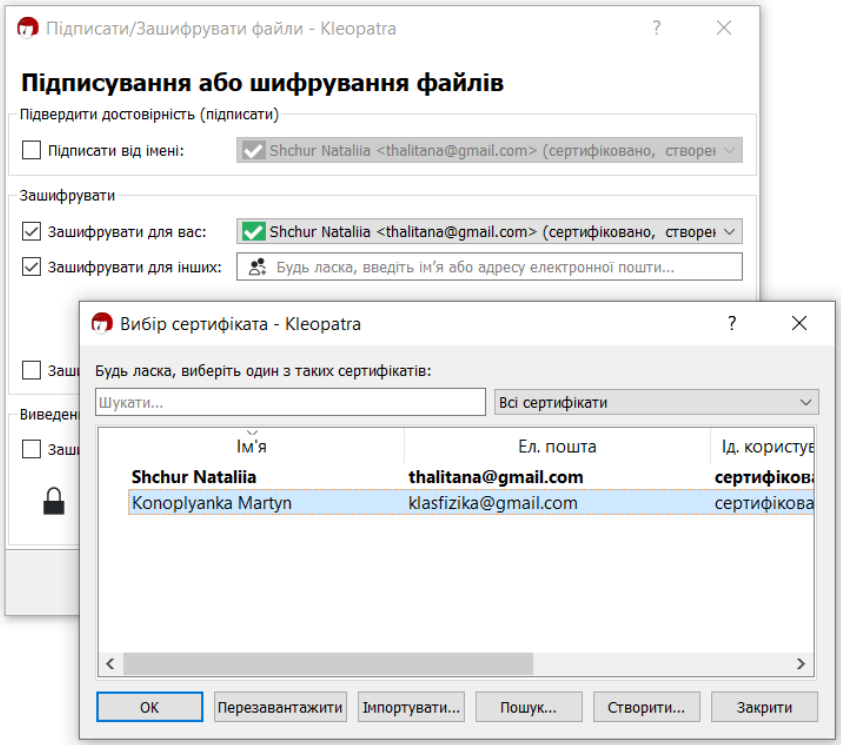
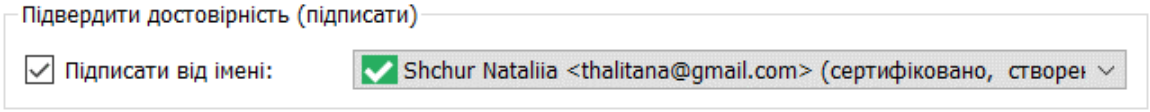


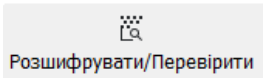
Рис. 7.6. Діалогове вікно Підписати/зашифрувати файли

Підписати файл за допомогою свого відкритого ключа дозволяє опція:



Також існує можливість виконати дві описані вище операції одночасно.

Розшифрування та (або) перевірка підпису файлів

Для розшифрування/перевірки цифрових підписів файлів використовуються кнопка 

та пункт меню *Файл⇒ Розшифрувати/Перевірити.*

Під час розшифрування на екрані з'явиться вікно перевірки пароля. Файл буде розшифрований після введення правильного пароля за умови, що його було зашифровано з використанням відкритого ключа отримувача (рис. 7.7). Очевидно також, що розшифрування файлу можливе тільки за умов наявності у середовищі вікна ***Kleopatra*** закритого ключа отримувача. Розшифрованому файлу автоматично присвоюється назва файлу-оригіналу (файлу, який було зашифровано).

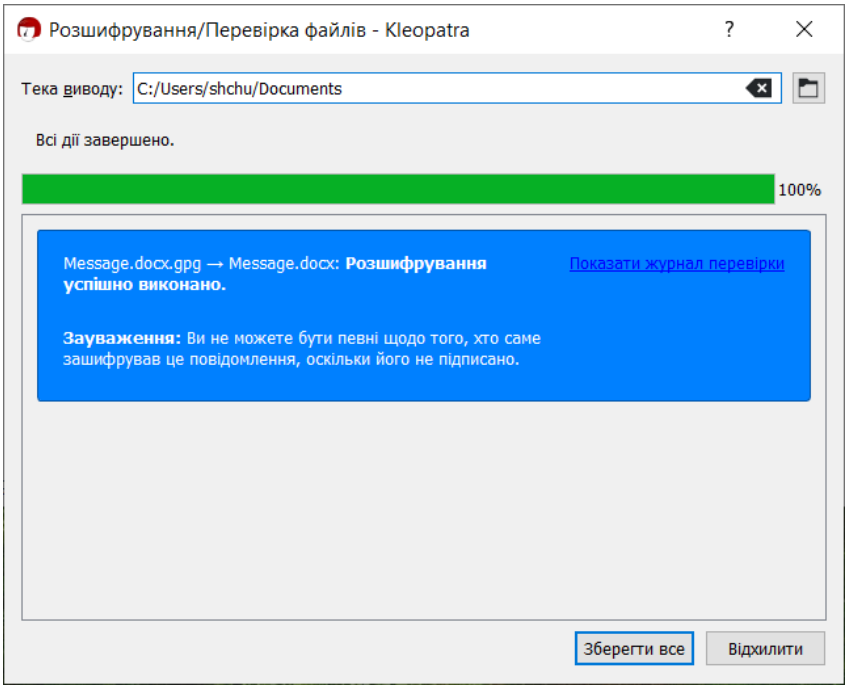


Рис. 7.7. Вікно розшифрування файлу

Якщо файл має підпис, на екрані з'являється вікно з повідомленням, яке містить назву файлу, відомості про особу, яка підписала файл, дату і час накладання підпису та позначку, чи залишається підпис дійсним (рис. 7.8.).

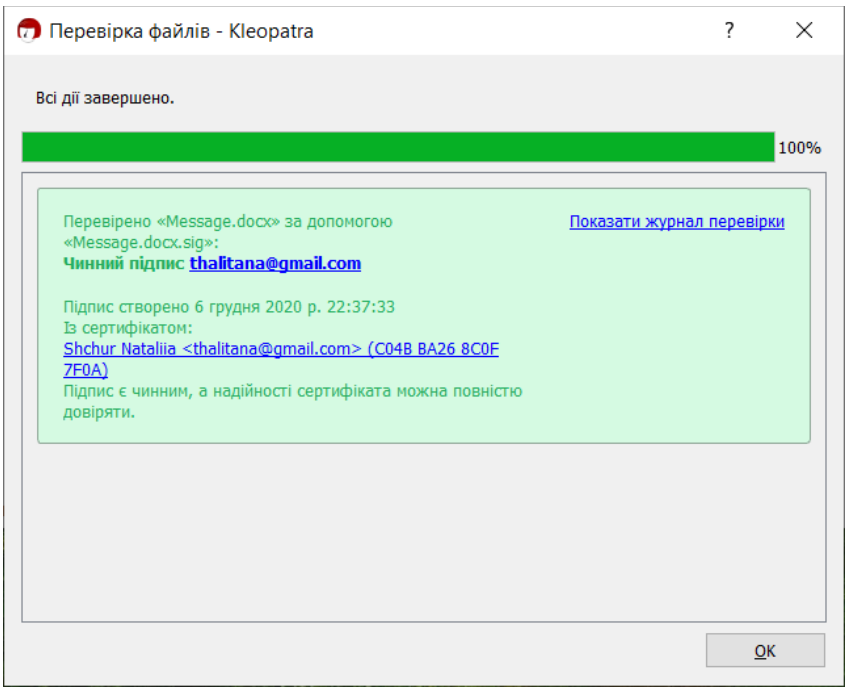


Рис. 7.8. Вікно перевірки підпису

Доступ до функцій GPG

Для забезпечення зручного виконання операцій шифрування, підписування, дешифрування, перевірки підпису тощо, у контекстному меню файлу (рис. 7.8) можна обрати *Sign and encrypt* або *More GpgEX option*.

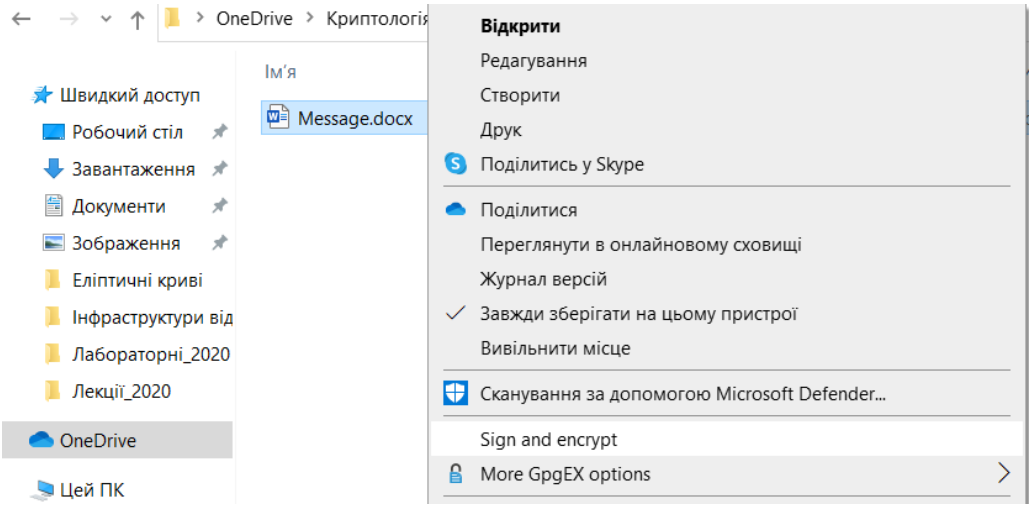


Рис. 7.9. Вибір у контекстному меню документа команд GPG

Завдання до лабораторної роботи

Завдання 1

Виконати створення та перевірку ЦП повідомлення згідно варіанту (визначається номером студента у журналі: непарний – 1 варіант, парний – 2 варіант). Усі кроки алгоритму описати у звіті.

1. З використанням алгоритму RSA створіть та перевірте підпис повідомлення, якщо його хеш $h(M) = 7$, а параметри p=13 та q=17. Самостійно оберіть відкритий ключ e та обчисліть закритий ключ d.

   За алгоритмом Ель-Гамаля виконайте формування та перевірку підпису повідомлення, якщо його хеш $h(M) = 8$, а параметри p=23 та g=5. Оберіть закритий ключ x, сесійний ключ k та обчисліть відкритий ключ y.
2. З використанням алгоритму RSA створіть та перевірте підпис повідомлення, якщо його хеш $h(M) = 6$, а параметри p=11 та q=13. Самостійно оберіть відкритий ключ e та обчисліть закритий ключ d.

За алгоритмом Ель-Гамаля виконайте формування та перевірку підпису повідомлення, якщо його хеш $h(M) = 7$, а параметри p=19 та g=3. Оберіть закритий ключ x, сесійний ключ k та обчисліть відкритий ключ y.

Завдання 2

Виконати завдання у системі GPG та додати до звіту скріншоти вікна GPG на кожному кроці: створення ключів, експортування/імпортування ключів, шифрування/підписування, дешифрування/перевірки підпису, а також скріншот дешифрованого текстового повідомлення від викладача.

2.1. Створити ключі у діалоговому вікні ***Kleopatra*** на основі алгоритму RSA, довжиною 3072 біт. Заповнити поля *Ім'я* та *Елек. пошта* (латинськими літерами).

2.2. Експортувати свій публічний ключ у свою робочу папку. Відповідний файл повинен мати назву за шаблоном, наприклад *Shchur Nataliia_0x8C0F7F0A_public.asc*.

2.3. Відправити свій публічний ключ викладачеві thalitana@ztu.edu.ua.

2.4. Імпортувати публічний ключ викладача до середовища ***Kleopatra***.

2.5. За допомогою текстового редактора створити файл, вказати у ньому своє прізвище, ім'я, по батькові. Присвоїти файлу назву *Enc_N.docx*, де *N* – номер студента за списком групи, впорядкованим за алфавітом (наприклад, *Enc _12.docx*).

2.6. Із використанням відкритого ключа викладача зашифрувати *Enc_N.docx* за допомогою GPG. Схема зашифрування повідомлення із використанням GnuPG представлена на рис. 7.10.

2.7. За допомогою текстового редактора створити файл, вказати у ньому свій варіант, курс, групу. Присвоїти файлу назву *Enc_Sign_N.docx*, де *N* – номер студента за списком групи, впорядкованим за алфавітом (наприклад, *Enc_Sign _12.docx*).

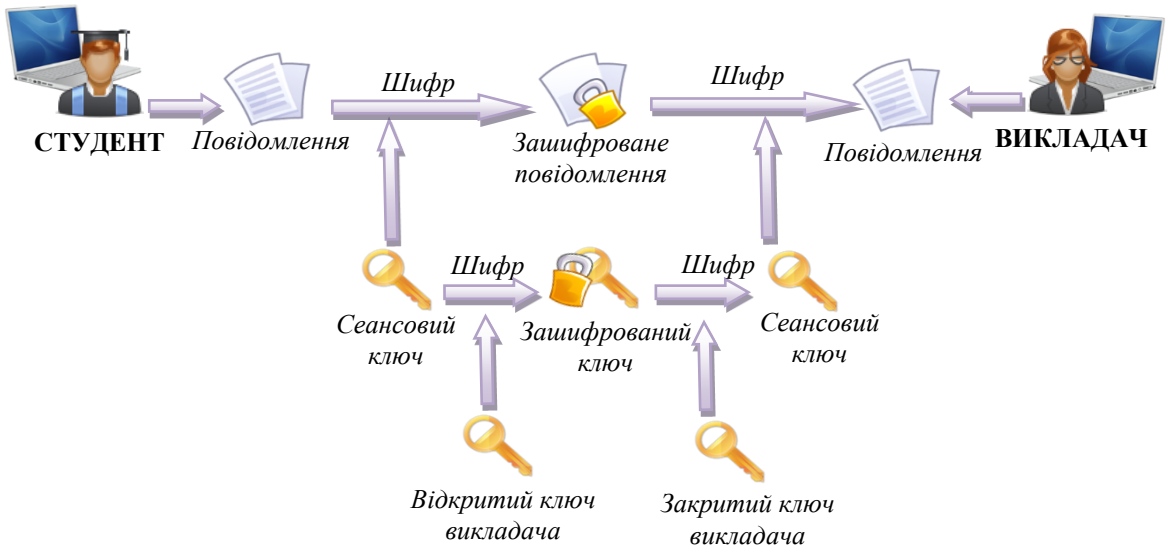


Рис. 7.10. Схема зашифрування повідомлення із використанням GnuPG

2.8. Із використанням свого ключа підписати *Enc_Sign_N.docx* та зашифрувати за допомогою ключа викладача. Схема алгоритму створення та перевірки підпису з використанням GnuPG представлена на рис. 7.11.

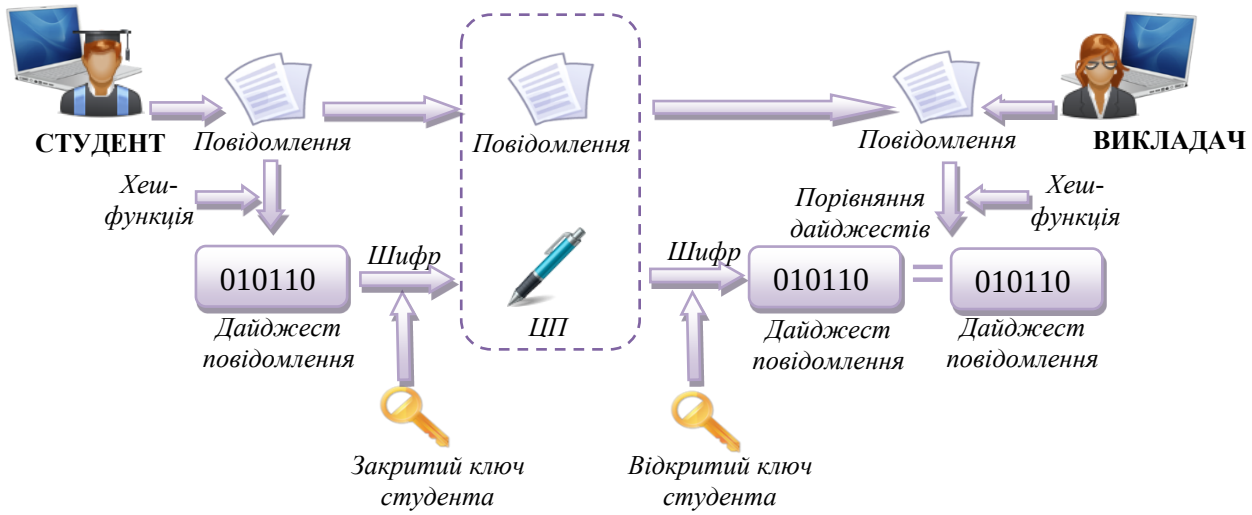


Рис. 7.11. Схема створення та перевірки підпису з використанням GnuPG

2.9. Відправити два файли викладачеві: зашифрований *Enc_N.docx* та підписаний/зашифрований *Enc_Sign_N.docx*.

2.10. Отримати від викладача зашифроване повідомлення, підписане його цифровим підписом.

2.11. Розшифрувати повідомлення викладача та перевірити дійсність його підпису у системі GPG.

Контрольні питання:

1. Для чого потрібен цифровий підпис?
2. Дайте визначення поняттям «хешування», «хеш-функція».
3. Опишіть схему створення і перевірки ЦП.
4. Який порядок використання відкритого та закритого ключів при створенні і перевірці ЦП?
5. Які схеми цифрового підпису існують?
6. Як здійснюється підпис RSA? Яка відмінність підпису RSA від шифру RSA?
7. Як здійснюється підпис Ель-Гамаля?
8. Як здійснюється перевірка на дійсність підпису Ель-Гамаля?

ТЕМА № 8. КРИПТОГРАФІЧНІ ПЕРЕТВОРЕННЯ В ГРУПАХ ТОЧОК ЕЛІПТИЧНИХ КРИВИХ

Мета роботи: ознайомитися з алгоритмами криптографічних перетворень на еліптичних кривих, здійснивши формування спільного ключа між двома абонентами за алгоритмом Діффі-Хеллмана на еліптичних кривих.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням MS Excel та доступом до мережі Інтернет.

Теоретичні відомості

КРИПТОГРАФІЯ НА ЕЛІПТИЧНИХ КРИВИХ

Криптографія на еліптичних кривих (elliptic curve cryptography, ECC) вивчає асиметричні криптосистеми, засновані на еліптичних кривих над скінченими полями. Їх безпека, як правило, базується на складності розв'язання задачі дискретного логарифмування в групі точок еліптичної кривої над скінченним полем. Використання еліптичних кривих у криптографії було незалежно запропоновано Нілом Кобліцом (Neal Koblitz) та Віктором Міллером (Victor Miller) у 1985 році. З 1998 року використання еліптичних кривих для вирішення криптографічних завдань було закріплено в стандартах США ANSI X9.62 і FIPS 186-2 (FIPS 186-3 з 2009 року). В Україні на рівні національного стандарту (ДСТУ 4145-2002) прийнято алгоритм цифрового підпису, що ґрунтується на еліптичних кривих.

Основною перевагою криптосистем на еліптичних кривих у порівнянні із звичайними асиметричними алгоритмами є те, шо вони забезпечують еквівалентний захист за меншої довжини ключа (табл. 8.1).

Таблиця. 8.1. Порівняння звичайних асиметричних алгоритмів та криптосистем на еліптичних кривих

Ступінь захисту (на кожен біт ключа)	*Мінімальна довжина ключа (в бітах)*	
	RSA/DSA/DH	**ECC**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Розглянемо рівняння еліптичної кривої у спрощеному вигляді (рівняння Вейєрштрасса):

$$y^2 = x^3 + ax + b \tag{8.1}$$

Залежно від значень параметрів a і b еліптичні криві можуть приймати на площині різні форми. Так як $y = \pm\sqrt{x^3 + ax + b}$, то графік кривої симетричний відносно *Ox*.

Дискримінант рівняння: $D = \left(\frac{a}{3}\right)^3 + \left(\frac{b}{2}\right)^2$.

- $D < 0$ – три різних дійсних корені (рис. 8.1, графік 1);
- $D = 0$ – три дійсних корені, два з яких однакові (рис. 8.1, графік 2 – сингулярна крива, такі криві виключають з розгляду);
- $D > 0$ – один дійсний корінь та два комплексних (рис. 8.1, графік 3).

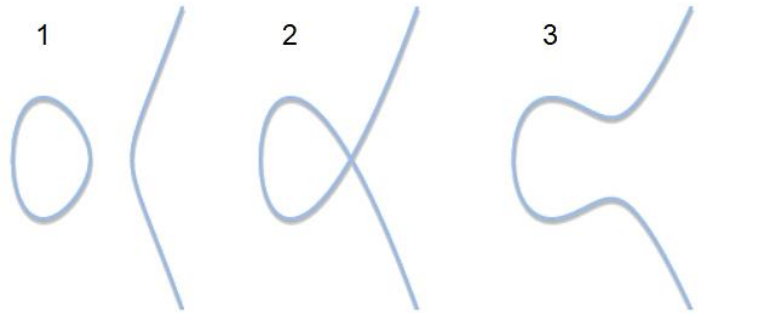


Рис. 8.1. Варіанти еліптичних кривих при $D < 0$, $D = 0$ та $D > 0$

У реальних криптосистемах використовуються еліптичні криві над скінченним полем p, що описуються рівнянням:

$$y^2 \equiv x^3 + ax + b \ (mod\ p), \tag{8.2}$$

де $(x,\ y)$ – точки еліптичної кривої,

$a,\ b$ – параметри кривої,

p – просте число ($p \neq 2, p \neq 3$).

При цьому параметри кривої a та b мають задовольняти умову:

$$4a^3 + 27b^2 \neq 0 \ (\mathrm{mod}\ p).$$

Позначимо через $E_p(a,\ b)$ множину точок еліптичної кривої. У множину точок еліптичної кривої також включається нескінченно віддалена точка O.

Точка належить еліптичній кривій, якщо пара чисел (x, y) задовольняє рівнянню (8.2).

Кількість точок кривої називається *порядком кривої.*

Приклад 8.1:

Множина точок $E_5(2, 1)$ еліптичної кривої $y^2 \equiv x^3 + 2x + 1 \ (mod\ 5)$, складається з 6 точок, а також точки O. Порядок кривої – 7. На рис.8.2 зображено усі точки, що задовольняють рівнянню кривої.

solve	$y^2 \equiv x^3 + 2x + 1 \pmod 5$

Solutions in the least residue system:

$x \equiv 0, y \equiv 1 \pmod 5$

$x \equiv 0, y \equiv 4 \pmod 5$

$x \equiv 1, y \equiv 2 \pmod 5$

$x \equiv 1, y \equiv 3 \pmod 5$

$x \equiv 3, y \equiv 2 \pmod 5$

$x \equiv 3, y \equiv 3 \pmod 5$

Рис. 8.2. Точки, що належать еліптичній кривій $y^2 \equiv x^3 + 2x + 1 \ (mod\ 5)$

ОПЕРАЦІЇ НАД ТОЧКАМИ ЕЛІПТИЧНИХ КРИВИХ

Оберненою точкою до $P(x, y)$ називають точку еліптичної кривої, що симетрична відносно осі Ox та позначають $-P(x, -y)$. Варто зауважити, що $-P$ має належати $E_p(a, b)$.

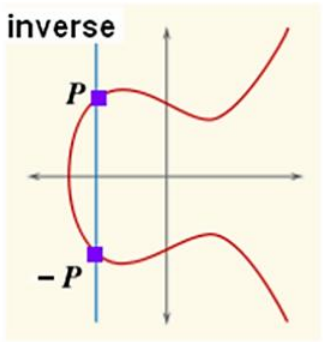


Рис. 8.3. Обернена точка еліптичної кривої

Приклад 8.2:

Якщо $P(3,\ 2)$ – точка еліптичної кривої $y^2 \equiv x^3 + 2x + 1\ (mod\ 5)$, то точка $-P\ (3,\ -2)$. Проте $-2\ mod\ 5 = 3$, тому $-P\ (3,\ 3)$.

Додавання точок. Візьмемо дві різні точки $P(x_1,\ y_1)$ та $Q(x_2,\ y_2)$, які належать E_p і проведемо через них пряму. Ця пряма обов'язково перетне криву в третій точці R. Проведемо через точку R вертикальну пряму до перетину з кривою у точці $-R = P + Q$. Отже, сумою двох точок P та Q буде точка, обернена до третьої точки перетину еліптичної кривої і прямої, що проходить через задані точки.

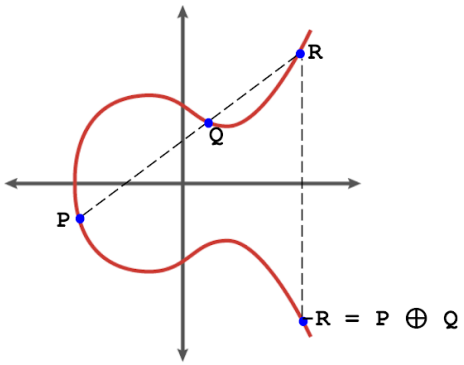


Рис. 8.4. Додавання точок еліптичної кривої

Подвоєння точки. Якщо дві точки $P(x_1,\ y_1)$ та $Q(x_2,\ y_2)$ співпадають, то $P + Q = P + P$, що рівнозначно подвоєнню точки $2P = -R$. При $P = Q$ січна перетворюється на дотичну, тому точка $2P$ є оберненою до точки R.

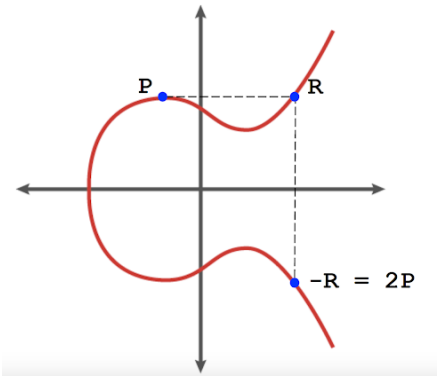


Рис. 8.5. Подвоєння точки еліптичної кривої

Координати $-R(x_3,\ y_3)$ визначаються за формулами, де λ – кутовий коефіцієнт січної, що проведена через точки $P(x_1,\ y_1)$ та $Q(x_2,\ y_2)$.

Додавання точок (якщо $P \neq Q$)	***Подвоєння точки (якщо $P = Q$)***
$x_3 = \lambda^2 - x_1 - x_2 (mod\ p);$	$x_3 = \lambda^2 - 2x_1\ (mod\ p);$
$y_3 = \lambda(x_1 - x_3) - y_1 (mod\ p);$	$y_3 = \lambda(x_1 - x_3) - y_1 (mod\ p);$
$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (mod\ p).$	$\lambda = \frac{3x_1^2 + a}{2y_1} (mod\ p).$

Приклад 8.3:

Рівняння еліптичної кривої має вигляд:

$$y^2 \equiv x^3 + x + 1\ (mod\ 23), \tag{8.3}$$

Потрібно перевірити чи точки $P(3,\ 10)$ та $Q(9,\ 7)$ належать кривій та знайти $P + Q$.

Підставимо значення $P(3,\ 10)$ та $Q(9,\ 7)$ у рівняння еліптичної кривої (8.3) та переконаємося, що точки належать кривій:

$$10^2 \equiv 3^3 + 3 + 1\ (mod\ 23) \rightarrow 100\ mod\ 23 \equiv 31 (mod\ 23);$$

$$7^2 \equiv 9^3 + 9 + 1\ (mod\ 23) \rightarrow 49\ mod\ 23 \equiv 739 (mod\ 23).$$

Виконаємо додавання точок $P(3,\ 10)$ та $Q(9,\ 7)$:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (mod\ p) = \frac{7-10}{9-3} (mod\ 23) = -\frac{3}{6} (mod\ 23) = -\frac{1}{2} (mod\ 23) =$$

$$= \frac{22}{2} (mod\ 23) = 11.$$

Знаходимо:

$$x_3 = \lambda^2 - x_1 - x_2 (mod\ p) = 121 - 3 - 9 (mod\ 23) = 109 (mod\ 23) = 17$$

$$y_3 = \lambda(x_1 - x_3) - y_1 (mod\ p) = 11(3 - 17) - 10 (mod\ 23) = -164 (mod\ 23) =$$

$$= 20.$$

Отже $P + Q = (3,\ 10) + (9,\ 7) = (17,\ 20)$.

Приклад 8.4:

Додати точки $P(12,\ 19)$ та $Q(5,\ 4)$ еліптичної кривої 2.1.

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} (mod\ p) = \frac{4-19}{5-12} (mod\ 23) = \frac{-15}{-7} (mod\ 23) = \frac{15}{7} (mod\ 23).$$

Якщо записати $15 \cdot \frac{1}{7} (mod\ 23) \rightarrow 5 \cdot 7^{-1} (mod\ 23)$, то потрібно знайти обернений елемент, розв’язавши рівняння:

$7 \cdot Z \equiv 1\ (mod\ 23) \rightarrow Z = 10$ (*за розширеним алгоритмом Евкліда*).

$$\lambda = 15 \cdot 10\ (mod\ 23) = 12.$$

$$x_3 = \lambda^2 - x_1 - x_2 (mod\ p) = 144 - 12 - 5 (mod\ 23) = 127 (mod\ 23) = 12.$$

$$y_3 = \lambda(x_1 - x_3) - y_1 (mod\ p) = 12(12 - 12) - 19 (mod\ 23) = 4 (mod\ 23) = 4.$$

Отже $P + Q = (12,\ 19) + (5,\ 4) = (12,\ 4)$.

Приклад 8.5:

Дано точку $P(5,\ 4)$ еліптичної кривої 2.1. Знайти $2P$ та $3P$.

$$\lambda = \frac{3x_1^2 + a}{2y_1} (mod\ p) = \frac{3 \cdot 25 + 1}{2 \cdot 4} (mod\ 23) = \frac{76}{2 \cdot 4} (mod\ 23) = \frac{19}{2} (mod\ 23).$$

Знайдемо обернений елемент $2^{-1} (mod\ 23)$, розв'язавши рівняння:

$$2 \cdot Z \equiv 1\ (mod\ 23) \rightarrow Z = 12.$$

$$\lambda = 19 \cdot 12\ (mod\ 23) = 21.$$

$$x_3 = \lambda^2 - 2x_1 (mod\ p) = 441 - 10 (mod\ 23) = 431 (mod\ 23) = 17.$$

$$y_3 = \lambda(x_1 - x_3) - y_1 (mod\ p) = 21(5 - 17) - 4 (mod\ 23) = -256 (mod\ 23) =$$

$$= 20.$$

Отже $2P = (17,\ 20)$.

Далі знайдемо суму точок $P + 2P = (5,\ 4) + (17,\ 20)$.

$$\lambda = \frac{20 - 4}{17 - 5} (mod\ 23) = \frac{16}{12} (mod\ 23) = \frac{4}{3} (mod\ 23).$$

Знайдемо обернений елемент, розв'язавши рівняння:

$$3 \cdot Z \equiv 1\ (mod\ 23) \rightarrow Z = 8.$$

$$\lambda = 4 \cdot 8\ (mod\ 23) = 9.$$

$$x_3 = 9^2 - 5 - 17 (mod\ 23) = 81 - 22 (mod\ 23) = 13.$$

$$y_3 = 9(5 - 13) - 4 (mod\ 23) = 9 \cdot (-8) - 4 (mod\ 23) = -76 (mod\ 23) = 16.$$

Отже $3P = (13,\ 16)$.

Множина точок еліптичної кривої $E_p(a,\ b)$ разом із введеною точкою на нескінченності O утворює комутативну групу щодо операції додавання точок. Для цього виконуються усі необхідні властивості:

1) Якщо P і $Q \in E_p(a,\ b)$, то $P + Q \in E_p(a,\ b)$ – замкнутість;

2) $P + Q = Q + P$ – комутативність;

3) $(P + Q) + R = P + (Q + R)$ – асоціативність;

4) $P + (-P) = O$ – обернений елемент;

5) $P + O = O + P = P$ – нейтральний елемент.

Скалярне множення точки на число. Із попередніх операцій додавання точок та подвоєння точки випливає операція скалярного множення точки на число:

$$2P = P + P$$
$$3P = P + P + P$$
$$\dots$$
$$mP = \underbrace{P + P + P + \cdots + P}_{m \text{ разів}}$$

Скалярне множення є аналогом піднесення до степеню в звичайних асиметричних шифрах. Прямою задачею є обчислення $mP = Q$. Зворотна задача полягає у тому, що знаючи точки P та Q, знайти m важко (дискретне логарифмування у групі точок еліптичної кривої).

Точка $G \in E_p(a, b)$ називається **базовою точкою** підгрупи точок еліптичної кривої $E_p(a, b)$, якщо будь-яка точка P цієї підгрупи може бути подана у вигляді $P = mG$, де $m = 1,\ 2, \dots n$, де n – порядок підгрупи.

Для базової точки G має місце рівність $nG = O$.

Приклад 8.6:

Точка $G = (0,\ 1)$ є базовою точкою для групи точок еліптичної кривої $y^2 \equiv x^3 + x + 1\ (mod\ 5)$. Вона генерує усі інші точки підгрупи:

$$\boldsymbol{G} = (0,\ 1) \to \boldsymbol{2G} = (4,\ 2) \to \boldsymbol{3G} = (2,\ 1) \to \boldsymbol{4G} = (3,\ 4) \to \boldsymbol{5G} = (3,\ 1) \to \boldsymbol{6G} = (2,\ 4) \to \boldsymbol{7G} = (4,\ 3) \to \boldsymbol{8G} = (0,\ 4) \to \boldsymbol{9G} = O.$$

АЛГОРИТМ ДІФФІ-ХЕЛМАНА НА ЕЛІПТИЧНИХ КРИВИХ

1. Абоненти А і В спільно обирають просте число p та параметри еліптичної кривої a та b.
2. У групі точок еліптичної кривої $E_p(a,\ b)$ також обирається спільна базова точка $G = (x, y)$, що має дуже великий порядок n.
3. Абонент A обирає $x < n$, обчислює $X_A = xG$ та відправляє його B.
4. Абонент B обирає $y < n$, обчислює $Y_B = yG$ та відправляє його A.

5. Абонент A обчислює закритий ключ за формулою $K_A = xY_B$.
6. Користувач B обчислює закритий ключ за формулою $K_B = yX_A$.

Приклад 8.7:

1. Нехай абоненти обрали параметри еліптичної кривої $p = 23$, $a = -2$, $b = 15$, тобто $y^2 \equiv x^3 - 2x + 15\ (mod\ 23)$.
2. Нехай $G = (4,\ 5)$ – базова очка.
3. Абонент A обирає $x = 3$ та обчислює $X_A = 3G = 2G + G = (13,\ 22)$.
4. Абонент B обирає $y = 7$ обчислимо $Y_B = 7G = 2G + 4G + G = (17,\ 8)$.
5. Абонент A обчислює закритий ключ $K_A = 3Y_B = 2Y_B + Y_B = (15,\ 5)$.
6. Абонент B обчислює закритий ключ $K_B = 7X_A = 2X_A + 4X_A + X_A = (15,\ 5)$.

Секретний ключ, обчислений обома сторонами – $(15,\ 5)$.

Завдання до лабораторної роботи

Знайти спільний секретний ключ K_A та K_B, *що формується обома абонентами за алгоритмом обміну ключами Діффі-Хеллмана на еліптичних кривих.*

Кроки алгоритму з усіма повними обчисленнями описати у звіті.

Значення параметрів еліптичної кривої a і b, p та базова точка G визначається згідно варіанту. Окрім спільного ключа, необхідно обчислити значення відкритих параметрів X_A та Y_B для кожного абоненту.

Варіант №	***p***	***a***	***b***	***G***	Абонент А ***x***	Абонент В ***y***
1.	29	2	1	(5, 7)	4	6
2.	19	1	5	(1, 8)	6	7
3.	23	-2	4	(3, 5)	7	4
4.	29	-3	7	(1, 11)	3	7
5.	23	2	5	(8, 2)	6	4
6.	19	3	5	(4, 10)	5	6
7.	29	-1	2	(10, 8)	7	4
8.	23	5	3	(7, 17)	4	7
9.	31	1	-4	(7, 6)	6	5
10.	23	5	-3	(3, 4)	7	4
11.	29	-2	3	(11, 6)	4	6

12.	19	4	1	(0, 18)	5	8
13.	31	2	5	(5, 4)	3	6
14.	23	-1	3	(13, 5)	4	7
15.	19	-2	5	(0, 9)	3	7

Контрольні запитання:

1. Який загальний вигляд має крива, що використовується в криптографічних системах, заснованих на еліптичних кривих?
2. Дайте визначення порядку групи точок еліптичної кривої.
3. Дайте визначення порядку точки еліптичної кривої.
4. Яка математична проблема забезпечує стійкість криптосистем, побудованих на еліптичних кривих?
5. Які основні операції виконуються над точками еліптичних кривих при їх використанні в криптографічних системах?
6. Опишіть алгоритми додавання та подвоєння точки.
7. Опишіть алгоритм скалярного множення точки на число.
8. Опишіть алгоритм Діффі-Хелмана на еліптичних кривих.

СПИСОК ВИКОРИСТАНИХ ТА РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Бабенко Т.В. Криптологія у прикладах, тестах і задачах: навч. посібник / Т.В.Бабенко, Г.М.Гулак, С.О.Сушко, Л.Я.Фомичова. – Д.: Національний гірничий університет, 2013. – 318 с.
2. Бобало Ю. Я. Інформаційна безпека: навч. посібник / Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник та ін.; за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого. – Львів : Видавництво Львівської політехніки, 2019. – 580 с.
3. Болотов А. А. Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых / Болотов А. А., Гашков С. Б., Фролов А. Б. – М.: КомКнига, 2006. – 280 с.
4. Глинчук Л.Я. Криптологія: навч.-метод. посіб. / Л. Я. Глинчук – Луцьк: Вежа-Друк, 2014. – 164 с.
5. Горбенко І. Д. // Прикладна криптологія: Теорія. Практика. Застосування / І. Д. Горбенко, Ю. І. Горбенко. – Харків : Форт, 2013. – 880 с.
6. ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення. – Введ. 01–07–2015. – К. :Мінекономрозвитку України, 2015.
7. Задірака В.К. Комп'ютерні технології криптографічного захисту інформації на спеціальних цифрових носіях: навч. посіб. / В. К. Задірака, А. М. Кудін, В. О. Людвиченко, О. С.Олексюк. – К. -Тернопіль: Підручники і посібники, 2007. – 272 с.
8. Козіна Г.Л. Криптографія від історії до сучасних стандартів: навч.посібник / Г. Л. Козіна. – Запоріжжя : НУ «Запорізька політехніка», 2020. – 192 с.
9. Корченко О. Г. Прикладна криптологія: системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.
10. Маркова І.І. Захист інформації. Криптографічні методи: Підручник для вищих навчальних закладів. / І.І. Маркова, А.І. Рибак, Ю.С. Ямпольський. – Одеса, 2001. – 175 с.
11. Alfred J. Menezes. Handbook of Applied Cryptography/ Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Publisher: CRC Press, 2001. – 780 pages
12. Bruce Schneier. Applied cryptography: protocols, algorithms, and source code in C / 2nd ed. – New York : JohnWiley & Sons, Inc.,1995. – 792 pages.
13. Jean-Philippe Aumasson. Serious Cryptography: A Practical Introduction to Modern Encryption Paperback. Kindle Edition, 2017. – 313 pages.
14. The CrypTool Portal [Електронний ресурс]. — Режим доступу : http://www.cryptool.org/en
15. CrypTool-Online [Електронний ресурс]. – Режим доступу: https://www.cryptool.org/en/cto/

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

Навчально-методичне видання

Інформаційна безпека та захист ПЗ

Методичні рекомендації до виконання лабораторних робіт

Підготували
Щур Наталія Олександрівна

Редактори
Комп’ютерне верстання –
Свідоцтво про реєстрацію № __ від _____ 202_ року
Підписано до друку __.__.21. Формат 60×84/16.
Ум. друк. арк. ____. Зам. __ офс.

Безкоштовно