Міністерство освіти і науки України
Житомирський державний університет імені Івана Франка
Кафедра комп'ютерних наук та інформаційних технологій

Дмитрій Вербівський
Богданна Якимчук

Криптологія

Опорний конспект лекцій

Житомир 2023

УДК 004:056.52(075.8)
В 31

Рекомендовано до друку вченою радою Житомирського державного університету імені Івана Франка
(протокол № 6 від 31 березня 2023 р.)

Рецензенти:

ПОПЛАВСЬКА Світлана – кандидат педагогічних наук, доцент, доцент кафедри природничих і соціально-гуманітарних дисциплін, проректор з навчальної роботи Житомирського медичного інституту Житомирської обласної ради.

ТОПОЛЬНИЦЬКИЙ Павло – кандидат технічних наук, доцент, доцент кафедри комп'ютерних технологій і моделювання систем Поліського національного університету.

УСАТА Олена – кандидат педагогічних наук, доцент, завідувач кафедри комп'ютерних наук та інформаційних технологій Житомирського державного університету імені Івана Франка.

В 31

Криптологія: опорний коспект лекцій / уклад.: Дмитрій Вербівський, Богданна Якимчук. Житомир: Вид-во ЖДУ ім. Івана Франка, 2023. 173 с.

Опорний конспект лекцій спрямований на ознайомлення студентів із основними цілями та поняттями криптології, загальними принципами побудови систем криптографічного захисту даних шляхом використанням сучасних симетричних та асиметричних алгоритмів. Структуру та зміст опорного конспекту лекцій визначено, виходячи зі змісту робочої програми навчальної дисципліни «Криптологія»

Курс лекцій призначений для студентів освітньо-професійної програми «Сучасні інформаційні технології та програмування».

УДК 004:056.52(075.8)

ЗМІСТ

ВСТУП 4

Лекція 1. **Основні поняття криптології** 5
Лекція 2. **Класичні шифри** 6
Лекція 3. **Симетричні алгоритми шифрування** 11
Лекція 4. **Нові стандарти шифрування** 15
Лекція 5. **Асиметричні алгоритми шифрування** 18
Лекція 6. **Електронний цифровий підпис** 24

СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ 46

ВСТУП

Захист даних будь-якої інформаційної системи від несанкціонованого доступу є однією з найважливіших умов успішного її функціонування. Нині значної актуальності набуває питання якості підготовки закладами вищої освіти майбутніх фахівців в галузі комп'ютерних наук, які б у своїй діяльності ефективно використовували різноманітні методи захисту інформації, зокрема криптографічні. Криптографія займається розробкою алгоритмів перетворення повідомлень, в тому числі шляхом шифрування з використанням спеціальних (ключових) даних. Дослідженням вразливих місць таких алгоритмів та розробкою методів зламу зашифрованих повідомлень займається криптоаналіз. Ці два наукових напрями тісно пов'язаних між собою і разом складають науку криптологію.

Опорний конспект лекцій з дисципліни «Криптологія» містить шість лекцій, представлений у схемах й таблицях. В предсталеному лекційному курсі висвітлено та розглянуто: основні поняття й визначення криптології; класифікацію алгоритмів шифрування; класичні криптосистеми та методи їх криптоаналізу; симетричні потокові та блокові алгоритми шифрування, режими застосування блокових шифрів; асиметричні алгоритми шифрування; алгоритми обміну ключами; функції хешування; схеми створення цифрового підпису. Структуру та зміст опорного конспекту лекцій визначено, виходячи зі змісту робочої програми навчальної дисципліни «Криптологія», її змістовних модулів, тем навчальних занять, теоретичного та практичного досвіду авторів.

Опорний конспект лекцій призначений для здобувачів вищої освіти в галузі 12 «Інформаційні технології» за спеціальністю 122 «Комп'ютерні науки».

ЛЕКЦІЯ 1

Основні поняття криптології

План

1. Основні поняття

2. Основні цілі та принципи криптології

3. Класифікація шифрів

Як таємно від інших передати важливе повідомлення певній людині?

створити технічно захищений канал передачі даних

витрати на обладнання

приховати сам факт існування секретного повідомлення стеганографічними методами

перетворити повідомлення криптографічними методами

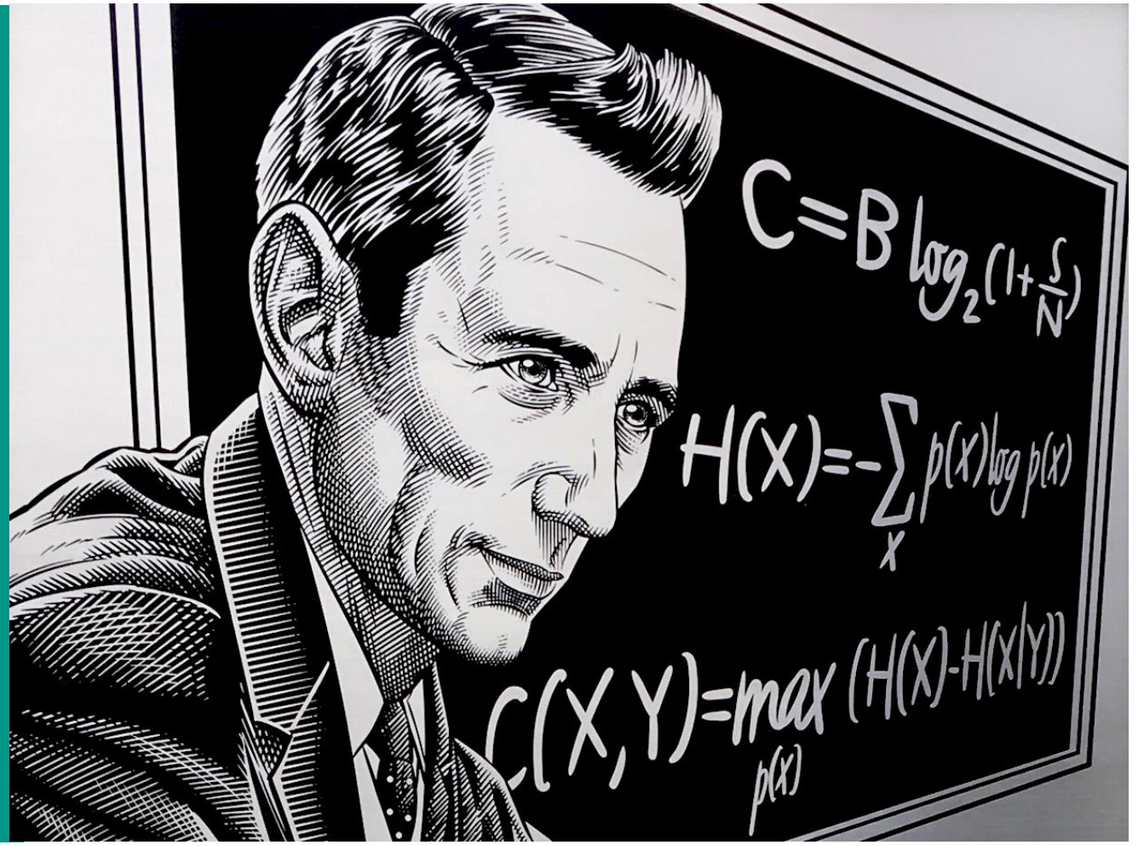
можна передавати по відкритих каналах зв'язку

1. Основні поняття

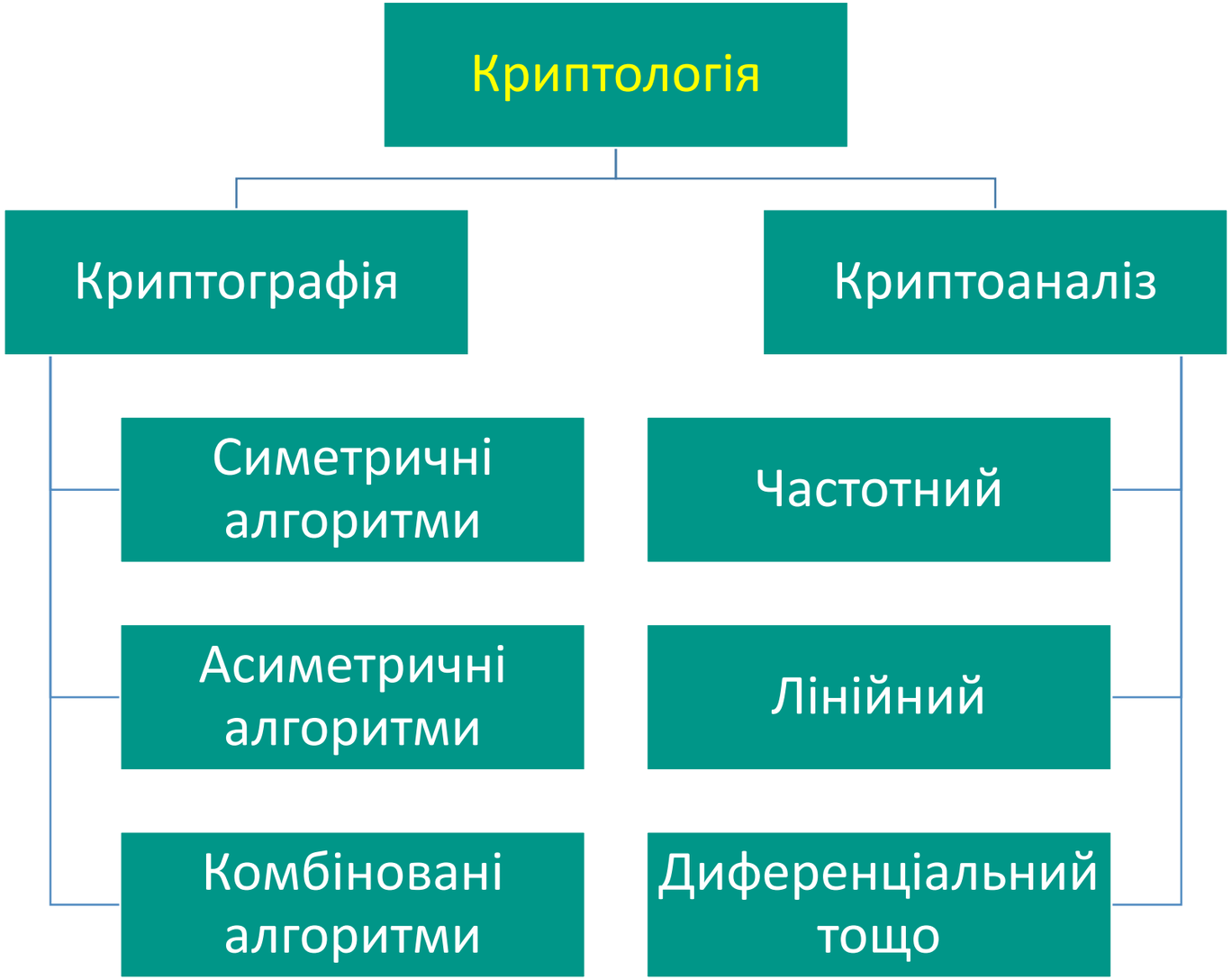
Криптологія (грецьк. «таємний» та «слово, вчення») – наука, яка вивчає методи побудови та аналізу систем захисту інформаційних ресурсів, основаних на математичних перетвореннях даних з використанням секретних параметрів.

1. Основні поняття

Фундамент криптології як науки у 1949 р. заклала робота американського вченого Клода Шеннона «Теорія зв’язку в секретних системах», у якій фактично вперше було представлено математичну модель шифрів.



1. Основні поняття



1. Основні поняття

Криптографія (грецьк. «таємний» та «писання», «тайнопис») – наука про принципи, засоби та методи перетворення даних з метою приховування їх змісту, запобігання несанкціонованого використання або підробки.

Криптоаналіз (грецьк. «таємний» та «аналіз») – наука про методи та способи розкриття зашифрованих повідомлень, а також про тактику та стратегію їх застосування.

1. Основні поняття

Криптографічний алгоритм – набір математичних правил та процедур, який описує такі види перетворень, як шифрування, формування та перевірка ЕЦП, обчислення хеш-значень, криптографічних контрольних сум, створення імітовставки тощо.

Сукупність криптографічних алгоритмів, що використовуються для шифрування називають **шифром**.

1. Основні поняття

Криптографічний ключ (ключ)– таємний стан деяких параметрів алгоритму криптографічного перетворення, що забезпечує вибір одного перетворення із сукупності можливих для даного алгоритму.

1. Основні поняття

Шифрування даних – це процес, що складається із

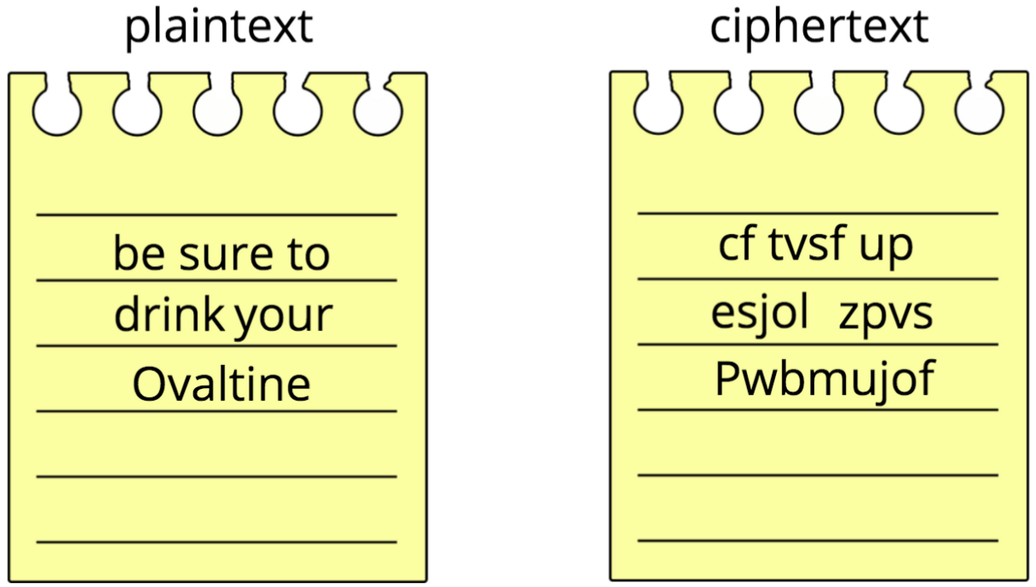
Зашифрування – процес перетворення відкритого тексту до виду, незрозумілого несанкціонованому користувачеві.

Розшифрування (син. **дешифрування**) – процес перетворення шифрованого повідомлення до початкової інформації (відкритого тексту) за допомогою певних правил шифру та відомого ключа.

1. Основні поняття

Відкритий текст являє собою вихідне повідомлення, що підлягає зашифруванню.

Результатом зашифрування відкритого тексту є **шифротекст**, що також називають **криптотекстом** або **криптограмою**.



1. Основні поняття

Криптосистема – це система криптографічного перетворення даних, що містить у собі п’ять компонентів:

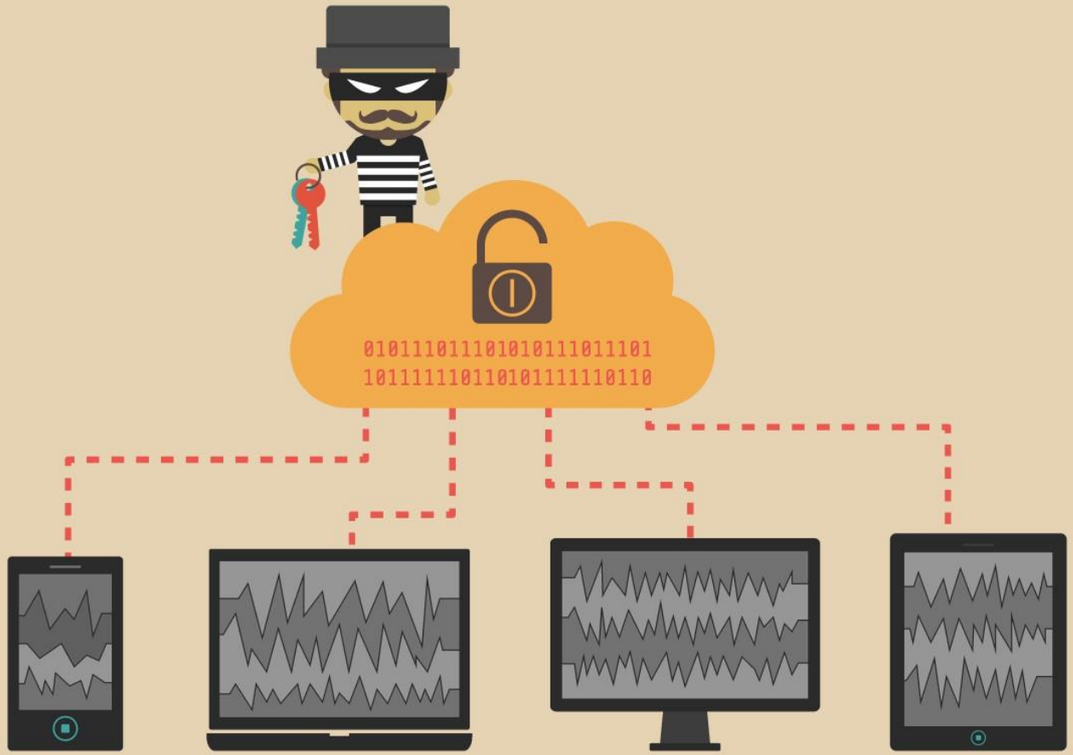
- множину відкритих текстів
- множину шифро-текстів
- множину ключів
- сімейство зашифро-вуючих перетворень
- сімейство розшифро-вуючих перетворень

1. Основні поняття

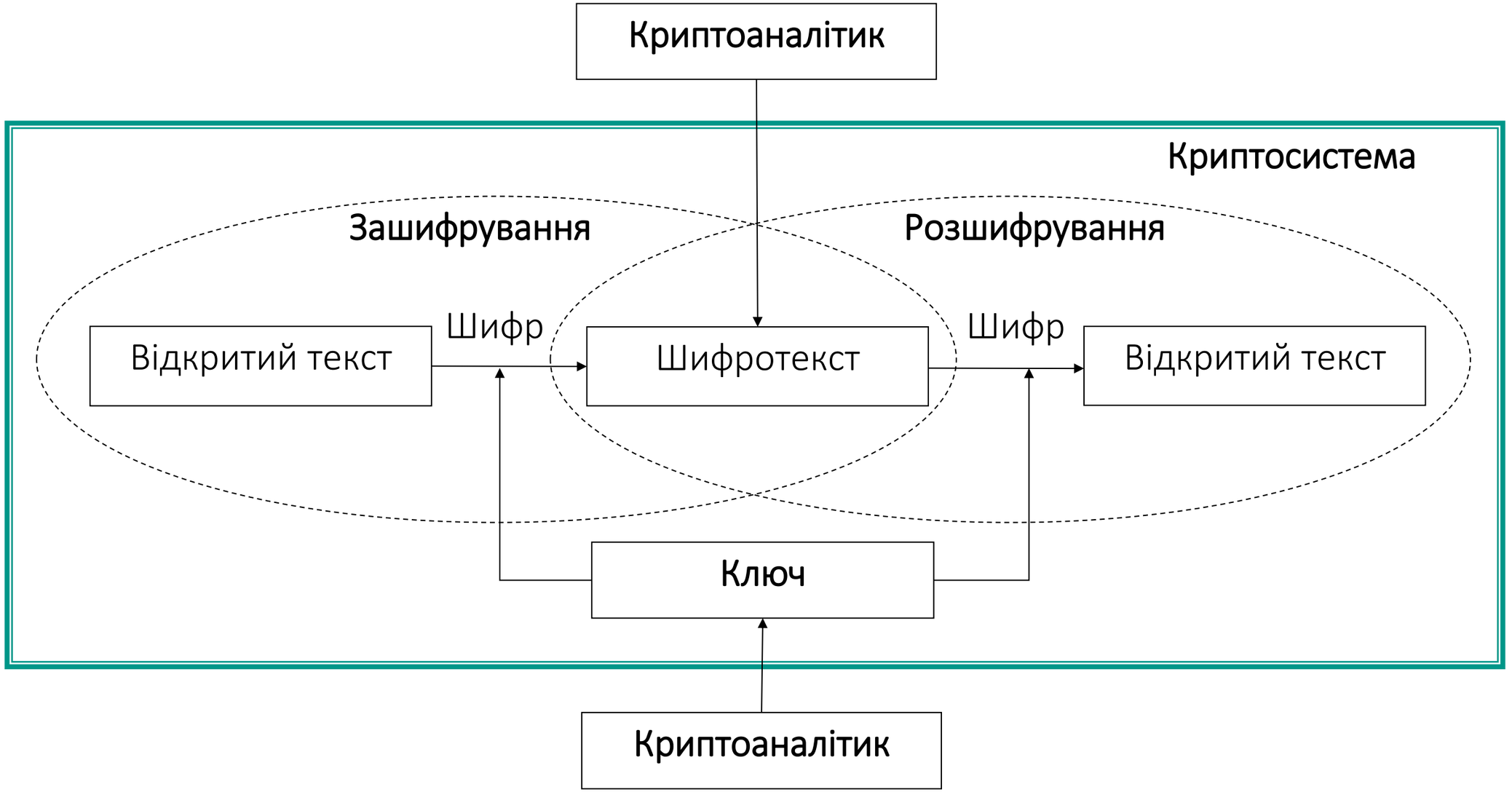
Криптостійкість – це властивість криптосистеми протидіяти атакам супротивника, спрямованим на отримання секретного ключа або відкритого повідомлення.

Під **атакою** на криптосистему розуміється спроба порушення безпеки конкретної реалізації криптосистеми.

Вдалу криптоатаку називають **зламом**.



1. Основні поняття



2. Основні цілі криптології

Конфіденційність

захист від несанкціонованого ознайомлення зі змістом

Цілісність

захист від несанкціонованої зміни повідомлення

Автентичність (достовірність)

правильність даних, відсутність помилок, автентичність сторін

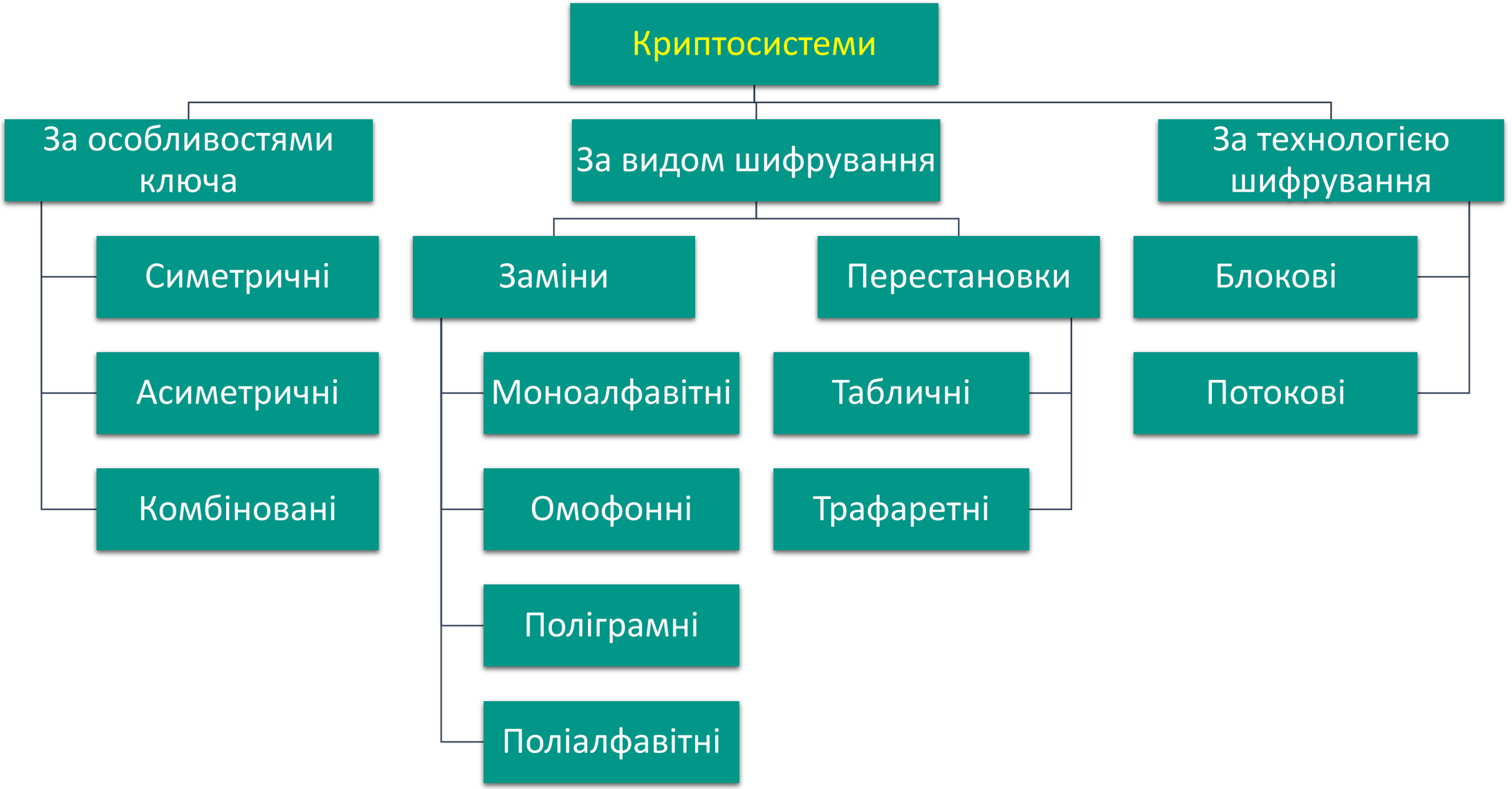
Невідмовність

неспростовність, неможливості відмови від авторства

2. Принципи криптології

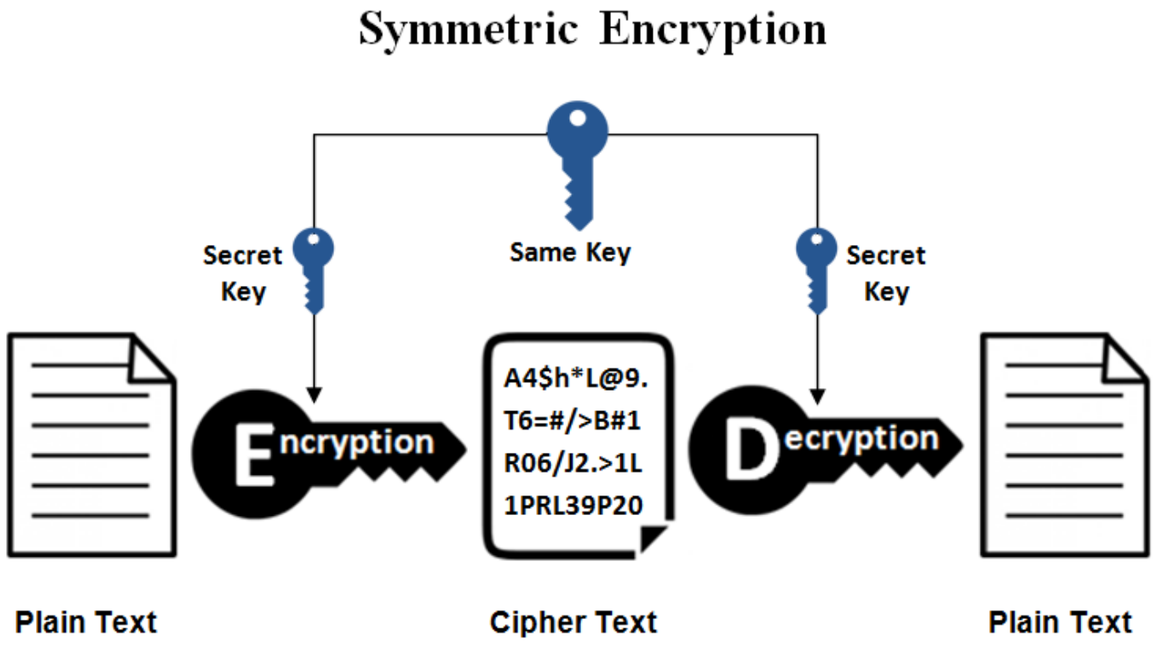
Принцип рівної міцності захисту	повідомлення мають бути **однаково міцно захищені** на **кожній ланці** передачі даних, в залежності від загроз, що виникають
Принцип доцільності захисту	проблема **співвідношення вартості** даних, витрат на їх захист та витрат на їх злам
Принцип використання ключа	без знання **ключа дешифрування** даних має бути практично **неможливим**
Принцип стійкості шифру	здатність шифру **протидіяти** різноманітним атакам на нього є його стійкістю, вона оцінюється шляхом різноманітних спроб його зламування
Принцип Керкхоффа	стійкість сучасного шифру має визначатись, **в першу чергу, ключем**
Принцип використання різноманітних шифрів	**вибір шифру** залежить **від особливостей інформації**, від її обсягу, цінності тощо

3. Класифікація шифрів



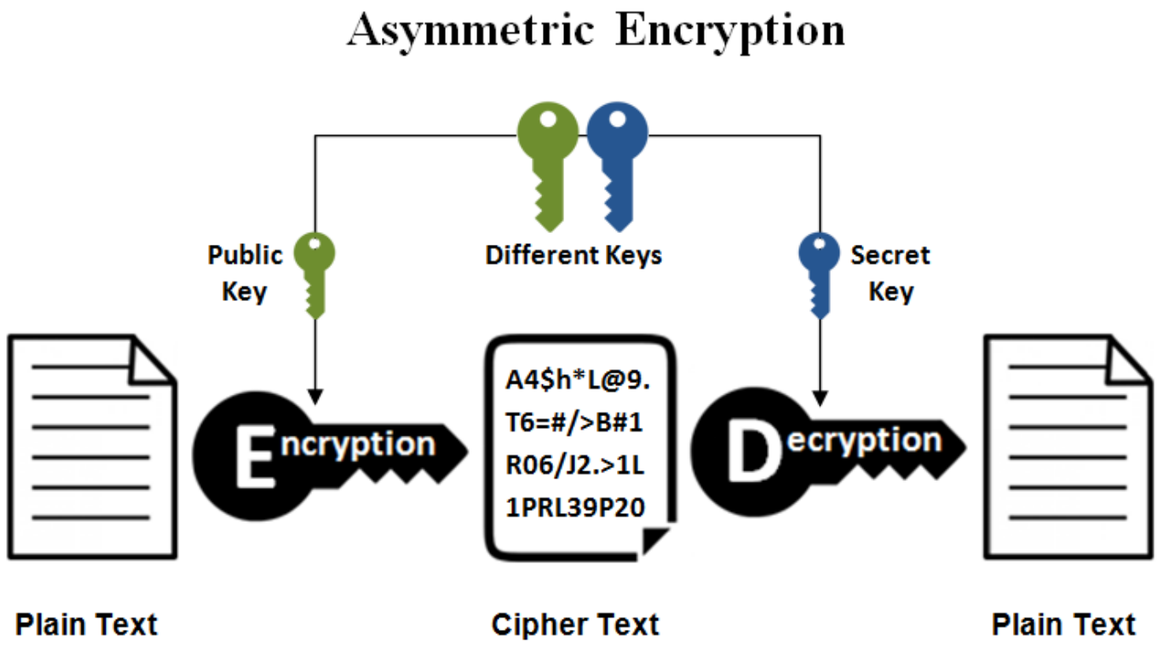
3. Класифікація шифрів

Симетричний шифр (з закритим ключем) – метод шифрування, в якому один і той самий алгоритм, а також один і той самий ключ використовується для шифрування та дешифрування повідомлень.



3. Класифікація шифрів

Асиметричний шифр (з відкритим ключем) – метод шифрування, в якому алгоритми шифрування та дешифрування різні, і використовуються два ключа – один для шифрування, а другий – для дешифрування повідомлень.

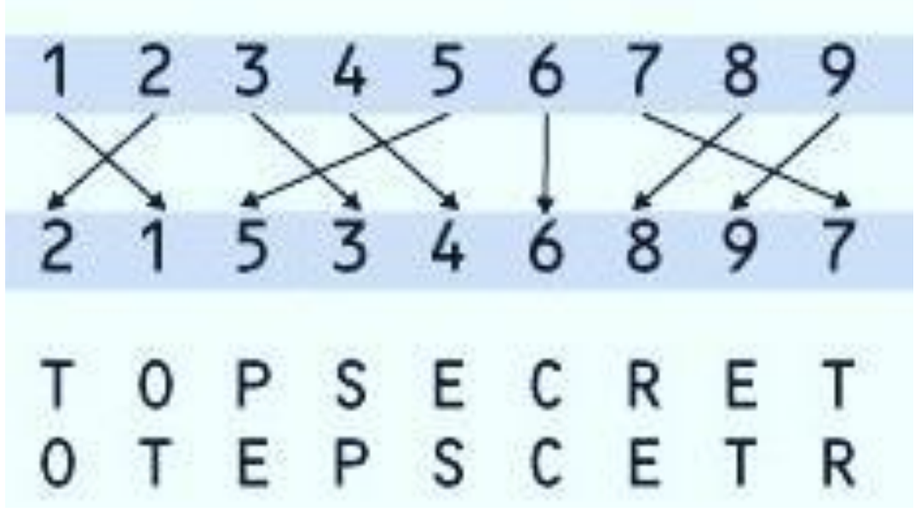


3. Класифікація шифрів

Шифр заміни (підстановки) – це шифр, у якому кожен символ відкритого тексту у шифротексті замінюється іншим символом.

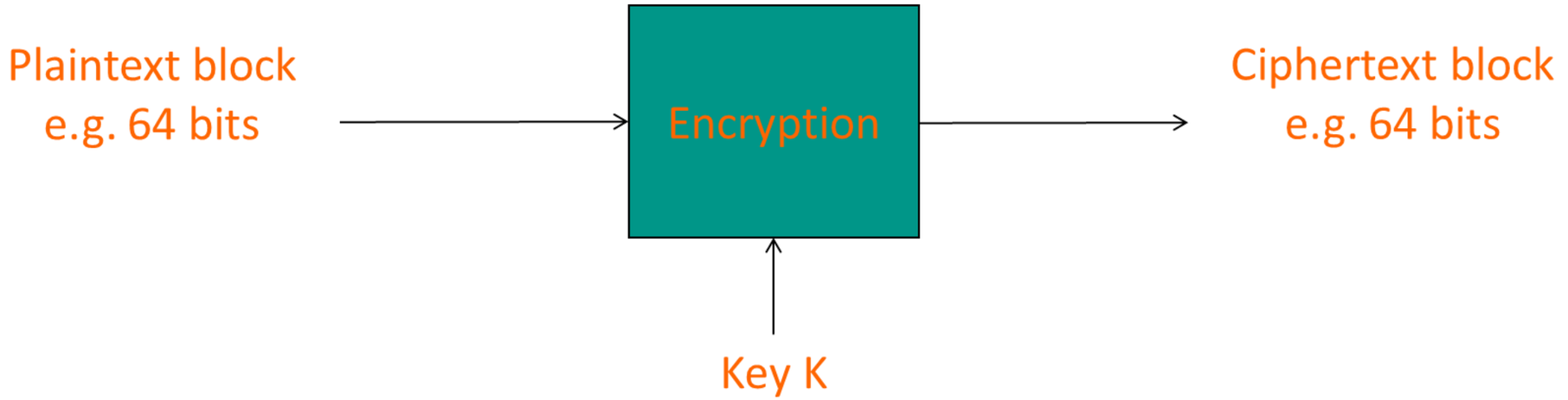
3. Класифікація шифрів

Шифр перестановки – це шифр, у якому символи повідомлення переставляються місцями безпосередньо у відкритому тексті за певним правилом, що залежить від ключа.



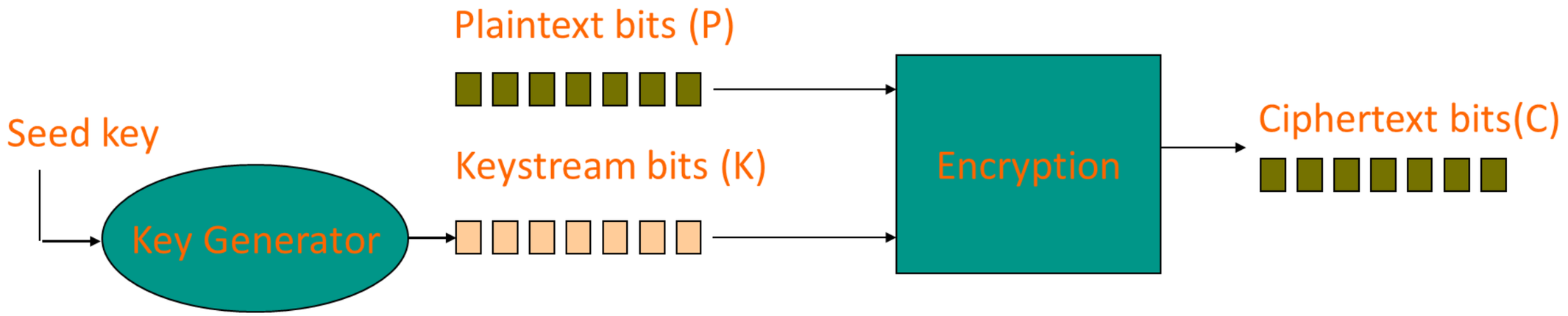
3. Класифікація шифрів

Блокові шифри здійснюють шифрування блоків фіксованої довжини, що складаються з послідовності символів відкритого тексту.



3. Класифікація шифрів

Потокові шифри здійснюють шифрування окремих символів відкритого тексту.



ЛЕКЦІЯ 2

Класичні шифри

План

1. Скитала
2. Квадрат Полібія
3. Шифр Цезаря
4. Шифр частоколу
5. Шифр Віженера
6. Шифр Плейфера
7. Криптосистема Хілла

1. Скитала

В якості носія повідомлення застосовувалася вузька та довга стрічка пергаменту (папірусу).

Ключ: діаметр палиці

1. Скитала

Шифрування:

стрічка намотувалася на палицю у вигляді спіралі і на неї вздовж палиці наносився текст секретного повідомлення. Після цього стрічка змотувалася і посилалася адресату

Дешифрування:

використовувалася палиця такого самого діаметру або дешифрувальний пристрій «Антискитала» (винайшов Аристотель – запропонував використовувати конусоподібний «спис»)

2. Квадрат Полібія

Ключ: розташування літер в квадраті

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

2. Квадрат Полібія

Шифрування:

літера, що зашифровується, замінюється на її координати в квадраті

Дешифрування:

пара літер однозначно визначає літеру в квадраті

2. Квадрат Полібія

Приклад 2.1:

Повідомлення: CRYPTO

Шифруємо: AC DB ED CE DD CD

Шифротекст: ACDBEDCEDDCD

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

3. Шифр Цезаря

Ключ:

число n – від 1 до 25

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

3. Шифр Цезаря

Шифрування:

літеру тексту замінюємо на літеру в абетці на n позицій праворуч. Зациклюємо абетку, ігноруємо коми та пробіли

Дешифрування:

літеру зашифрованого тексту замінюємо на літеру розташовану в абетці на n позицій ліворуч. Зациклюємо абетку, ігноруємо коми та пробіли

3. Шифр Цезаря

Приклад 3.1:

Ключ: n=3

Повідомлення: CAESAR

Шифруємо: C=2+3=5=F A=0+3=3=D E=4+3=7=H

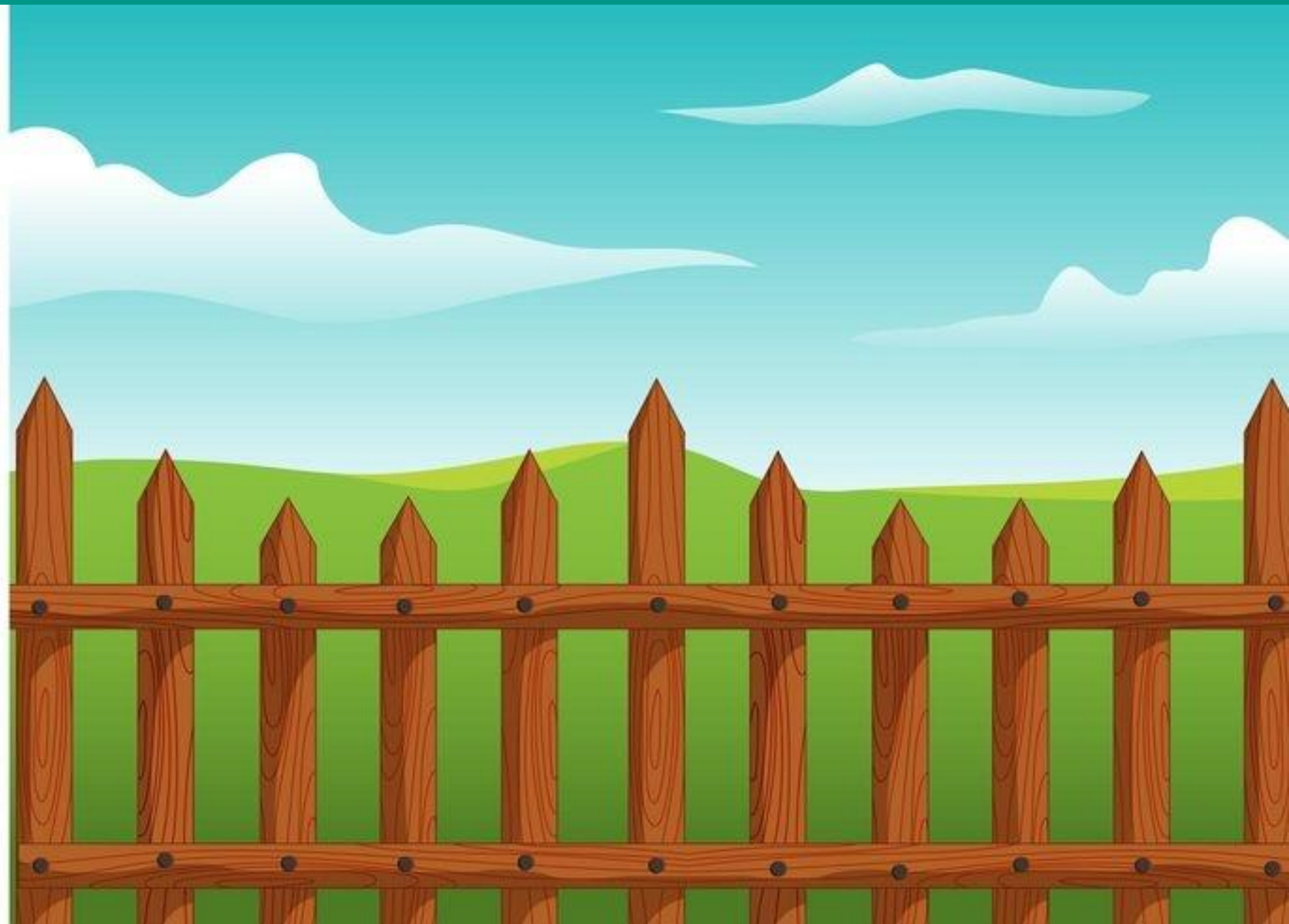
S=18+3=21=V A=0+3=3=D R=17+3=20=U

Шифротекст: FDHVDU

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

4. Шифр частоколу

Ключ: ціле число n – висота частоколу



4. Шифр частоколу

Шифрування:

літери повідомлення записуємо як степені (їх кількість висота частоколу), а потім записуємо літери в степенях по рядках зверху донизу

Дешифрування:

підраховуємо літери, ділимо на ключ, записуємо літери по n штук (n - ключ) в порядку зверху донизу

4. Шифр частоколу

Приклад 4.1:

Ключ: n=3.

Повідомлення: я отримаю залік автоматом

Шифруємо:

		т	м	з	і	в	м	о
о	и	ю	л	а	о	т		
я	р	а	а	к	т	а	м	

Шифротекст: тмзівмооиюлаотяраактам

5.Шифр Віженера

Ключ: ключове слово K, якщо воно менше за повідомлення, то воно циклічно повторюється.

	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	K	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

5. Шифр Віженера

Шифрування:

кожна літера повідомлення замінюється на літеру, що знаходиться на перетині літер першого рядка (алфавіт повідомлення) і першого стовпчика (алфавіт ключа) в таблиці Віженера.

Дешифрування:

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

5. Шифр Віженера

Приклад 5.1:

Повідомлення: PURPLE

Ключ: SMART

Шифротекст: HGRGEW

	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	K	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

6. Шифр Плейфера

Шифр Плейфера є біграмним, тобто текст повідомлення розбивається на біграми (групи з двох символів).

Ключ: секретне слово та розташування літер у матриці.

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

Клітинки матриці заповнюються літерами ключового слова (виключаючи літери, що повторюються), в решту комірок записуються літери алфавіту, які не зустрічаються в ключовому слові, по порядку.

6. Шифр Плейфера

Шифрування:

дві літери біграми відповідають кутам прямокутника в ключовій матриці. Визначаються положення кутів цього прямокутника відносно один одного. Після чого кожну біграму зашифровують згідно правил (див. далі)

Дешифрування:

за правилами шифрування, тільки циклічно зміщуємо на крок вліво (вгору)

6. Шифр Плейфера

Правила шифрування біграм

1. Якщо дві літери біграми однакові – додаємо після першого символу «Х», 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.

6. Шифр Плейфера

Приклад 6.1:

Повідомлення: HIDE THE GOLD IN THE TREE STUMP

Ключ: PLAYFAIR EXAMPLE

Шифрування: HI DE TH EG OL DI NT HE TR EX ES TU MP

Шифротекст: BM ND ZB XD KY BE JV DM UI XM MN UV IF

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

7. Криптосистема Хілла

Літери алфавіту нумеруються в порядку їхнього зростання від 0 до 25. Всі операції з літерами відбуваються по модулю 26.

Ключ: матриця K(d×d), елементи якої числа від 0 до 25, det K≠0, d ≥ 2

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \quad Mod\ 26$$

Літери повідомлення перетворюють в набір цифр, потім розбивають на d-розмірні стовпчики

7. Криптосистема Хілла

Шифрування:

$K \cdot P_i = C_i \pmod{26}$,

де C_i – набір цифр, елементи яких числа від 0 до 25 ⇨ перетворюються на літери шифротексту

Дешифрування:

$P = K^{-1} \cdot C_i \pmod{26}$,

де K^{-1} – обернена матриця

7. Криптосистема Хілла

Приклад 7.1:

Повідомлення: HELP

Ключ: $K = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$

Шифрування: $K \cdot P_i = C_i \pmod{26}$

Шифротекст: HIAT

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

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

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

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

7. Криптосистема Хілла

Приклад 7.2:

Повідомлення: HIAT

Ключ: $K^{-1} = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$

Дешифрування: $P = K^{-1} \cdot C_i \pmod{26}$

Відкритий текст: HELP

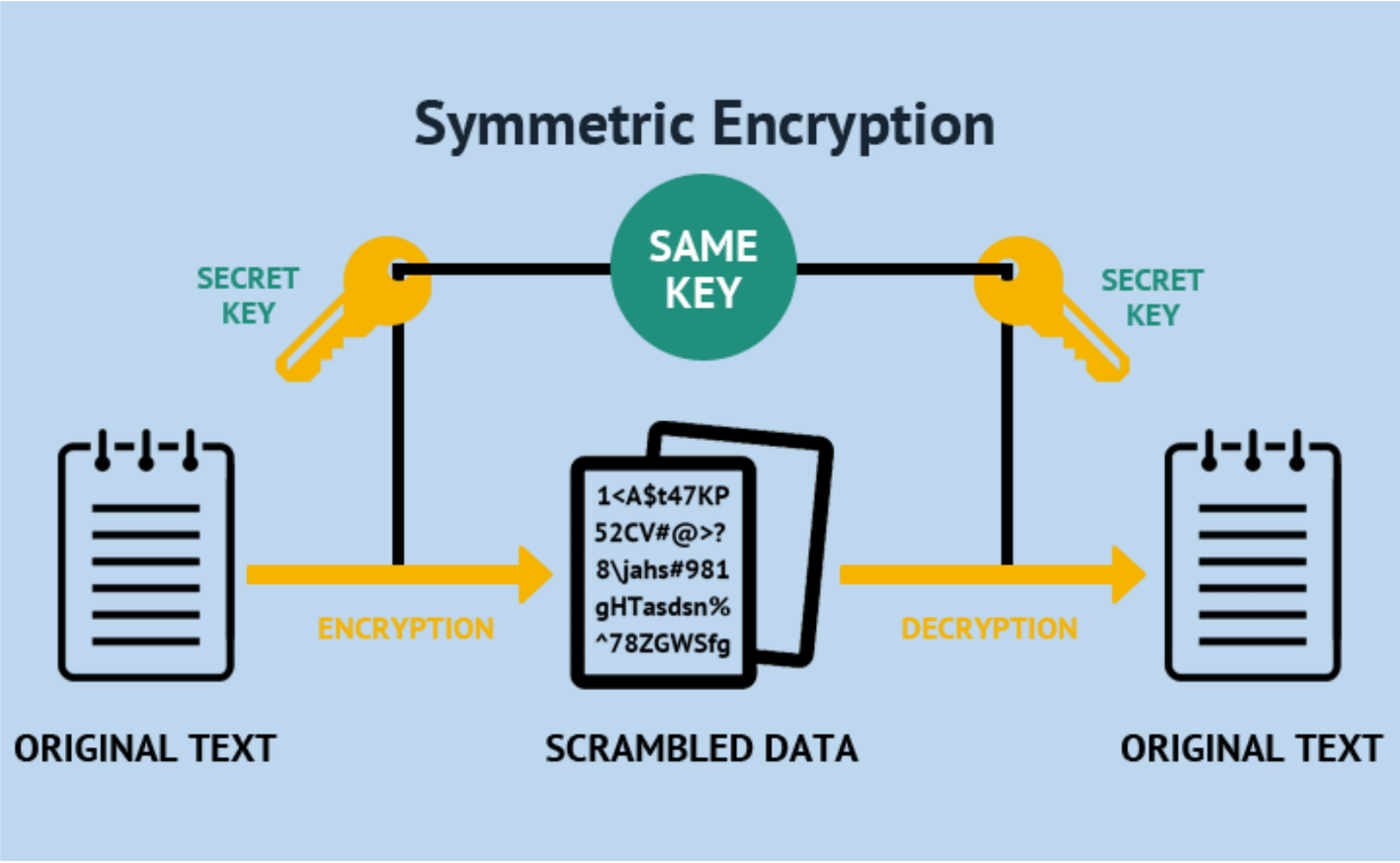
$$K^{-1} = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$$

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

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

ЛЕКЦІЯ 3

Симетричні алгоритми шифрування



План

1. Шифр одноразового блокнота
2. Мережа Фейстеля
3. Стандарт шифрування даних DES

1. Шифр одноразового блокнота

Творці – Гільберт Вернам зі співробітниками телеграфної компанії AT&T, а також офіцер армії США Джозеф Моборн (1917 рік)

Шифр Вернама є єдиною системою шифрування, для якої доведена абсолютна криптографічна стійкість (Клод Шеннон, 1949 рік)

Гільберт Вернам

Джозеф Моборн

1. Шифр одноразового блокнота

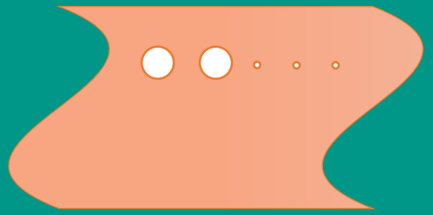
Ідея автоматичного шифрування телеграфних повідомлень

Відкритий текст представлявся у вигляді п'ятизначних імпульсних комбінацій на перфострічці

Ключ: перфострічка з випадковими знаками – «гама»

Наприклад, літера «А» мала вигляд:

+ + - - -



«+» – отвір

«-» – його відсутність

1. Шифр одноразового блокнота

Шифрування:

імпульси «гами» електромеханічно складалися з імпульсами знаків відкритого тексту. Отримана сума представляла собою шифротекст

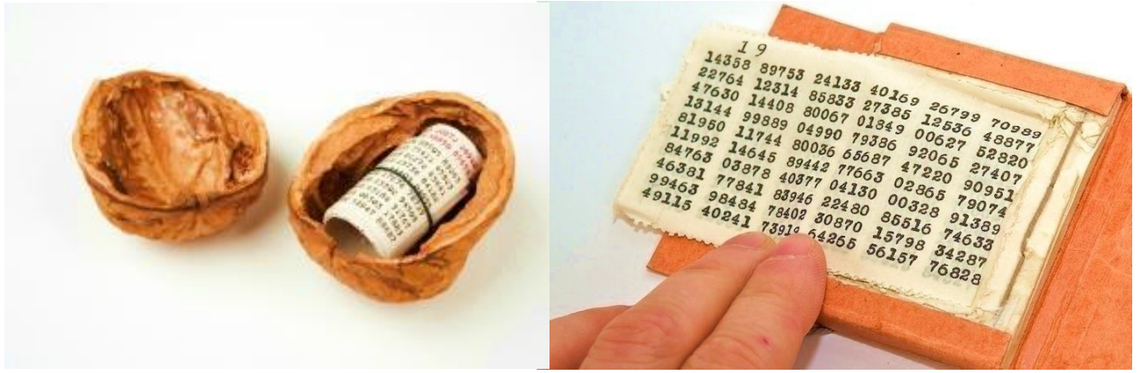
Дешифрування:

імпульси, отримані по каналу зв'язку, електромеханічно складалися з імпульсами тієї самої «гами», в результаті чого відновлювалися вихідні імпульси повідомлення

1. Шифр одноразового блокнота

Класичний одноразовий блокнот

Ключ: одноразовий блокнот – послідовність випадкових символів, написаних на аркушах паперу



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!

1. Шифр одноразового блокнота

Шифрування:

кожен символ ключа використовується для шифрування одного символу повідомлення

Дешифрування:

одержувач, використовуючи точно такий самий блокнот, дешифрує кожний символ шифротексту

1. Шифр одноразового блокнота

Приклад 1.1:
Ключ: SECRET LISTENERS
Повідомлення: MONITORING TIMES
Шифрування:

Відкритий текст	13 15 14 09 20 15 18 09 14 07 20 09 13 05 19
Ключова гама	19 05 03 18 05 20 12 09 19 20 05 14 05 18 19
Результат додавання	32 20 17 27 25 35 30 18 33 27 25 23 18 23 38
За модулем 26	06 20 17 01 25 09 04 18 07 01 25 23 18 23 12
Шифротекст	FTQAYIDRGAYWRWL

1. Шифр одноразового блокнота

Приклад 1.2:
Ключ: SECRET LISTENERS
Шифротекст: FTQAYIDRGAYWRWL
Дешифрування:

Шифротекст	06 20 17 01 25 09 04 18 07 01 25 23 18 23 12
Ключова гама	19 05 03 18 05 20 12 09 19 20 05 14 05 18 19
Результат віднімання	-13 15 14 -17 20 -11 -08 09 -12 -19 20 09 13 05 -07
За модулем 26	13 15 14 09 20 15 18 09 14 07 20 09 13 05 19
Відкритий текст	MONITORING TIMES

1. Шифр одноразового блокнота

Для шифрування бінарних даних (потоку бітів)

Ключ: послідовність випадкових бітів

$\oplus$	0	1
0	0	1
1	1	0

Виконується додавання бітів за модулем 2 (операція XOR, eXclusive OR – виключне або)

1. Шифр одноразового блокнота

Приклад 1.3:
Ключ: 00001011 00010010 00001111
Повідомлення: SUN
Шифрування:

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

2. Мережа Фейстеля

У 1971 році Хорст Фейстель, дослідницький центр IBM створив шифр Lucifer, на базі якого згодом було розроблено шифр DES

Мережа Фейстеля – один з методів побудови блокових шифрів

Хорст Фейстель

2. Мережа Фейстеля

Дані, представлені в комп'ютерній пам'яті, розбиваються на блоки фіксованої довжини (наприклад, 64 біта, 128 біт).

Якщо останній фрагмент коротше довжини блоку – його доповнюють будь-яким способом (наприклад, незначущими нулями)

Ключ: послідовність бітів фіксованої довжини, з якої генеруються *N* раундових ключів за яким-небудь математичним правилом

2. Мережа Фейстеля

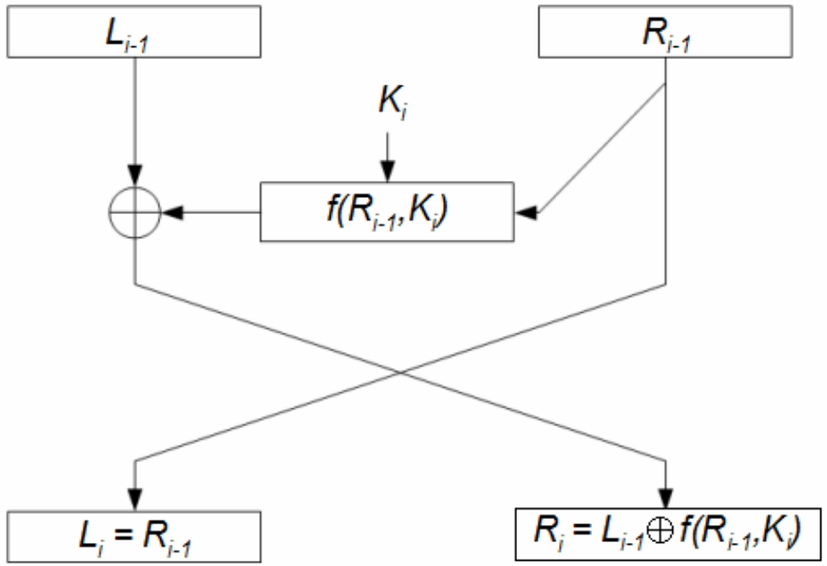
1. Кожен блок даних розбивається на дві рівні частини – ліву (*L*) і праву (*R*)

2. Права частина видозмінюється деякою функцією *f*(*R*,*K*) залежно від раундового ключа *K*

3. Здійснюється додавання за модулем 2 лівої частини *L* та *f*(*R*,*K*)

4. Результат додавання присвоюється новому правому підблоку, а правий підблок присвоюється без змін новому лівому підблоку (вхідні дані для наступного раунду)

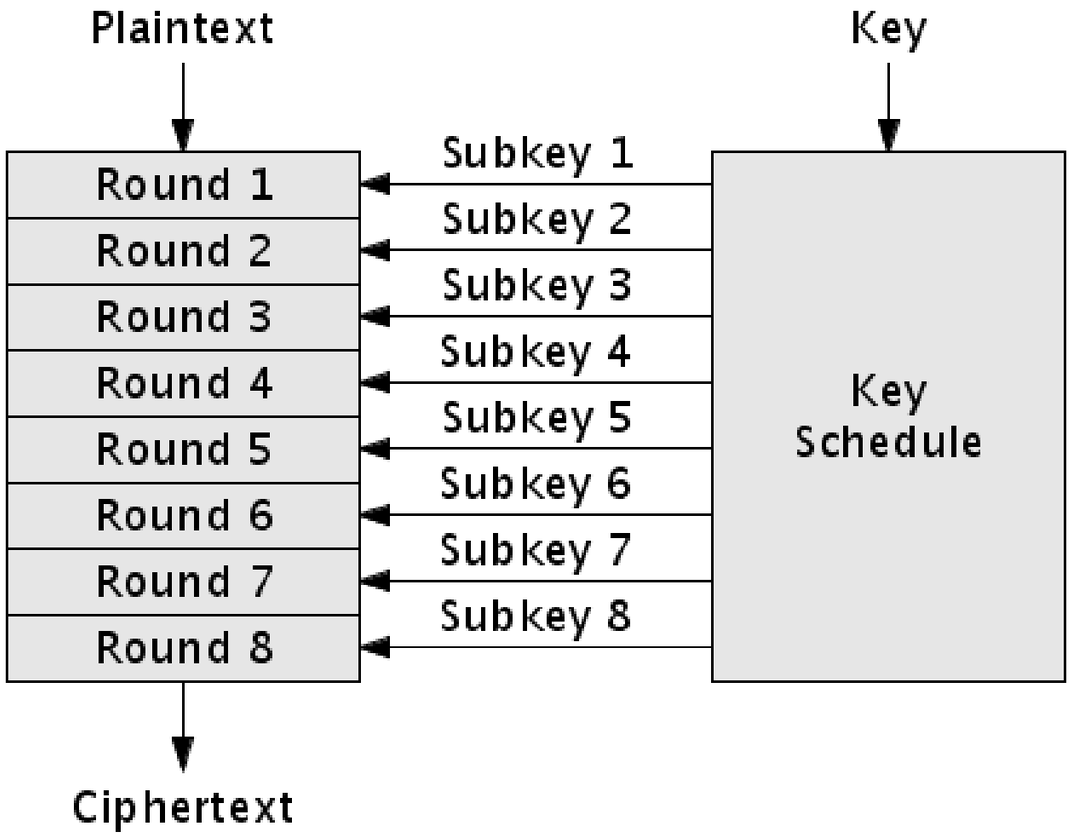
Шифрування:



2. Мережа Фейстеля

Описані операції повторюються *N-1* разів, де *N* – кількість раундів.

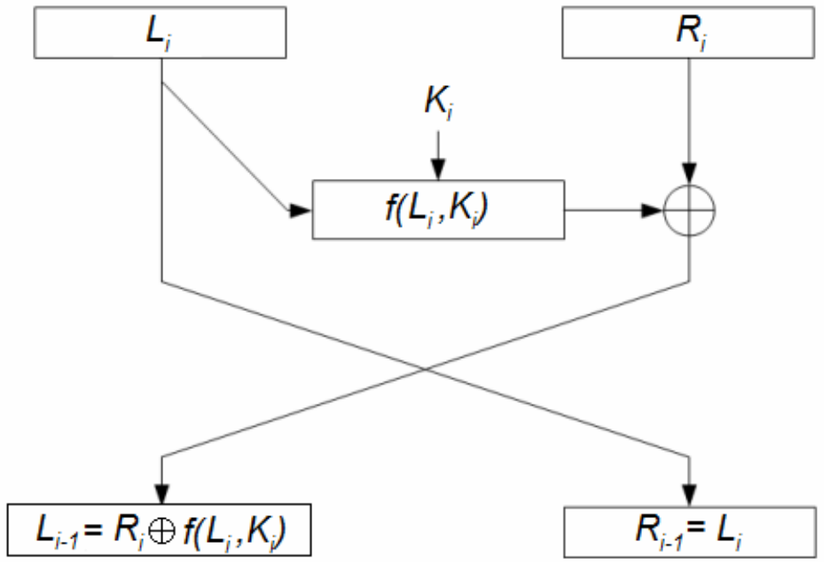
При переході від одного раунду до іншого міняються раундові ключі (K_0 на K_1 і т.д.)



2. Мережа Фейстеля

Дешифрування відбувається так само, як і шифрування, за винятком лише того, що ключі йдуть у зворотному порядку, тобто не від першого до *N*-го, а від *N*-го до першого

Дешифрування:



3. Стандарт шифрування даних DES

DES (Data Encryption Standard) розроблений фірмою IBM і затверджений урядом США у 1977 році як офіційний стандарт шифрування

Ключ: 56 бітний блок + 8 біт контролю парності, що розміщені в позиціях 8, 16, 24, 32, 40, 48, 56, 64 (використовується при знаходженні помилок при обміні та зберіганні ключів)

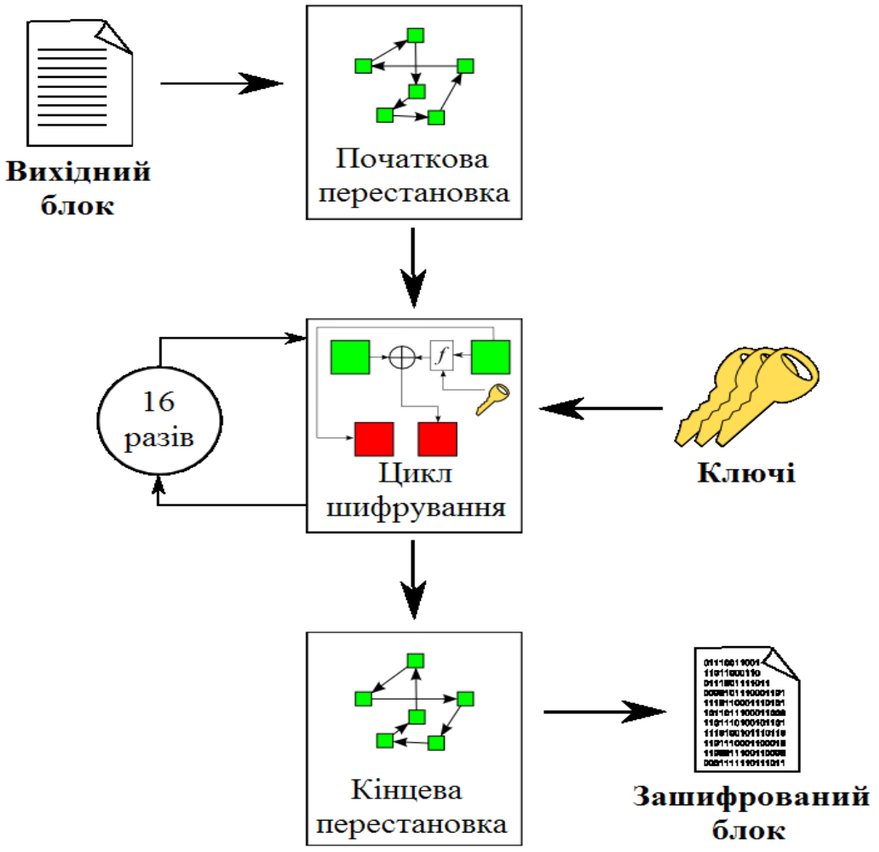
3. Стандарт шифрування даних DES

Загальна схема алгоритму DES

Вихідний текст – блок 64 біт

Процес шифрування складається з
- початкової перестановки;
- 16-ти раундів шифрування;
- кінцевої перестановки

Зашифрований текст – блок 64 біт



3. Стандарт шифрування даних DES

Початкова перестановка

Початковий текст T (блок 64 біт) перетворюється за допомогою початкової перестановки IP(Initial Permutation) за таблицею

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

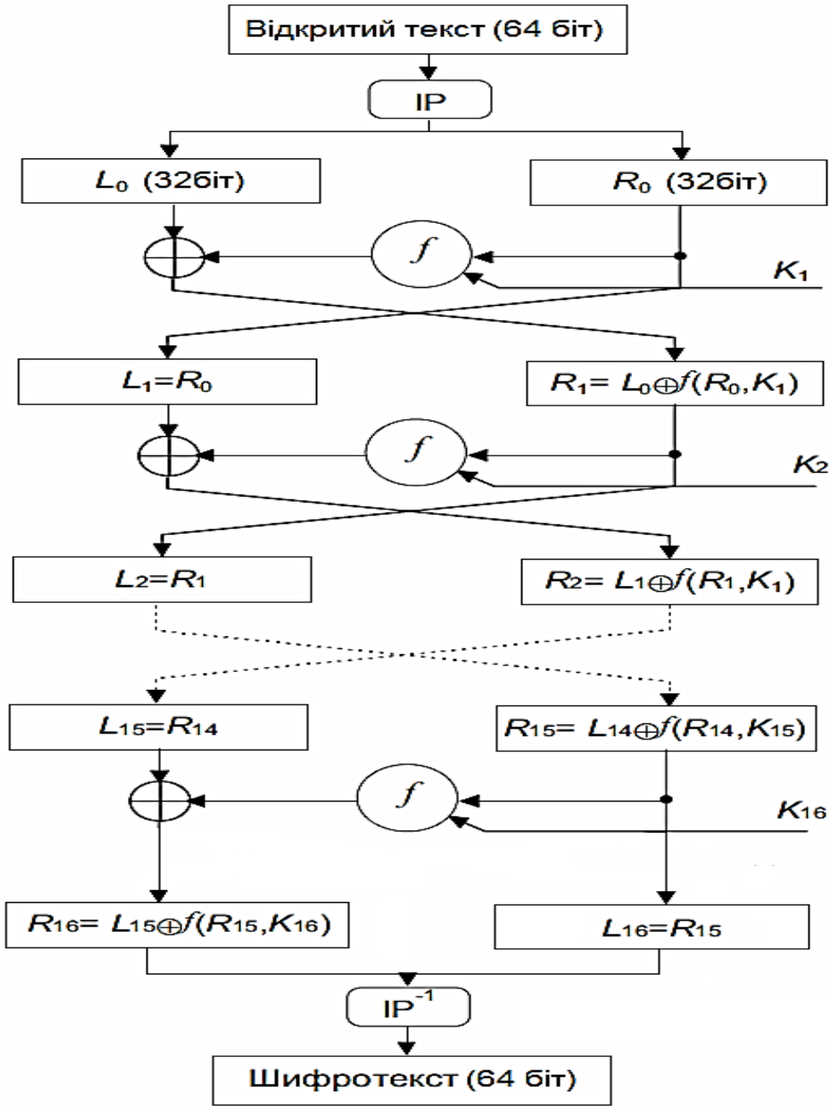
3. Стандарт шифрування даних DES

Раунди шифрування за мережею Фейстеля

1. Розбити IP(T) на дві половини L_0, R_0, де L_0 – 32 старших біта, а R_0 – 32 молодших біта даного блоку

2. Права половина R_i – це бітове додавання L_{i-1} та $f(R_{i-1},K_i)$ за модулем 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: $L_i = R_{i-1}$



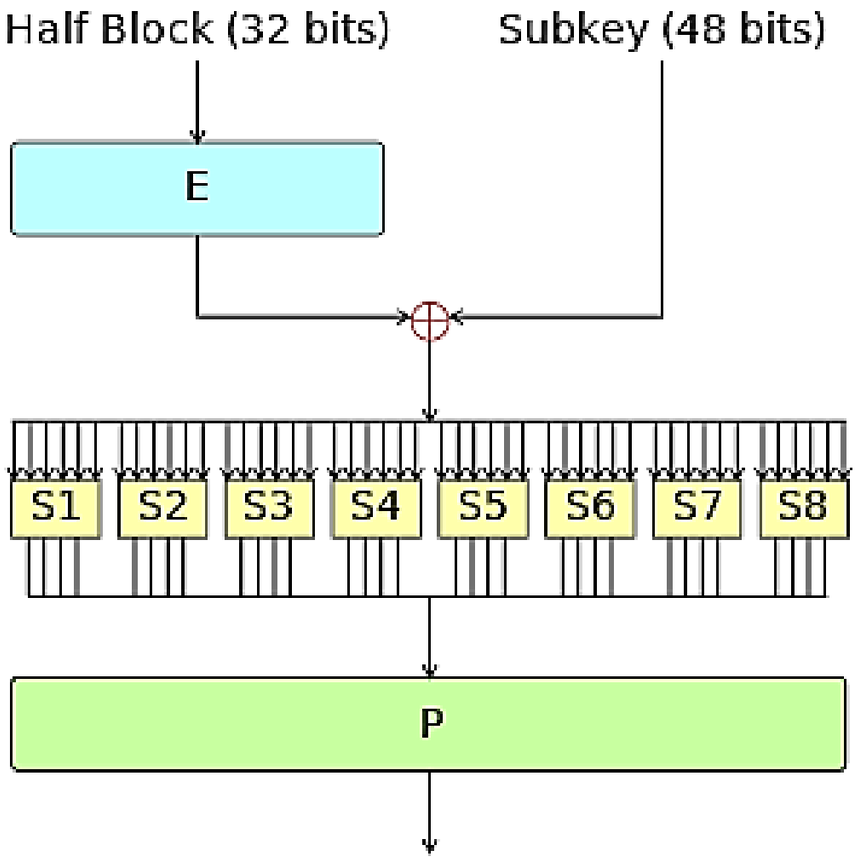
3. Стандарт шифрування даних DES

Основна функція шифрування (функція Фейстеля)

Аргументи функції *f:*
32-бітовий вектор R_{i-1} та
48-бітовий раундовий ключ K_i

Для обчислення функції *f* використовуються:

- функція розширення *E;*
- перетворення *S*, яке складається з 8 перетворень *S*-блоків;
- перестановка *P*



3. Стандарт шифрування даних DES

Функція *E* розширює 32-бітовий вектор R_{i-1} до 48-бітового вектора $E(R_{i-1})$ шляхом дублювання деяких бітів R_{i-1}

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 за модулем 2 із раундовими ключами K_i. Потім представляється у вигляді восьми послідовних блоків $E(R_{i-1}) = B_1B_2...B_8$

3. Стандарт шифрування даних DES

Кожний B_j являється 6-бітовим блоком, що перетворюється у 4-бітовий блок B'_j за допомогою S-перетворень, що визначаються таблицями

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S1
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	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S2
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	S3
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	S4
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	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S5
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	S6
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	S7
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	S8
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	

3. Стандарт шифрування даних DES

Приклад 3.1:

B_3 = 101111. Знайти B'_3 - ?

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

$a = 11_2 = 3$

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

$b = 0111_2 = 7$

Пара чисел (a,b) визначає число, що знаходиться на перетині рядка a та стовпця b в таблиці S-перетворень.

У таблиці S3, число обумовлене парою (3,7)=7. Звідси B'_3 = 0111.

3. Стандарт шифрування даних DES

Отриманий 32-бітовий блок $B'_1B'_2...B'_8$ перетворюється за допомогою перестановки P, що задана таблицею

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

3. Стандарт шифрування даних DES

Генерація ключів

З ключа розміром 56 біт генерується 16 штук 48-бітних раундових ключів

Видаляються 8 контрольних бітів та виконується перестановка початкового 56-бітного 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	

3. Стандарт шифрування даних DES

56-бітовий ключ ділиться на дві половини по 28 біт C_0 та D_0, які потім циклічно зсуваються на один чи два біти ліворуч в залежності від раунду

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

3. Стандарт шифрування даних DES

Приклад 3.2:

Побудуємо C_1, D_1 з C_0, D_0.

Для цього виконаємо циклічний зсув ліворуч на один біт кожної половини

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

49	41	33	25	17	9	1	
58	50	42	34	26	18	10	C_1
2	59	51	43	35	27	19	
11	3	60	52	44	36	57	
55	47	39	31	23	15	7	
62	54	46	38	30	22	14	D_1
6	61	53	45	37	29	21	
13	5	28	20	12	4	63	

3. Стандарт шифрування даних DES

Після зсуву C_i, D_i вибирається 48 бітів з 56 бітів та міняється їх порядок за таблицею

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
44	49	39	56	34	53
46	42	50	36	29	32

3. Стандарт шифрування даних DES

Кінцева перестановка

Кінцева перестановка IP^{-1} є оберненою до перестановки IP та використовується для відновлення позицій. Кінцева перестановка визначається таблицею

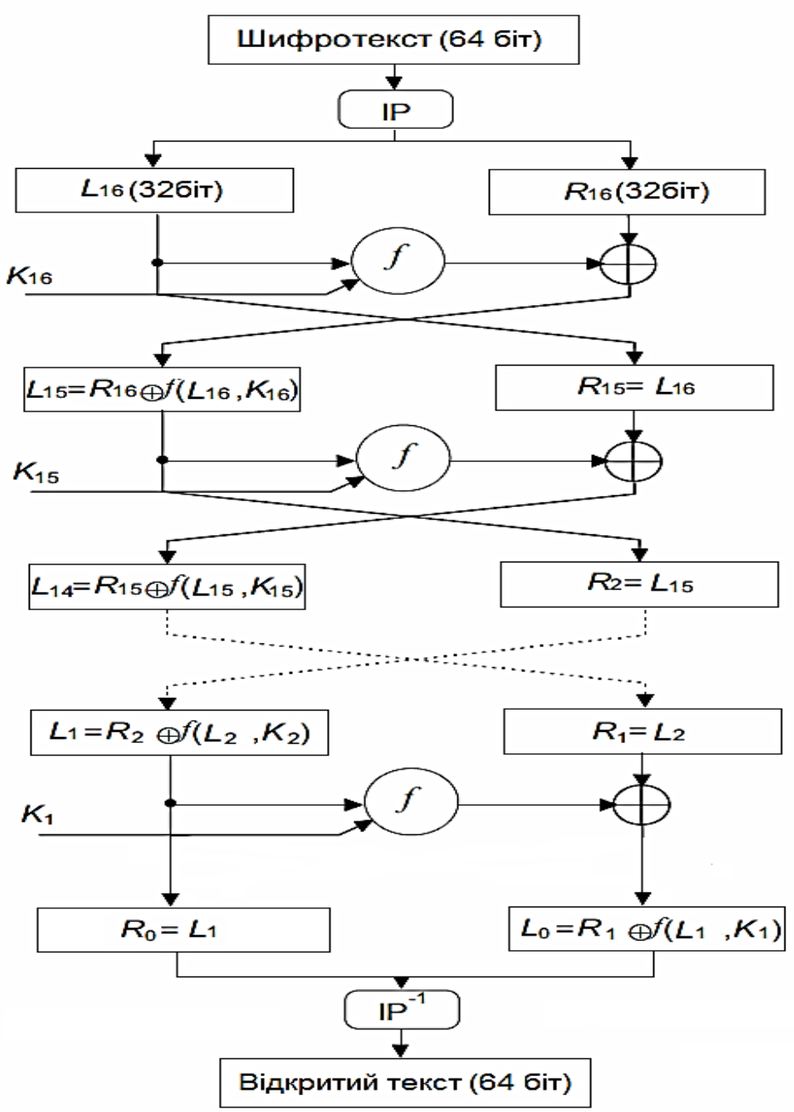
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

3. Стандарт шифрування даних DES

Дешифрування

При дешифруванні даних всі дії відбуваються в зворотному порядку. Ключі також застосовуються в зворотному порядку

Функція *f*, перестановки *IP* і IP^{-1} такі самі як і в процесі зашифрування



3. Стандарт шифрування даних DES

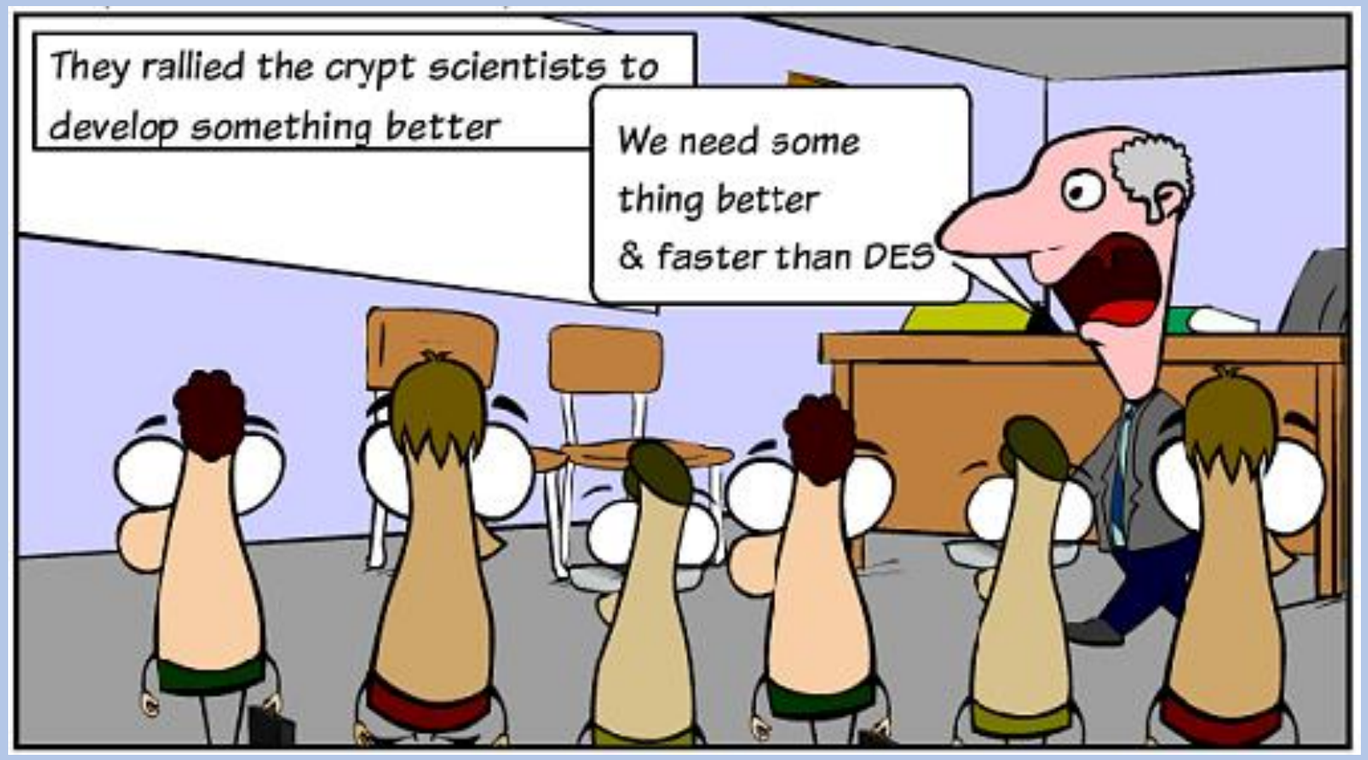
DES Visualization

3. Стандарт шифрування даних DES

Модифікації DES	Опис
Подвійний DES (2DES) **Потрійний DES** (3DES)	збільшено довжину ключів (2DES – 112 біт, 3DES – 168 біт)
DES з незалежними підключами	використання різних підключів на кожному раунді, не створюючи їх з одного 56-бітового ключа
DESX	3 підключі (64+56+64 біт)
CRYPT(3)	використовується у системах UNIX; додано незалежну від ключа перестановку з розширенням з 2^{12} варіантами
Узагальнений DES (Generalized DES, GDES)	загальний розмір блоку збільшився, а кількість обчислень залишилася незмінною
DES зі зміненими S-блоками	використовується змінний порядок S-блоків або міняється вміст самих S-блоків
RDES	наприкінці кожного раунду обмінюються місцями права й ліва половини з використанням залежної від ключа перестановки

ЛЕКЦІЯ 4

Нові стандарти шифрування



План

1. Новий стандарт шифрування AES

2. Міжнародний стандарт шифрування IDEA

3. Режими шифрування

1. Новий стандарт шифрування AES

Творці – бельгійські криптографи Вінсент Реймен та Йоан Дамен

Бельгійський алгоритм RIJNDAEL переміг у конкурсі NIST на новий блоковий симетричний стандарт шифрування, після чого був затверджений як стандарт та отримав назву AES (2001 рік)

Йоан Дамен

Вінсент Реймен

1. Новий стандарт шифрування AES

Математична база

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

$$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0,$$

де $a_i \in \{0,1\}$.

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

Приклад 1.1:

Байт: 01011010

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

1. Новий стандарт шифрування AES

Додавання байтів

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

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

де $c_i = a_i \oplus b_i$

Приклад 1.2:

У двійковій формі:

$$\begin{array}{rcccccccc} & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ \oplus & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}$$

У вигляді многочленів додавання коефіцієнтів при однакових степенях відбувається за модулем 2 ($\mathbf{1} \cdot \boldsymbol{x^7} + \mathbf{1} \cdot \boldsymbol{x^7} = (\mathbf{1} \oplus \mathbf{1}) \cdot \boldsymbol{x^7} = \mathbf{0} \cdot \boldsymbol{x^7}$):

$$\left(\boldsymbol{x^7} + \boldsymbol{x^5} + \boldsymbol{x^4} + \mathbf{1}\right) + \left(\boldsymbol{x^7} + \boldsymbol{x^3} + \boldsymbol{x^2} + \boldsymbol{x} + \mathbf{1}\right) = \boldsymbol{x^5} + \boldsymbol{x^4} + \boldsymbol{x^3} + \boldsymbol{x^2} + \mathbf{1}$$

1. Новий стандарт шифрування AES

Множення байтів

Для множення у полі $GF(2^8)$ в AES використовується нерозкладний многочлен $m(x) = x^8 + x^4 + x^3 + x + 1$

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.

1. Новий стандарт шифрування AES

Приклад 1.3:

$$(x^6+x^5+x^4+x^2)\cdot(x^7+x^5+x^4+x)=x^{13}+x^{11}+x^{10}+x^7+x^{12}+x^{10}+x^9+$$
$$+x^6+x^{11}+x^9+x^8+x^5+x^9+x^7+x^6+x^3=x^{13}+x^{12}+x^9+x^8+x^5+x^3$$

$$\begin{array}{l|l}
x^{13}+x^{12}+x^9+x^8+x^5+x^3 & x^8+x^4+x^3+x+1 \\
\cline{2-2}
x^{13}+x^9+x^8+x^6+x^5 & x^5+x^4+1 \\
\cline{1-1}
x^{12}+x^6+x^3 & \\
x^{12}+x^8+x^7+x^5+x^4 & \\
\cline{1-1}
x^8+x^7+x^6+x^5+x^4+x^3 & \\
x^8+x^4+x^3+x+1 & \\
\cline{1-1}
x^7+x^6+x^5+x+1 &
\end{array}$$

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

1. Новий стандарт шифрування AES

Основним елементом, яким оперує AES, є байт – послідовність 8 біт, що обробляються як єдине ціле (в шістнадцятковій системі числення)

Розмір **блока** 128 біт

Довжина **ключа** може бути 128, 192 або 256 бітів

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, так і над рядками і стовпцями.

1. Новий стандарт шифрування AES

Блок проміжного результату називають станом

Матриця стану має 4 рядки та 4 стовпці (*Nb*)

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

Приклад 1.4:

Відкритий текст: AES USES A MATRIX ZZ

У шістнадцятковому вигляді:

00 04 12 14 12 04 12 00 0C 00 13 11 08 23 19

State			
00	12	0C	08
04	04	00	23
12	12	13	19
14	00	11	19

1. Новий стандарт шифрування AES

Ключ: матриця байтів, яка має 4 рядки і кількість стовпців (*Nk*), що дорівнює довжині ключа, поділеній на 32

Матриця ключа при *Nk*=4:

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

1. Новий стандарт шифрування AES

Кількість раундів 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

1. Новий стандарт шифрування AES

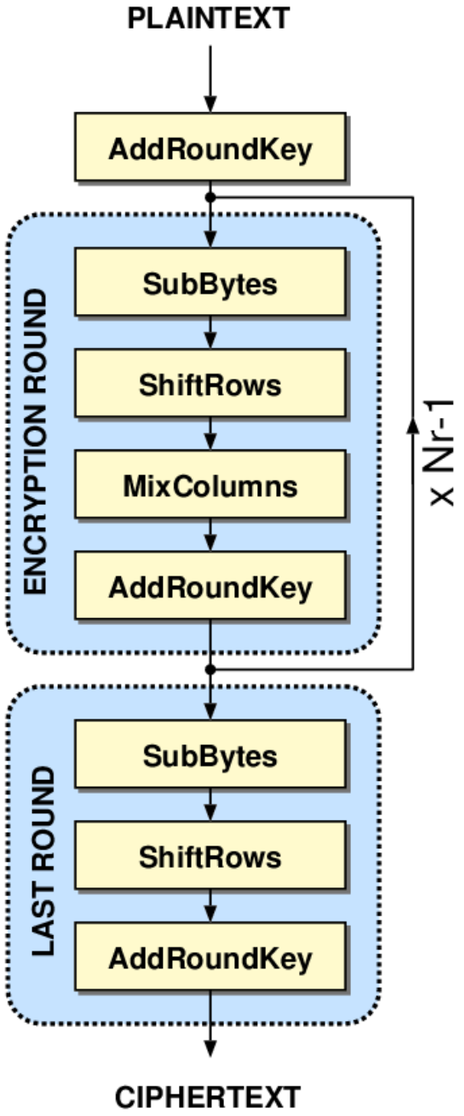
Шифрування за алгоритмом AES

I. Початкове 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. Завершальний раунд *Nr*, в якому пропускається перемішування стовпців



1. Новий стандарт шифрування AES

Підстановка байтів

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

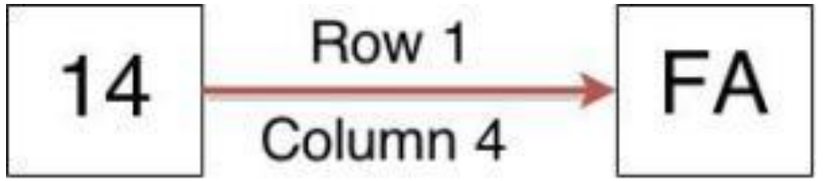
2. Над утвореним байтом виконують таке перетворення:

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

1. Новий стандарт шифрування AES

На основі двох попередніх перетворень створено спеціальну таблицю замін байтів, що називається *S*-боксом

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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



EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

1. Новий стандарт шифрування AES

Зсув рядків

Рядки стану циклічно зсувають праворуч 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

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

1. Новий стандарт шифрування AES

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:

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

1. Новий стандарт шифрування AES

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

•

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$02_{16} = 0000\ 0010_2 \rightarrow x$
$87_{16} = 1000\ 0111_2 \rightarrow x^7 + x^2 + x + 1$

$03_{16} = 0000\ 0011_2 \rightarrow x + 1$
$6E_{16} = 0110\ 1110_2 \rightarrow x^6 + x^5 + x^3 + x^2 + x$

$46_{16} = 0100\ 0110_2$
$A6_{16} = 1010\ 0110_2$

({02} · {87}) ⊕ ({03} · {6E}) ⊕ ({01}· {46}) ⊕ ({01}· {A6}) = (0000 0010 · 1000 0111) ⊕ (0000 0011 · 0110 1110) ⊕
⊕ 0100 0110 ⊕1010 0110 = 00010101 ⊕ 10110010 ⊕ 0100 0110 ⊕ 1010 0110 = 01000111=47_{16}

$\{02\} \cdot \{87\} = x \cdot (x^7 + x^2 + x + 1) = (x^8 + x^3 + x^2 + x) mod (x^8 + x^3 + x^2 + x + 1) = x^4 + x^2 + 1$

$$\begin{array}{l|l} x^8 + \quad x^3 + x^2 + x & x^8 + x^4 + x^3 + x + 1 \\ \hline x^8 + x^4 + x^3 + \quad x + 1 & 1 \\ \hline x^4 + \quad x^2 \quad + 1 & \end{array}$$

$\{03\} \cdot \{6E\} = (x + 1) \cdot (x^6 + x^5 + x^3 + x^2 + x) = x^7 + x^6 + x^4 + x^3 + x^2 +$
$+ x^6 + x^5 + x^3 + x^2 + x = x^7 + x^5 + x^4 + x$

1. Новий стандарт шифрування AES

Додавання раундового ключа

Виконується побітове додавання за модулем 2 раундового ключа до відповідних бітів, отриманих у попередньому раунді

Раундовий ключ отримують з розширеного ключа шифру

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$\oplus$

DC	9B	97	38
90	49	FE	81
37	DF	72	15
B0	E9	3F	A7

$=$

9B	DB	34	74
A7	9B	8E	1E
A3	3B	48	57
5D	4C	99	1B

1. Новий стандарт шифрування AES

Генерація ключів

Розширений ключ – одновимірний масив 4-байтових слів, які позначають $W[i]$

Загалом розширений ключ містить $W[Nb \cdot (Nr + 1)]$ слів, тобто для ключа в 128 біт отримаємо $4 \cdot (40 + 1)$= 44 слова

В процесі розширення ключа використовується масив раундових констант Rcon

0F	47	0C	AF
15	D9	B7	7F
71	E8	AD	67
C9	59	D6	98

*W*0 *W*1 *W*2 *W*3

DC	9B	97	38
90	49	FE	81
37	DF	72	15
B0	E9	3F	A7

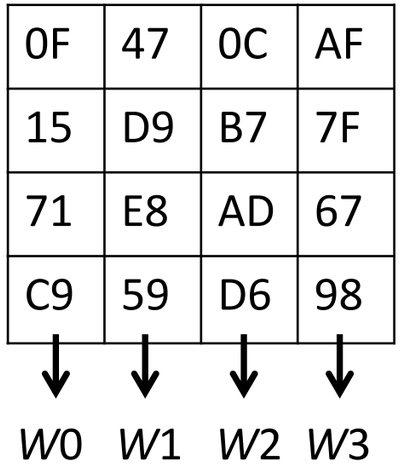
*W*4 *W*5 *W*6 *W*7

01	02	04	08	10	20	10	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

1. Новий стандарт шифрування AES

Розширення ключа

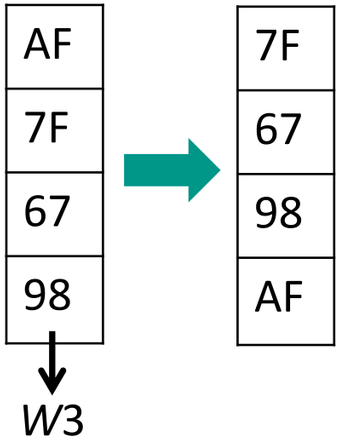
1. Перші *Nk* 4-байтових слів $W[i]$ послідовно вибираються з ключа шифру: 0-е слово – перші чотири байти, 1-е слово – другі чотири байти і т.д



2. Якщо i кратне *Nk:*

2. 1. У слові $W[i-1]$ виконують циклічний зсув байтів за схемою:

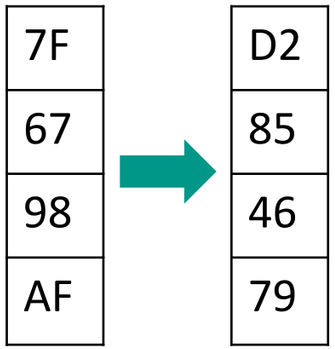
$(a, b, c, d) \rightarrow (b, c, d, a)$ де a, b, c, d – байти



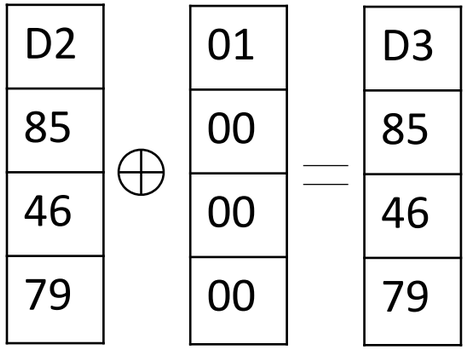
1. Новий стандарт шифрування AES

Розширення ключа

2.2. До кожного з 4-х байтів одержаного слова застосовують S-бокс

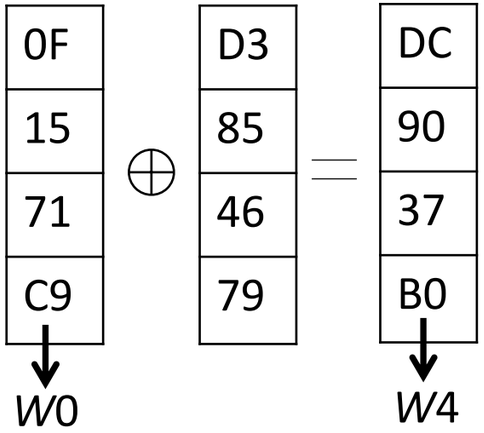


2.3. До результату додають раундову сталу Rcon за модулем 2



3. Решту слів $W[i]$ визначають за формулою:

$$W[i] = W[i - Nk] + W[i - 1](mod2)$$



1. Новий стандарт шифрування AES

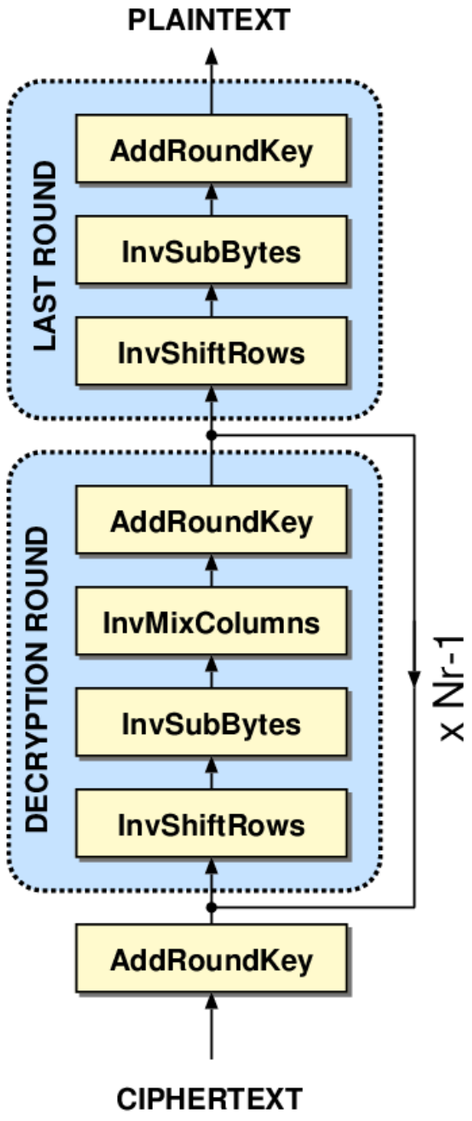
Дешифрування за алгоритмом AES

I. Перед першим раундом дешифрування виконується операція додавання з ключем

II. *Nr-1* раундів, кожен з яких кожен з яких здійснює такі операції:

1. Зсув рядків в зворотному порядку ;
2. Обернена операція до операції підстановки байтів;
3. Процедура, зворотна процедурі перемішування стовпців;
4. Додавання раундового ключа

III. Завершальний раунд *Nr*, в якому пропускається перемішування стовпців



1. Новий стандарт шифрування AES

Зсув рядків в зворотному порядку

Байти в останніх трьох рядках матриці зсуваються циклічно вліво на різне число байт

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

1. Новий стандарт шифрування AES

Обернена операція до операції підстановки байтів

Байти матриці замінюються новими значеннями за таблицею зворотної заміни, що є інвертованим *S*-боксом

	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

1. Новий стандарт шифрування AES

Процедура, зворотна процедурі перемішування стовпців

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

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

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

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

або у матричному вигляді:

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

1. Новий стандарт шифрування AES

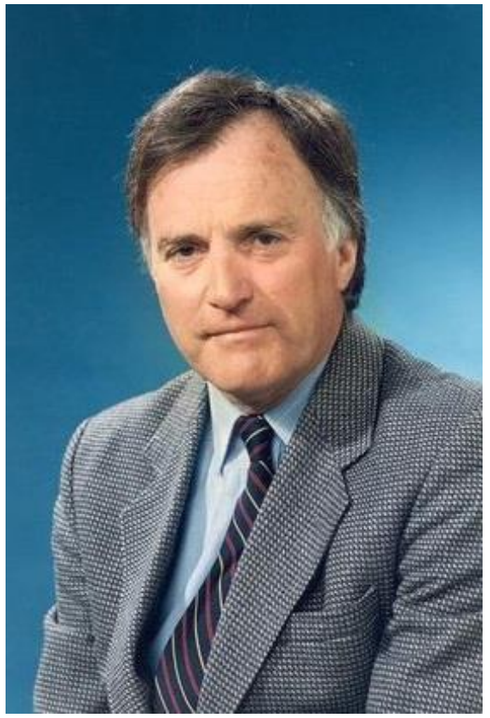
AES Visualization

2. Міжнародний стандарт шифрування IDEA

Розробники алгоритму Ксуеджа Лай і Джеймс Месі зі Швейцарського інституту ETH Zurich

Початковий варіант алгоритму IDEA з'явився у 1990 році та був запропонований на заміну DES

Ксуеджа Лай



Джеймс Месі

2. Міжнародний стандарт шифрування IDEA

Процес шифрування складається з 8 раундів, з яких випливає завершальне перетворення

Блок початкового тексту та шифротексту має довжину 64 біти

Ключ: блок довжиною 128 бітів

Фундаментальним нововведенням в алгоритмі є використання трьох операцій:

- $\boxplus$ додавання за модулем 2^{16} ;
- $\odot$ множення за модулем $2^{16} + 1$;
- $\oplus$ побітове виключне АБО (XOR)

2. Міжнародний стандарт шифрування IDEA

Шифрування за алгоритмом IDEA

1. Початковий 64-бітний блок ділиться на чотири 16-бітних підблоки: D_1, D_2, D_3, D_4

2. На кожному раунді чотири підблоки піддаються операціям XOR, додаванню і множенню один з одним та із шістьома 16-бітовими раундовими підключами

3. Між раундами обмінюються місцями другий і третій підблоки

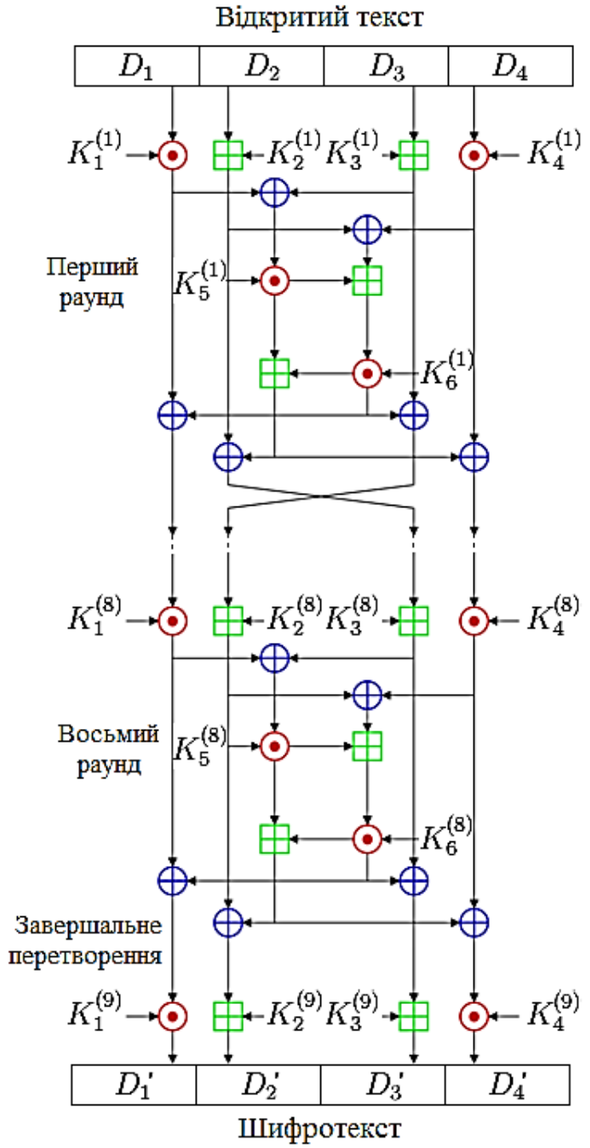
4. В завершальному перетворенні чотири підблоки поєднуються із чотирма 16-бітовими підключами (другий і третій блоки не міняються місцями)

5. Після виконання усіх перетворень конкатенація підблоків D_1', D_2', D_3' та D_4' являє собою зашифрований текст

2. Міжнародний стандарт шифрування IDEA

Етапи шифрування за алгоритмом IDEA

1.Перемножуються D_1 і перший підключ K_1.
2.Додаються D_2 і другий підключ K_2.
3.Додаються D_3 і третій підключ K_3.
4.Перемножуються D_4 і четвертий підключ K_4.
5.Виконується XOR над результатами етапів (1) і (3).
6.Виконується XOR над результатами етапів (2) і (4).
7.Перемножуються результати етапу (5) і п'ятий підключ K_5.
8.Додаються результати етапів (6) і (7).
9.Перемножуються результати етапу (8) і шостий підключ K_6.
10.Додаються результати етапів (7) і (9).
11.Виконується XOR над результатами етапів (1) і (9).
12.Виконується XOR над результатами етапів (3) і (9).
13.Виконується XOR над результатами етапів (1) і (10).
14.Виконується XOR над результатами етапів (4) і (10).



2. Міжнародний стандарт шифрування IDEA

Генерація підключів

1. Початковий 128-бітний ключ розбивається на вісім підключів по 16 біт кожен

$$(K_1^{(1)} K_2^{(1)} K_3^{(1)} K_4^{(1)} K_5^{(1)} K_6^{(1)} K_1^{(2)} K_2^{(2)})$$

2. Далі початковий 128-бітний ключ циклічно зсувається ліворуч на 25 позицій, після чого новий 128-бітний блок знову розбивається на вісім 16-бітних підключів

$$(K_3^{(2)} K_4^{(2)} K_5^{(2)} K_6^{(2)} K_1^{(3)} K_2^{(3)} K_3^{(3)} K_4^{(3)})$$

3. Процедура циклічного зсуву і розбиття на блоки продовжується до тих пір, поки не будуть згенеровані всі 52 16-бітних підключі

Номер раунду	Підключі
1	$K_1^{(1)} K_2^{(1)} K_3^{(1)} K_4^{(1)} K_5^{(1)} K_6^{(1)}$
2	$K_1^{(2)} K_2^{(2)} K_3^{(2)} K_4^{(2)} K_5^{(2)} K_6^{(2)}$
3	$K_1^{(3)} K_2^{(3)} K_3^{(3)} K_4^{(3)} K_5^{(3)} K_6^{(3)}$
4	$K_1^{(4)} K_2^{(4)} K_3^{(4)} K_4^{(4)} K_5^{(4)} K_6^{(4)}$
5	$K_1^{(5)} K_2^{(5)} K_3^{(5)} K_4^{(5)} K_5^{(5)} K_6^{(5)}$
6	$K_1^{(6)} K_2^{(6)} K_3^{(6)} K_4^{(6)} K_5^{(6)} K_6^{(6)}$
7	$K_1^{(7)} K_2^{(7)} K_3^{(7)} K_4^{(7)} K_5^{(7)} K_6^{(7)}$
8	$K_1^{(8)} K_2^{(8)} K_3^{(8)} K_4^{(8)} K_5^{(8)} K_6^{(8)}$
Завершальне перетворення	$K_1^{(9)} K_2^{(9)} K_3^{(9)} K_4^{(9)}$

2. Міжнародний стандарт шифрування IDEA

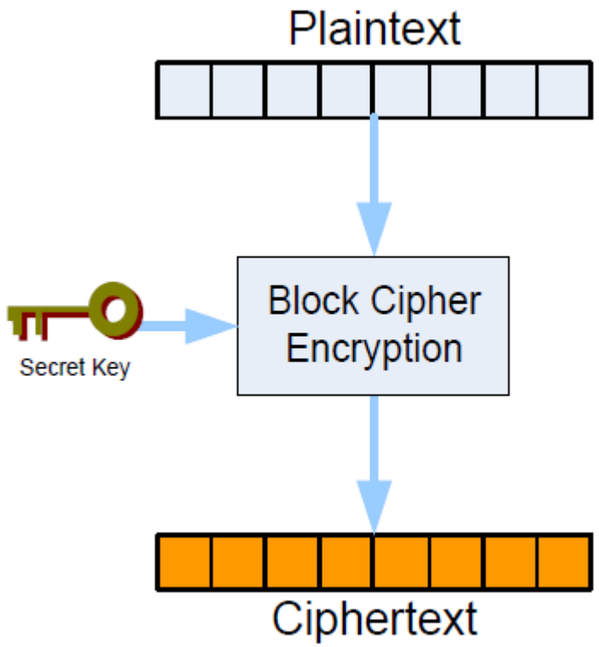
Дешифрування за алгоритмом IDEA

Аналогічно до процесу шифрування, але для дешифрування використовуються інші підключі в зворотному порядку

Номер раунду	Підключі
1	$1/K_1^{(9)}$ $-K_2^{(9)}$ $-K_3^{(9)}$ $1/K_4^{(9)}$ $K_5^{(8)}$ $K_6^{(8)}$
2	$1/K_1^{(8)}$ $-K_3^{(8)}$ $-K_2^{(8)}$ $1/K_4^{(8)}$ $K_5^{(7)}$ $K_6^{(7)}$
3	$1/K_1^{(7)}$ $-K_3^{(7)}$ $-K_2^{(7)}$ $1/K_4^{(7)}$ $K_5^{(6)}$ $K_6^{(6)}$
4	$1/K_1^{(6)}$ $-K_3^{(6)}$ $-K_2^{(6)}$ $1/K_4^{(6)}$ $K_5^{(5)}$ $K_6^{(5)}$
5	$1/K_1^{(5)}$ $-K_3^{(5)}$ $-K_2^{(5)}$ $1/K_4^{(5)}$ $K_5^{(4)}$ $K_6^{(4)}$
6	$1/K_1^{(4)}$ $-K_3^{(4)}$ $-K_2^{(4)}$ $1/K_4^{(4)}$ $K_5^{(3)}$ $K_6^{(3)}$
7	$1/K_1^{(3)}$ $-K_3^{(3)}$ $-K_2^{(3)}$ $1/K_4^{(3)}$ $K_5^{(2)}$ $K_6^{(2)}$
8	$1/K_1^{(2)}$ $-K_3^{(2)}$ $-K_2^{(2)}$ $1/K_4^{(2)}$ $K_5^{(1)}$ $K_6^{(1)}$
Завершальне перетворення	$1/K_1^{(1)}$ $-K_2^{(1)}$ $-K_3^{(1)}$ $1/K_4^{(1)}$

3. Режими шифрування

Режим шифрування – метод застосування блочного шифру, що дозволяє перетворити послідовність блоків відкритих даних у послідовність блоків зашифрованих даних

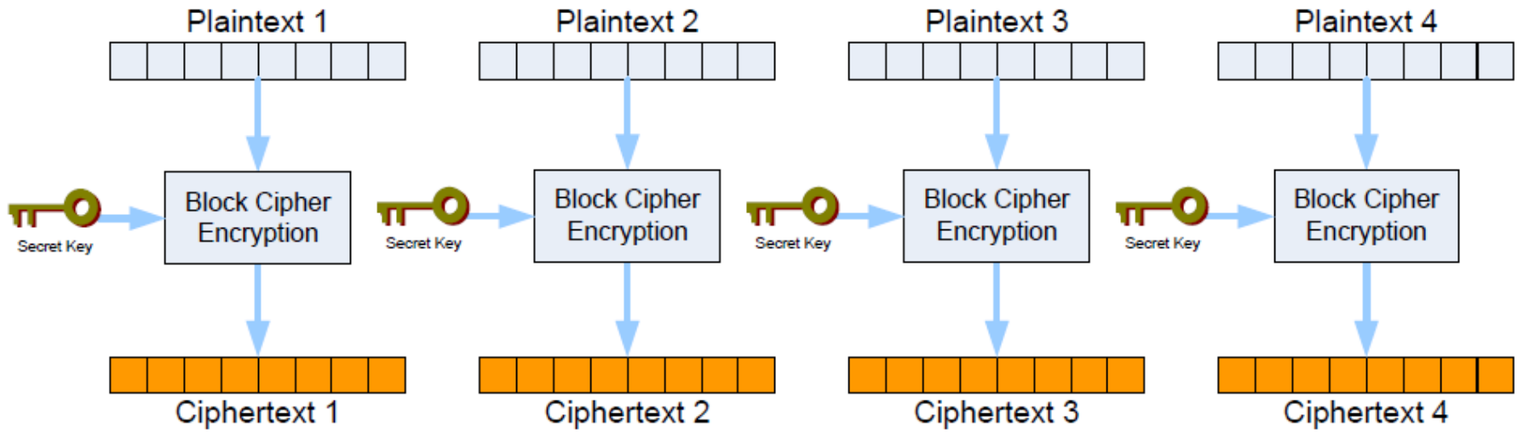


3. Режими шифрування

Режим простої заміни (ECB, Electronic Coding Book)

1. Повідомлення ділиться на блоки P_i однакового розміру по n біт
2. Кожен блок P_i шифрується окремо та незалежно від інших блоків алгоритмом E_k та ключем k

$$C_i=E_k(P_i)$$



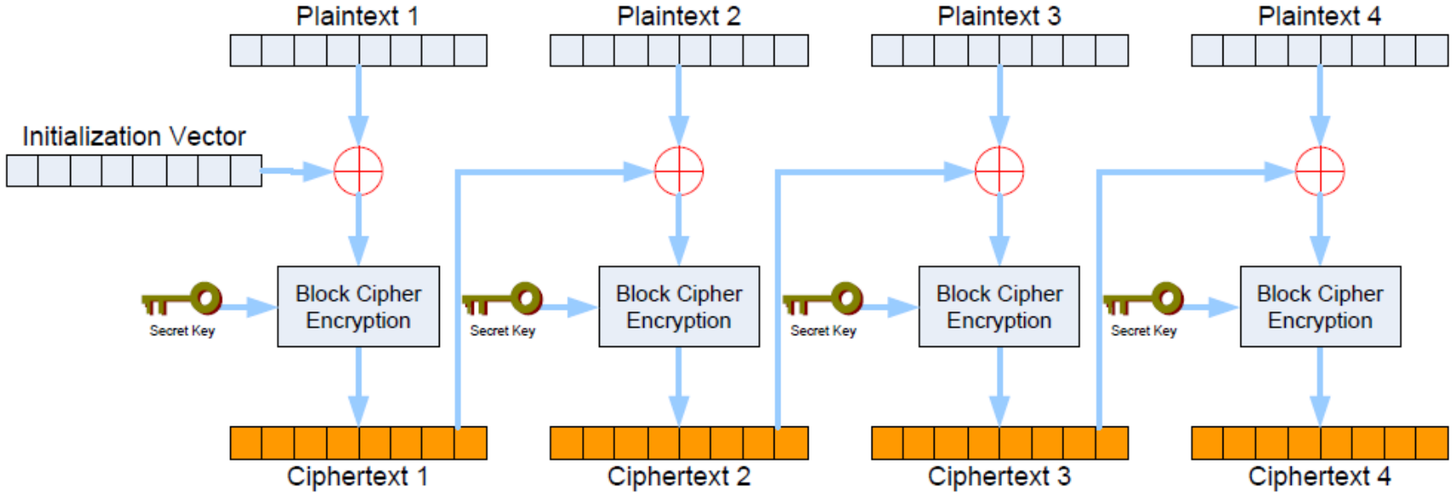
3. Режими шифрування

Режим зв’язування блоків (CBC, Cipher Block Chaining)

1. Повідомлення ділиться на блоки P_i однакового розміру по n біт
2. Кожен блок P_i додається за модулем 2 з попередньо зашифрованим блоком C_{i-1}, а потім результат зашифровується.

Для шифрування P_1 використовують вектор ініціалізації IV (Initialization Vector) – послідовність випадкових символів розміром n

$$C_i=E_k(P_i \oplus C_{i-1}),$$
$$C_0=IV$$

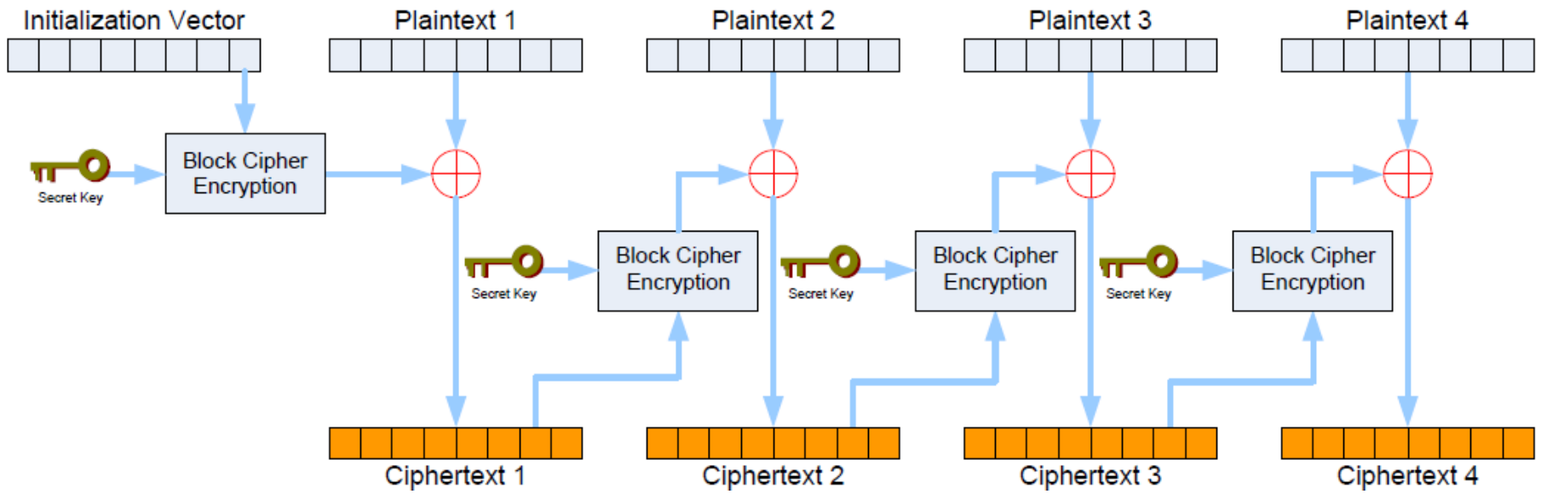


3. Режими шифрування

Режим зі зворотнім зв’язком по шифротексту (CFB, Cipher Feedback)

1. Повідомлення ділиться на блоки P_i однакового розміру по n біт
2. Попередньо зашифрований блок C_{i-1} шифрується ще раз і додається за модулем 2 з P_i.

$$C_i=P_i \oplus E_k(C_{i-1}),$$
$$C_0=IV$$

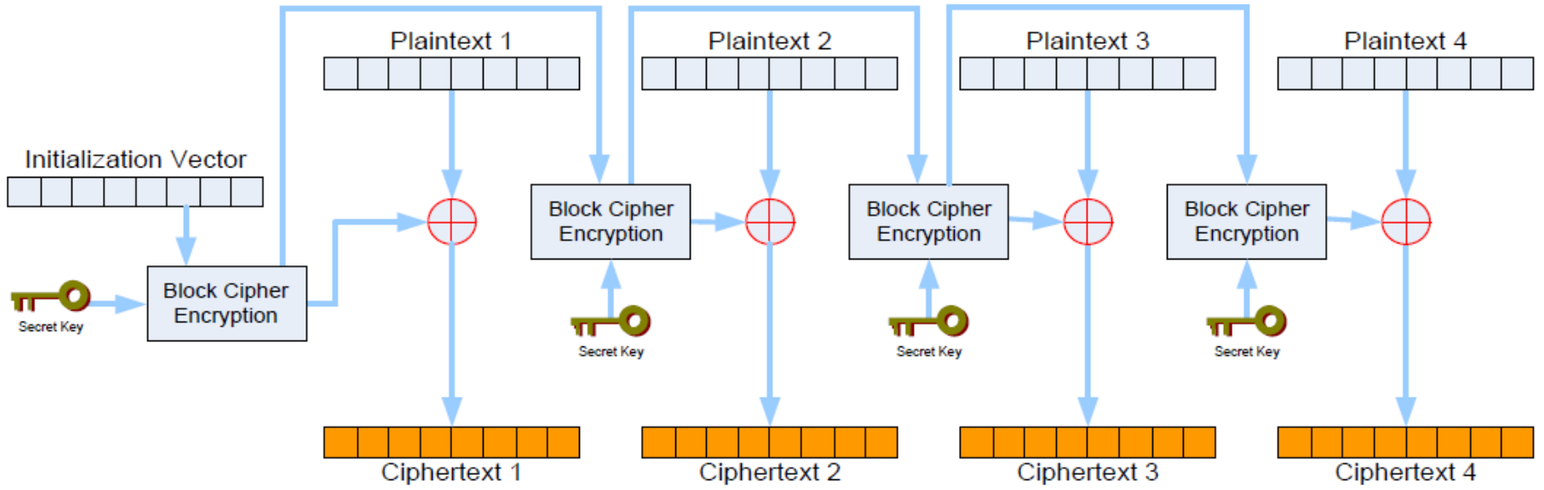


3. Режими шифрування

Режим зі зворотнім зв’язком по виходу (OFB, Output Feedback)

Опрацювати самостійно

$$C_i = P_i \oplus O_{i-1},$$
$$O_i = E_k(O_{i-1})$$
$$O_0 = IV$$

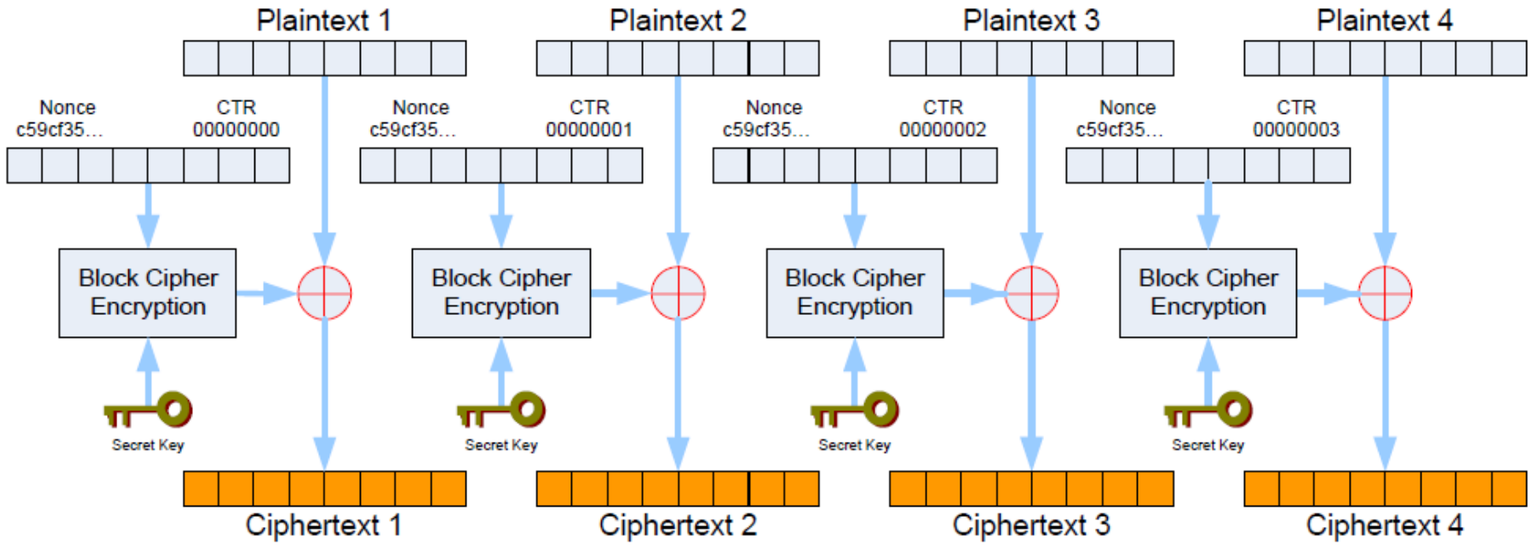


3. Режими шифрування

Режим лічильника (CTR, Counter Mode)

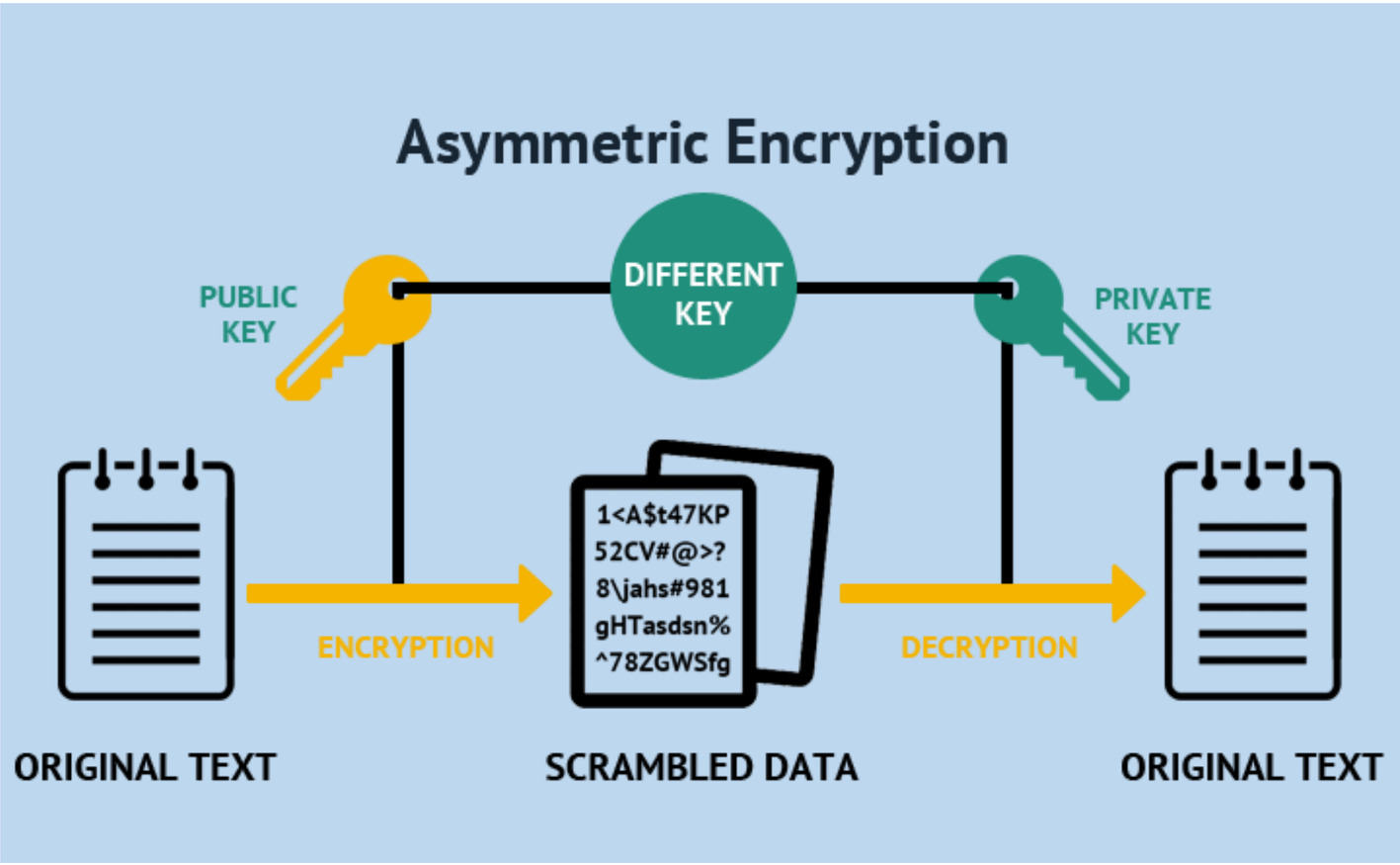
Опрацювати самостійно

$$C_i = P_i \oplus E_k(Ctr_i),$$
$$Ctr_{i-1} = IV \mid Nonce$$



ЛЕКЦІЯ 5

Асиметричні алгоритми шифрування



План

1. Ідея криптосистеми з відкритим ключем
2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)
3. Алгоритм RSA
4. Алгоритм Ель-Гамаля

1. Ідея криптосистеми з відкритим ключем

Ідея криптосистеми з відкритим ключем була висунута американськими криптографами Уітфілдом Діффі та Мартіном Хелманом (1976 рік), і окремо Ральфом Мерклом (1978 рік)

У асиметричних криптосистемах для шифрування використовується відкритий ключ (публічний), а для дешифрування – закритий (приватний)

Ральф Меркл | Мартін Хелман | Уітфілд Діффі

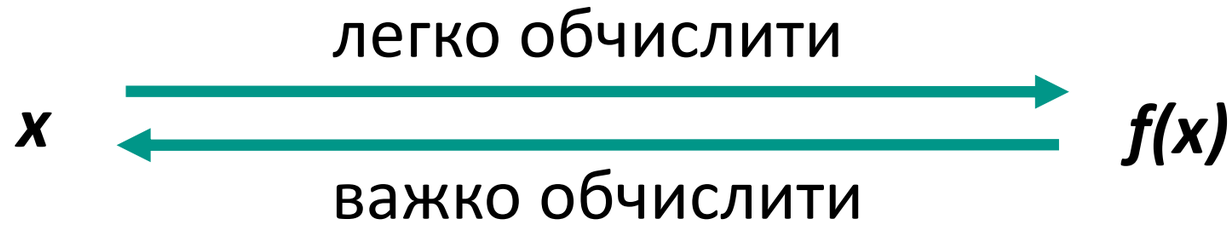
1. Ідея криптосистеми з відкритим ключем

How asymmetric (public key) encryption works

1. Ідея криптосистеми з відкритим ключем

Математична база

Ідея криптографії з відкритим ключем 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



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

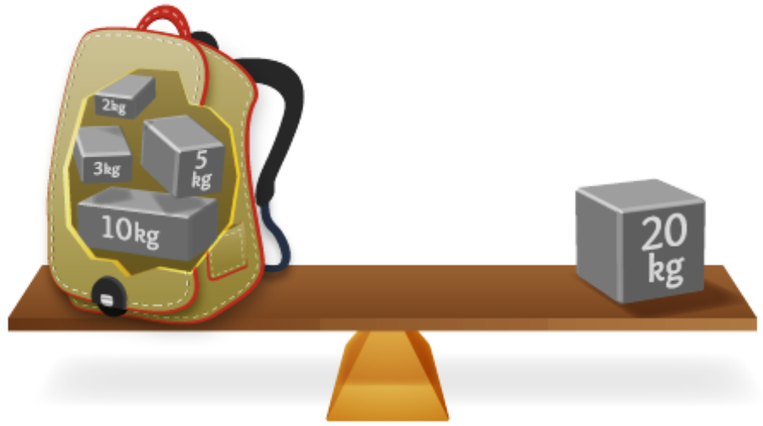
2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Проблема рюкзака

Дано набір предметів різної маси. Чи можна покласти деякі із цих предметів у рюкзак так, щоб маса рюкзака дорівнювала певному значенню?

Проблема: за вагою рюкзака визначити, які предмети поклали, а які ні

Наприклад, маси предметів 1, 5, 6, 11, 14 і 20. Можна спакувати рюкзак так, що його маса дорівнюватиме 22, використавши маси 5, 6 і 11. Неможливо спакувати рюкзак так, щоб його маса дорівнювала 24



2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Ідея шифрування повідомлення як розв'язання проблеми рюкзака

Дано набір значень $M_1, M_2,\ldots, M_n$ і сума S, обчислити значення b_i, такі що $S = M_1 b_1 + M_2 b_2 + \ldots + M_n b_n$, $b_i \in \{0,1\}$

$M_1, M_2,\ldots, M_n$ – рюкзак;
$b_1, b_2,\ldots, b_n$ – відкритий текст;
S – шифротекст

Приклад 2.1:

Відкритий текст	111001	010110	000000	011000
Рюкзак	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20
Шифротекст	1+5+6+20=32	5+11+14=30	0=0	5+6=11

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Проблема рюкзака

Легка

Якщо перелік мас предметів являє собою суперзростаючу послідовність, то проблему рюкзака легко розв’язати

Складна

Якщо перелік мас предметів являє собою нормальну послідовність, то проблему рюкзака розв’язати важко

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Суперзростаюча послідовність – це послідовність, у якій кожний елемент більший за суму усіх попередніх елементів

Нормальна послідовність – це послідовність, що містить довільні елементи

Наприклад, послідовність {1, 3, 6, 13, 27, 52} є суперзростаючою

Наприклад, послідовність {1, 3, 4, 9, 15, 25} не є суперзростаючою, тобто вона нормальна

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Алгоритм розв'язання проблеми суперзростаючого рюкзака

1. Повну вагу рюкзака порівнюємо з найбільшим числом послідовності

2. Якщо повна вага менша за це число, то його не кладемо у рюкзак

3. Якщо повна вага більша або дорівнює цьому числу, то воно кладеться у рюкзак. Зменшуємо масу рюкзака на це значення.

4. Переходимо до наступного по величині числа послідовності

5. Будемо повторювати, поки процес не закінчиться. Якщо повна вага зменшиться до нуля, то розв’язок знайдений

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.2:

Повна вага рюкзака – 70, послідовність мас {2, 3, 6, 13, 27, 52}

1. Найбільша маса – 52 < 70 ⇨ кладемо 52 у рюкзак.
2. Віднімаємо: 70-52=18.
3. Наступна маса – 27 > 18 ⇨ 27 у рюкзак не кладемо.
4. Вага 13 < 18 ⇨ кладемо 13 у рюкзак.
5. Віднімаємо: 18-13=5.
6. Наступна маса – 6 > 5 ⇨ 6 не кладемо у рюкзак.

Продовження цього процесу покаже, що й 2, і 3 кладемо у рюкзак, і повна вага зменшується до 0, що повідомляє про знайдений розв’язок.

Відкритий текст: 110101

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Криптосистема Меркла-Хелмана

Закритий ключ – суперзростаюча послідовність

Відкритий ключ – нормальна послідовність

Генерування відкритого ключа із закритого

1. Генерується суперзростаюча послідовність

2. Обирається число *m* (модуль), більше за суму усіх чисел послідовності

3. Знаходиться *n* взаємно просте з *m*

4. Усі значення суперзростаючої послідовності множаться по модулю m на число n

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.3:

Дано: закритий ключ – супер зростаюча послідовність {2, 3, 6, 13, 27, 52}, m = 105, n = 31

Нормальною послідовністю буде:
2·31 mod 105 = 62
3·31 mod 105 = 93
6·31 mod 105 = 81
13·31 mod 105 = 88
27·31 mod 105 = 102
52·31 mod 105 = 37

Відкритий ключ – {62, 93, 81, 88, 102, 37}

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Шифрування у криптосистемі Меркла-Хелмана

1. Розбити повідомлення на блоки, рівні по довжині кількості елементів послідовності рюкзака

2. Вважати, що у відкритому тексті одиниця вказує на присутність члена послідовності, а нуль – на його відсутність

3. Обчислити повні маси рюкзака – по одному для кожного блоку повідомлення

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.4:

Дано: повідомлення в бінарному виді 011000110101101110,
відкритий ключ – послідовність {62, 93, 81, 88, 102, 37}

Шифруємо: повідомлення = 011000 110101 101110
011000 відповідає 93 + 81 = 174
110101 відповідає 62 + 93 + 88 + 37 = 280
101110 відповідає 62 + 81 + 88 + 102 = 333

Шифротекст: послідовність {174, 280, 333}

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Дешифрування у криптосистемі Меркла-Хелмана

1. Спочатку визначають n^{-1}, таке що n (n^{-1}) ≡ 1 (mod m)

2. Кожне значення шифротексту множиться на n^{-1}mod m

3. Одержати значення відкритого тексту за допомогою закритого ключа – одиниця вказує на присутність члена послідовності, а нуль – на його відсутність

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.5:

Дано: шифротекст {174,280,333}, закритий ключ – {2,3,6,13,27,52},
m =105, n = 31

Дешифруємо:

У нашому випадку n^{-1} дорівнює 61, тому значення шифротекста помножемо на 61 mod 105.

174 · 61 mod 105 = 9 = 3 + 6, що відповідає 011000

280 · 61 mod 105 = 70 = 2 + 3 + 13 + 52, що відповідає 110101

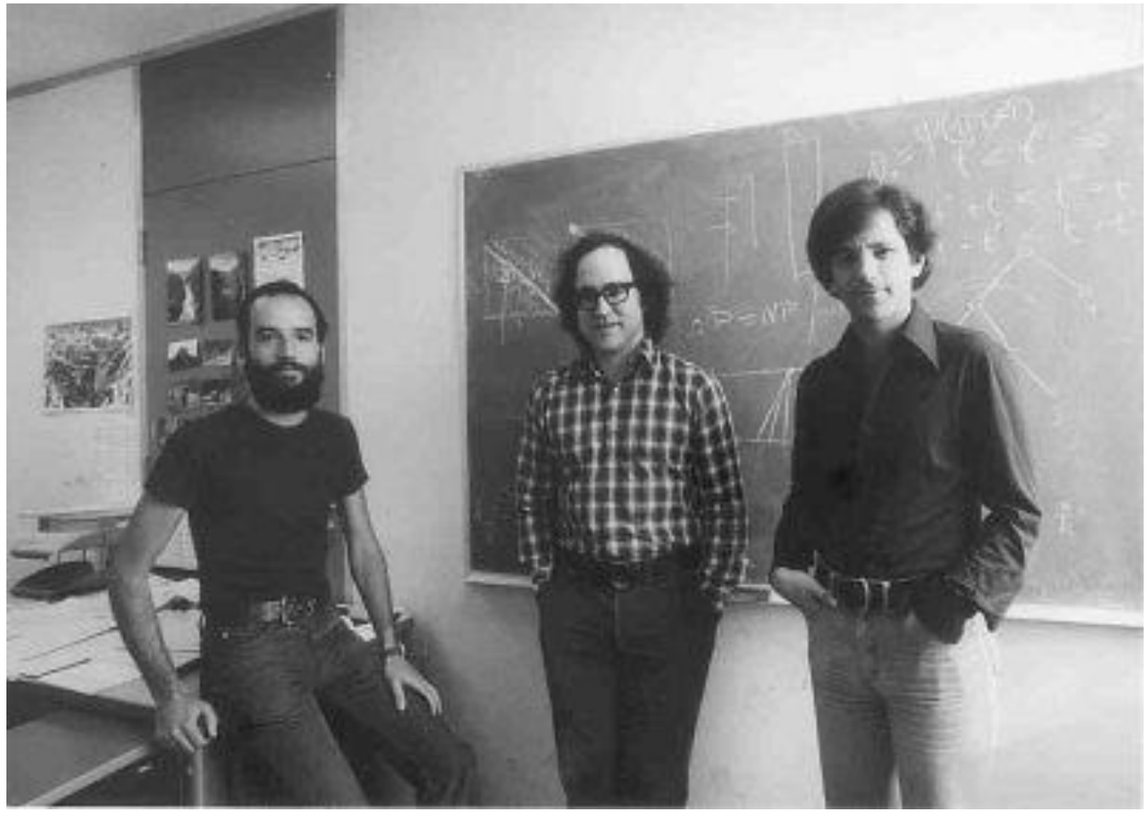
333 · 61 mod 105 = 48 = 2 + 6 + 13 + 27, що відповідає 101110

Відкритий текст: 011000 110101 101110

3. Алгоритм RSA

Автори криптоалгоритму RSA (Rivest-Shamir-Adleman) – Рон Рівест, Аді Шамір і Леонард Едлман (1977 рік)

Безпека RSA заснована на складності розкладання на множники великих чисел



Аді Шамір | Рон Рівест | Леонард Едлман

3. Алгоритм RSA

Генерація ключів

1. Вибираються два великих випадкових простих числа p і q
2. Обчислюється добуток: $n=pq$
3. Обчислюється функція Ейлера $\varphi(n) = \varphi(pq) = (p-1)(q-1)$
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 $ed \equiv 1 \pmod{\varphi(n)}$
6. Пара (e, n) публікується у якості відкритого ключа
7. Пара (d, n) виконує роль закритого ключа і тримається таємниці

3. Алгоритм RSA

Шифрування:

повідомлення m розбивається на цифрові блоки, менші n;

кожен блок повідомлення m_i зашифровують за формулою:

$$c_i = m_i^e \bmod n$$

Дешифрування:

для кожного зашифрованого блоку c_i обчислюють:

$$m_i = c_i^d \bmod n$$

3. Алгоритм RSA

Приклад 3.1 (генерація ключів):
Дано: повідомлення КНИГА, що складається із символів українського алфавіту та представляється як послідовність цілих чисел
m = 14 17 10 3 0

1. Оберемо $p = 3$ і $q = 11$, тоді $n = p \cdot q = 3 \cdot 11 = 33$.
2. Обчислимо $\varphi(33) = 2 \cdot 10 = 20$.
3. Виберемо (випадково) e рівним 3 та перевіримо виконання умов: $1<3<20$, НСД(3, 20) =1.
4. Визначимо d – ключ дешифрування з рівняння $3d \equiv 1 \pmod{20}$. Для розв’язання рівняння використаємо розширений алгоритм Евкліда та зайдемо $d = 7$.

3. Алгоритм RSA

Приклад 3.1 (шифрування):

Отже відкритий ключ (*e, n*) = (3, 33), закритий ключ (*d, n*) = (7, 33)

Зашифруємо повідомлення *m* = 14 17 10 3 0, що складається із п’яти блоків m_i та отримаємо шифротекст *c* = 5 29 10 27 0

$$m_1 = 14^3 \bmod 33 = ((14^2 \bmod 33) \cdot (14^1 \bmod 33)) \bmod 33 = (31 \cdot 14) \bmod 33 = 434 \bmod 33 = 5 = c_1$$

$$m_2 = 17^3 \bmod 33 = ((17^2 \bmod 33) \cdot (17^1 \bmod 33)) \bmod 33 = (25 \cdot 17) \bmod 33 = 425 \bmod 33 = 29 = c_2$$

$$m_3 = 10^3 \bmod 33 = 1000 \bmod 33 = 10 = c_3$$

$$m_4 = 3^3 \bmod 33 = 27 \bmod 33 = 27 = c_4$$

$$m_5 = 0^3 \bmod 33 = 0 \bmod 33 = 0 = c_5$$

3. Алгоритм RSA

Приклад 3.1 (дешифрування):

Для дешифрування потрібно також виконати піднесення до степеню, використовуючи ключ дешифрування 7.

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

$$c_1 = 5^7 \bmod 33 = ((5^4 \bmod 33)\cdot(5^3 \bmod 33)) \bmod 33 = (31\cdot 26) \bmod 33 = 806 \bmod 33 = 14 = m_1$$

$$c_2 = 29^7 \bmod 33 = ((29^4 \bmod 33)\cdot(29^3 \bmod 33)) \bmod 33 = (((29^2))^2 \bmod 33)\cdot(29^2 \bmod 33)\cdot(29 \bmod 33)) \bmod 33 =$$
$$= (25\cdot 16\cdot 29) \bmod 33 = 11600 \bmod 33 = 17 = m_2$$

$$c_3 = 10^7 \bmod 33 = ((10^4 \bmod 33)\cdot(10^3 \bmod 33)) \bmod 33 = (((10^2))^2 \bmod 33)\cdot(10^2 \bmod 33)\cdot(10 \bmod 33)) \bmod 33 =$$
$$= (1\cdot 1\cdot 10) \bmod 33 = 10 \bmod 33 = 10 = m_3$$

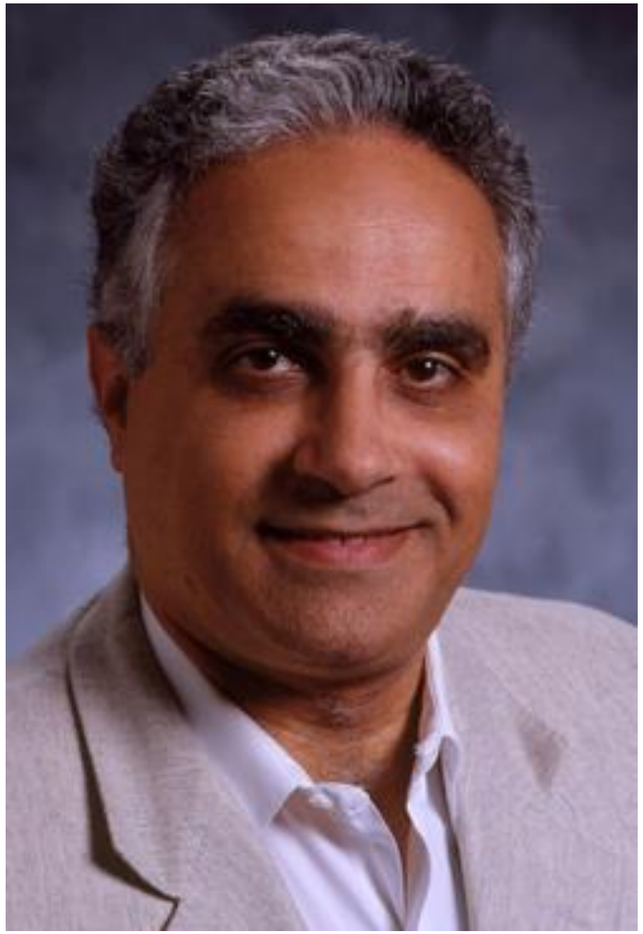
$$c_4 = 27^7 \bmod 33 = ((27^4 \bmod 33)\cdot(27^3 \bmod 33)) \bmod 33 = (((27^2))^2 \bmod 33)\cdot(27^2 \bmod 33)\cdot(27 \bmod 33)) \bmod 33 =$$
$$= (9\cdot 3\cdot 27) \bmod 33 = 729 \bmod 33 = 3 = m_4$$

$$c_5 = 0^7 \bmod 33 = 0 \bmod 33 = 0 = m_5$$

4. Алгоритм Ель-Гамаля

Автор – американський вчений єгипетського походження
Тахер Ель-Гамаль
(1985 рік)

Безпека алгоритму заснована на складності обчислення дискретних логарифмів у скінченному полі



Тахер Ель-Гамаль

4. Алгоритм Ель-Гамаля

Генерація ключів

1. Генерується просте випадкове число p
2. Вибирається випадкове число $g < p$
3. Вибирається випадкове число x, таке що $1 < x < p - 1$
4. Обчислюється $y = g^x \bmod p$
5. Відкритим ключем є трійка (y, g і p)
6. Закритим ключем є x

4. Алгоритм Ель-Гамаля

Шифрування:

вибирається сесійний ключ – випадкове число k, таке що $1 < k < p - 1$;

потім обчислюються

$$a = g^k \bmod p$$

$$b = y^k M \bmod p$$

Пара чисел (a, b) є шифротекстом

Дешифрування:

для дешифрування (a, b) обчислюється

$$M = b(a^x)^{-1} \bmod p$$

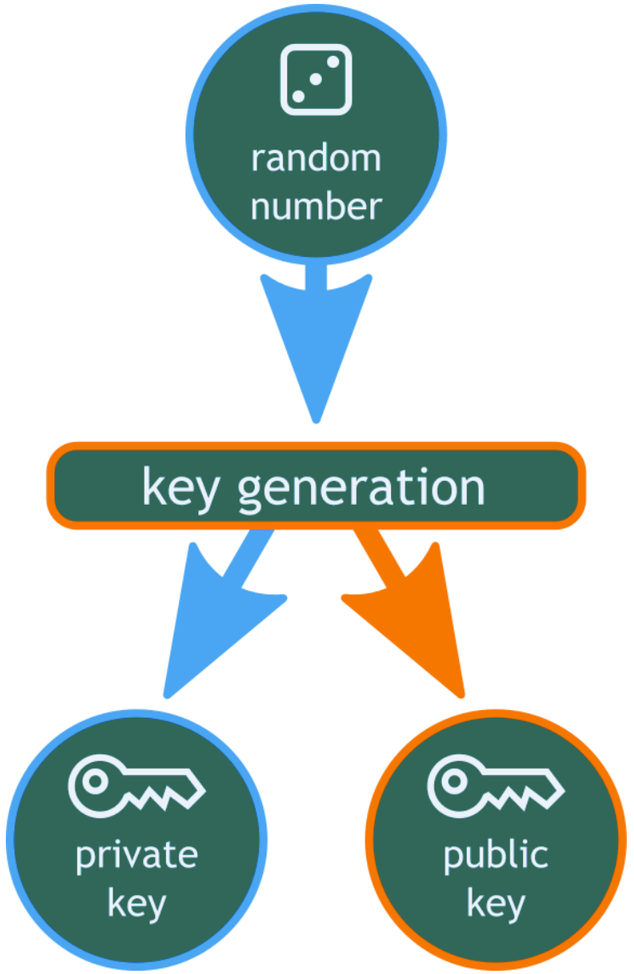
або

$$M = b(a^x)^{-1} \bmod p = b \cdot a^{(p-1-x)} \bmod p$$

4. Алгоритм Ель-Гамаля

Приклад 4.1 (генерація ключів):
Дано: повідомлення $M = 5$

1. Нехай $p = 11$, $g = 2$.
2. Виберемо $x = 8$ – випадкове ціле число x таке, що таке що $1 < x < p - 1$.
3. Обчислимо $y = g^x \bmod p = 2^8 \bmod 11 = 3$.
4. Отже, відкритим ключем є трійка $(p, g, y) = (11,2,3)$, закритим ключем є число $x = 8$.



4. Алгоритм Ель-Гамаля

Приклад 4.1 (шифрування):

Вибираємо випадкове ціле число k таке, що $1 < k < p - 1$. Нехай k = 9.

Обчислюємо число

$$a = g^k \bmod p = 2^9 \bmod 11 =$$

$$= 512 \bmod 11 = 6.$$

Обчислюємо число

$$b = y^k M \bmod p = 3^9\, 5 \bmod 11 =$$

$$= 19683^*\, 5 \bmod 11 = 9.$$

Пара $(a, b) = (6,9)$ є шифротекстом.

Приклад 4.1 (дешифрування):

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

Обчислюємо M за формулою:

$$M = b\,((a^x)^{-1}) \bmod p =$$

$$= 9\,((6^8)^{-1}) \bmod 11 = 5$$

Отримали вихідне повідомлення

$$M = 5.$$

Симетричні шифри vs. асиметричні шифри

Характеристика	Симетричні шифри	Асиметричні шифри
Ключ	Один і той самий ключ використовується для шифрування та дешифрування	Один ключ (відкритий) використовується для шифрування, інший (закритий) – для дешифрування
Обмін ключами	Потрібен секретний канал для передачі ключа або інший надійний механізм обміну ключами	Відкритий ключ доступний всім, але його справжність має перевірятися центром сертифікації ключів
Математична складність	Відносно прості математичні операції	Складні математичні обчислення
Швидкість роботи	Висока	Низька
Криптографічна стійкість	Задовільна	Достатня
Вид захисту	Конфіденційність	Конфіденційність, цілісність, автентичність, невідмовність

ЛЕКЦІЯ 6

Електронний цифровий підпис

План

1. Поняття електронного цифрового підпису
2. Хеш-функції
3. Процедури підписування та перевірки ЕЦП
4. Стандарт цифрового підпису DSS
5. Схеми цифрового підпису RSA та El Gamal

1. Поняття електронного цифрового підпису

Ідею «електронного цифрового підпису» вперше описали творці криптографії з відкритим ключем Уітфілд Діффі та Мартін Хелман (1976 рік)

Електронний цифровий підпис (ЕЦП) – 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 (Закон України «Про ЕЦП»)

1. Поняття електронного цифрового підпису

Призначення ЕЦП

Контроль цілісності документа	при будь-якій випадковій або навмисній зміні документа підпис стане недійсним, тому що обчислений він на підставі початкового стану документа та відповідає лише йому
Захист від зміни (підробки) документа	гарантія виявлення підробки у процесі контролю цілісності робить підробку недоцільною у більшості випадків
Неможливість відмови від авторства	власник підпису під документом не може відмовитися від нього, оскільки можна довести, що підпис був створений закритим ключем, який відомий тільки власнику ключа (автору документа)
Доказове підтвердження авторства документа	знаючи закритий ключ, власник (автор документа) може однозначно довести своє авторство підпису під документом

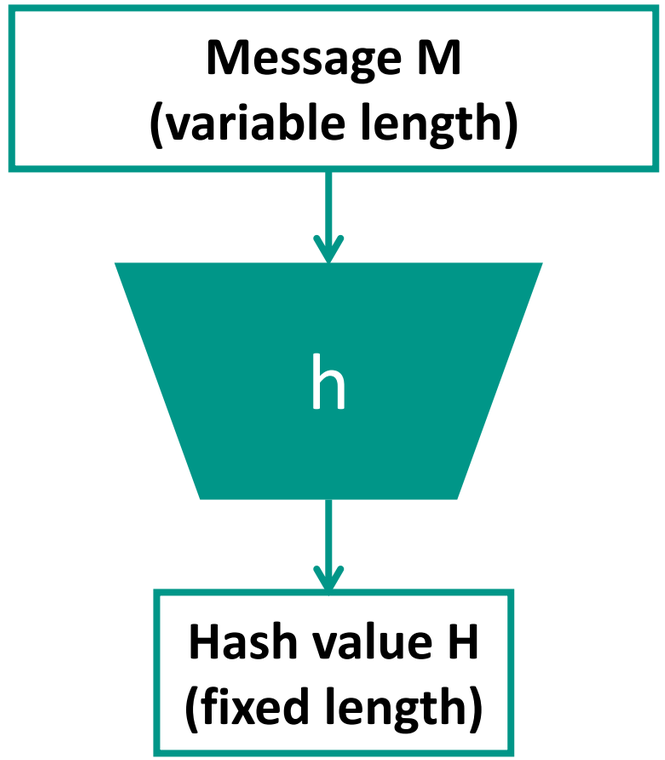
2. Хеш-функції

Хеш-функція являє собою функцію, математичну або іншу, що отримує на вхід рядок змінної довжини і перетворює його в рядок фіксованої, зазвичай меншої, довжини

Результат хеш-функції називають хешем, хеш-значенням, контрольною сумою або дайджестом

2. Хеш-функції

Хеш-функція *h*, яка використовується у протоколі ЕЦП, призначена для того, щоб стиснути підписуваний документ *M* довільної довжини до двійкового хеш-значення *H* фіксованої довжини



Зміна хоча б одного біту у початковому повідомленні призводить до зміни у дайджесті, що формується на його основі

2. Хеш-функції

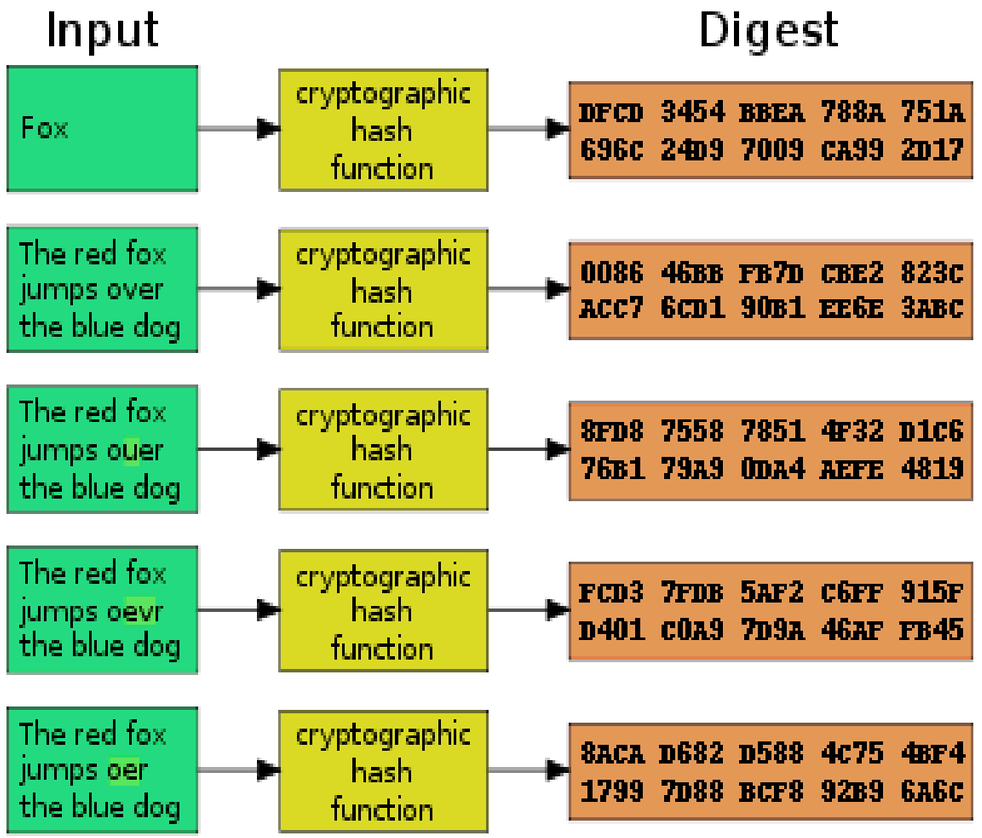
Основні властивості хеш-функції:

1) хеш-значення *H* залежить від усього документу *M* надзвичайно складним чином, завдяки чому за значенням *H* неможливо відновити документ *M*;

2) хеш-значення *H* є чутливим до будь-яких, навіть незначних, змін у документі *M*;

3) хеш-функція *h* є незворотною, тобто підбір деякого фіктивного документу *M* з таким самим хеш-значенням *H* є задачею практично нерозв'язуваною;

4) ймовірність співпадіння хеш-значень двох різних документів с надзвичайно малою.



2. Хеш-функції

Приклад 2.1:
Хеш-функцій існує багато. Розглянемо роботу хеш-функції SHA-256 (Secure Hash Algorithm – безпечний алгоритм хешування), яка формує хеш у вигляді рядка з 64 символів (256 біт)
Спробуємо за допомогою SHA-256 hash калькулятора отримати хеш для назви теми лекції

Текст	Дайджест
Електронний цифровий підпис	ca794ad2e3bf0c741786cf9f3f6e9d20ecfb33527f12b7d5cb18cd64466e8347
Електронний цифровий підпис!	cf158480422fd52aff9a28ea1f879a8d136ef6f45011a47fa98f5ce636d3cdff

2. Хеш-функції

Принцип роботи хеш-функцій

Документ *M* має бути представлений у двійковій формі і розбитий на окремі блоки M_i довжиною *n* біт кожний

Більшість хеш-функцій мають вигляд $H_i = h(M_i, H_{i-1})$, де

M_i – черговий блок документу *M*;
H_{i-1} – хеш-значення усіх попередніх блоків документу (має довжину також n біт);

Правило утворення одного хеш-значення із двох вхідних аргументів залежить від типу хеш-функції.

У найпростішому випадку тут може використовуватись додавання за модулем 2, тобто $H_i=M_i \oplus H_{i-1}$

2. Хеш-функції

Принцип роботи хеш-функцій

Хеш-значення, обчислене при використанні останнього блоку документу, вважається хеш-значенням усього документу *M*

При обчисленні хеш-значення для першого блоку M_1 використовується деяке початкове хеш-значення H_0, яке можна вибрати випадковим або фіксованим (наприклад, $H_0 = 0$ – у найпростішому випадку)

Визначення кількості біт *n* для хеш-значення може здійснюватися на підставі значення модуля системи ЕЦП, а саме: кількість біт хеш-значення має бути на одиницю меншою кількості біт значення модуля системи ЕЦП

2. Хеш-функції

Приклад 2.2:

Відомо, що модуль системи ЕЦП $19_{10} = 10011_2$

Кількість біт цього значення рівна 5

Виберемо кількість біт хеш-значення $n = 4$

Припустимо, що необхідно підписати наступний документ

M: 29, 7, 11, 3, 20, 36

Необхідно знайти найпростіше хеш-значення для цього документу.

Утворюємо двійковий образ документу: 11101, 111, 1011, 11, 10100, 100100

Суцільна послідовність біт має вигляд: 111011111011111010100100100

Доповнюємо кількість біт цієї послідовності до числа, кратного $n = 4$, за рахунок початку цієї ж послідовності і отримуємо:

1110 1111 1011 1110 1001 0010 0111

2. Хеш-функції

Приклад 2.2 (продовження):

Таким чином, даний документ розбито на сім блоків довжиною 4 біт кожний. Тепер утворюємо послідовність хеш-значень:

$H_0 = 0000$

$H_1 = M_1 \oplus H_0 = 1110 \oplus 0000 = 1110$

$H_2 = M_2 \oplus H_1 = 1111 \oplus 1110 = 0001$

$H_3 = M_3 \oplus H_2 = 1011 \oplus 0001 = 1010$

$H_4 = M_4 \oplus H_3 = 1110 \oplus 1010 = 0100$

$H_5 = M_5 \oplus H_4 = 1001 \oplus 0100 = 1101$

$H_6 = M_6 \oplus H_5 = 0010 \oplus 1101 = 1111$

$H_7 = M_7 \oplus H_6 = 0111 \oplus 1111 = 1000$

Остаточно, хеш-значенням усього документу M вважається значення

$$H = 1000_2 = 8_{10}$$

2. Хеш-функції

Порівняння деяких хеш-функцій			
Хеш-функція	**Довжина дайджесту**	**Довжина блоку**	**Кількість раундів**
MD4	128	512	3
MD5	128	512	64
RIPEMD	128	512	48
SHA-1	160	512	80
SHA-256	256	512	80
SHA-512	512	1024	80
Whirlpool	512	512	10

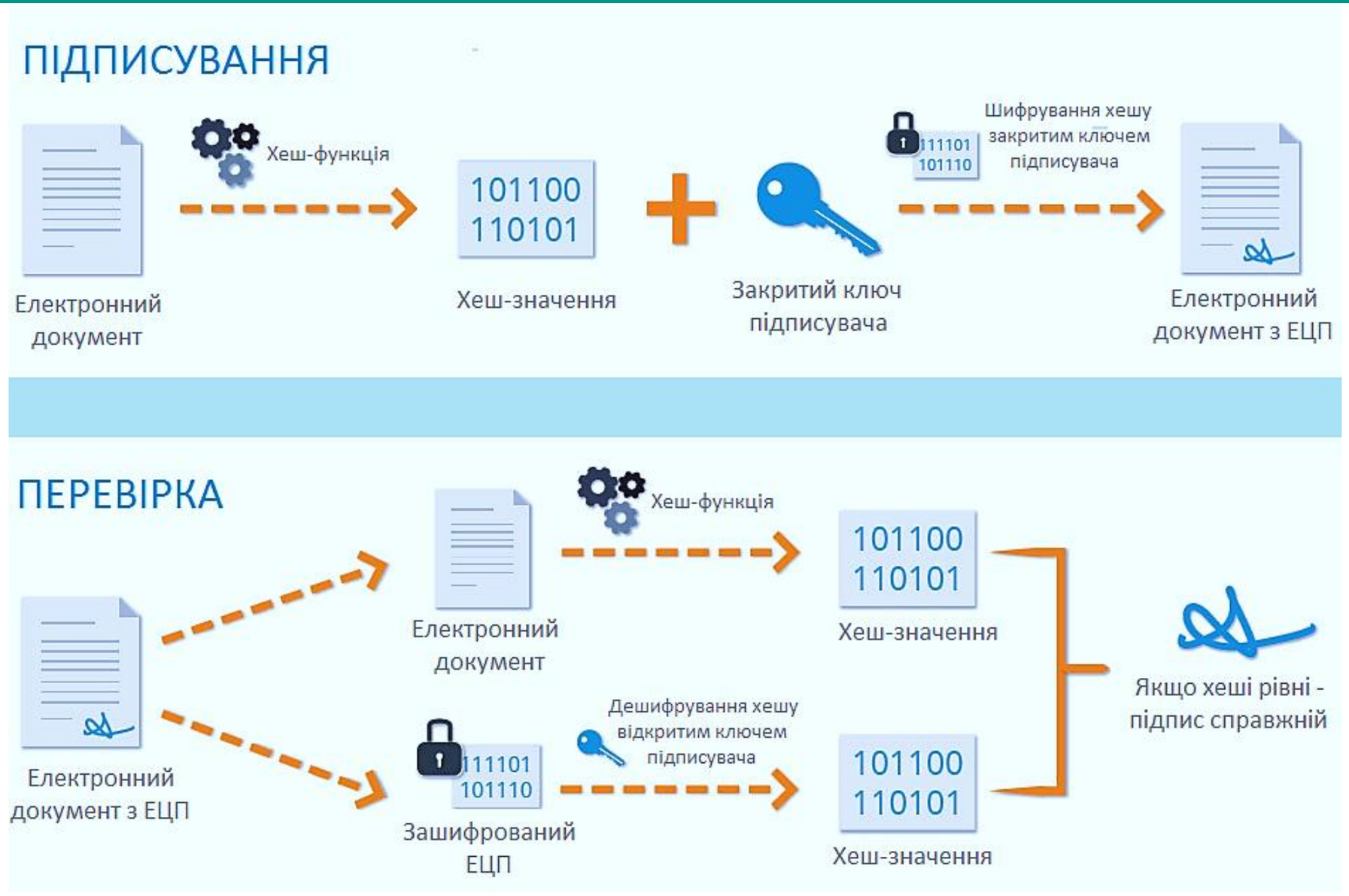
3. Процедури підписування та перевірки ЕЦП

1. Генерація пари ключів. За допомогою алгоритму генерації ключа рівноймовірним чином з набору можливих закритих ключів вибирається закритий ключ, обчислюється відповідний йому відкритий ключ

2. Формування підпису. Для заданого електронного документу за допомогою деякої хеш-функції обчислюється хеш-значення, після чого воно зашифровується із використанням закритого ключа підписувача. Зашифрований дайджест і є ЕЦП для даного документу

3. Перевірка (верифікація) підпису. Для отриманого документу одержувач знову обчислює його хеш-значення, після чого за допомогою відкритого ключа підписувача дешифрує ЕЦП. Якщо хеші рівні – підпис справжній

3. Процедури підписування та перевірки ЕЦП



3. Процедури підписування та перевірки ЕЦП

Управління ключами

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, який видається ЦСК дозволяє підтвердити дані про власника і його відкритий ключ

4. Стандарт цифрового підпису DSS

Національний інститут стандартів і технології США (NIST) розробив федеральний стандарт цифрового підпису DSS (Digital Signature Standard)

Для створення цифрового підпису використовується алгоритм DSA (Digital Signature Algorithm)

Як хеш-алгоритму стандарт передбачає використання алгоритму SHA (Secure Hash Algorithm)

4. Стандарт цифрового підпису DSS

Генерація ключів

1. Вибирається просте число q, таке що $2^{159} < q < 2^{160}$

2. Вибирається просте число p , таке що $2^{L-1} < p < 2^{L}$,
   $512 \le L \le 1024$ і L кратне 64

3. Обчислюється $g = h^{(p-1)/q} \bmod p$, де h будь-яке ціле число $1 < h < p - 1$
   таке, що $h^{(p-1)/q} \bmod p > 1$

4. Вибирається x – випадкове ціле число, таке що $0 < x < q$

5. Обчислюється $y = g^{x} \bmod p$

6. x і y є закритим і відкритим ключами, відповідно

4. Стандарт цифрового підпису DSS

Підпис повідомлення

Підпис повідомлення M із використанням закритого ключа підписувача виглядає наступним чином:

1. Вибирається k – випадкове ціле число, де $0 < k < q$ (k – разовий секретний ключ)
2. Обчислюється $r = (g^k \bmod p) \bmod q$
3. Обчислюється $s = (k^{-1}(\text{SHA}(M) + xr)) \bmod q$, де $\text{SHA}(M)$ – значення хеш-функції від повідомлення M
4. Підписом для повідомлення M є пара (r,s).
   Підпис разом з повідомленням пересилається одержувачеві

4. Стандарт цифрового підпису DSS

Перевірка підпису

Числа p, q, g і відкритий ключ y перебувають у відкритому доступі

1. Перевіряється, чи $0 < r < q$ та $0 < s < q$.
   Якщо це не так, підпис відхиляється

2. Обчислюється $w = (s)^{-1} \bmod q$

3. Обчислюється $u1 = ((\text{SHA}(M)w) \bmod q$

4. Обчислюється $u2 = ((r)w) \bmod q$

5. Обчислюється $v = (((g)^{u1} (y)^{u2}) \bmod p) \bmod q$

6. Підпис дійсний, лише, коли $v = r$

5. Схеми цифрового підпису RSA та El Gamal

Схема цифрового підпису RSA

Відкритий ключ: (e, n)

Закритий ключ: (d, n)

p, q – загальнодоступні параметри

Підписування

Для дайджесту H повідомлення M ЕЦП буде мати вигляд

$S = H^d \pmod{n}$

Перевірка підпису

Приймається пара (M, S) і перевіряється дійсність підпису

$H = S^e \pmod{n}$

5. Схеми цифрового підпису RSA та El Gamal

Схема цифрового підпису El Gamal

Відкритий ключ: (y, g і p)

Закритий ключ: x

k – сесійний ключ

$$a = g^k \bmod p$$

$$b = y^k M \bmod p$$

Підписування

Для дайджесту H повідомлення M ЕЦП буде мати вигляд

$$H = (xa + kb) \bmod (p - 1)$$

Перевірка підпису

Підпис вважається дійсним, якщо

$$y^a a^b \bmod p = g^H \bmod p$$

СРИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ:

1. В. І. Єсін, О. О. Кузнецов, Л. С. Сорока. – Х. : ХНУ імені В. Н. Каразіна, 2013. – 632 с.
2. Блінцов В. С. Математичні основи криптології + CD : Навчальний посібник для студ. вищих навч. закл. / В. С. Блінцов, Ю. Л. Гальчевський. – Миколаїв : Національний ун-т кораблебудування ім. адмірала Макарова, 2006. – 232 с.
3. Богуш В. М. Криптографічні застосування елементарної теорії чисел : Навч. посібник / В. М. Богуш, В. А. Мухачов. – К. : Державний ун-т інформаційно-комунікаційних технологій, 2006. – 126 с.
4. Бурячок В.Л., Гулак Г.М., Толубко В.Б. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби: Підручник. – К.: ДУТ, 2015. – 449 с.
5. Горбенко І. Д. Захист інформації в інформаційно-телекомунікаційних системах : Навч. посіб. для студ. Ч. 1. Криптографічний захист інформації / І. Д. Горбенко, Т. О. Гріненко. – Х. : Харк. нац. ун-т радіоелектрон., 2004. – 368 с.
6. Горбенко І. Д. Прикладна криптологія. Теорія. Практика. Застосування : Монографія / І. Д. Горбенко, Ю. І. Горбенко. – Харків : Видавництво "Форт", 2012. – 880 с.
7. Грайворонський М. В. Безпека інформаційно-комунікаційних систем : Підручник / М. В. Грайворонський, О. М. Новіков. – К. : Видавнича група BHV, 2009. – 608 с.
8. Грищук Р.В., Даник Ю.Г. Основи кібернетичної безпеки: Монографія. – Житомир: ЖНАЕУ, 2016. – 636 с.
9. Задірака В. К. Комп'ютерна криптологія : підручник / В. К. Задірака, О. С. Олексюк. – К. : Тернопільська академія народного господарства; НАН України; Інститут кібернетики ім. В. М. Глушкова, 2002. – 504 с.
10. Задірака В.К. Методи захисту фінансової інформації / В.К. Задірака, О.С. Олексюк. – К.: Вища школа, 2009. – 460 с.
11. Захист інформації в мережах передачі даних: Підручник / Юдін О. К., Корченко О. Г., Конахович Г. Ф. – К. : Вид-во ТОВ "НВП" ІНТЕРСЕРВІС", 2009. – 716 с.
12. Касянчук М. Досконала форма системи залишкових класів: методи побудови та застосування (Монографія) / М. Касянчук. – Тернопіль: ТНЕУ, 2019. – 224 с.
13. Корченко О. Г. Охорона конфіденційної інформації підприємства : Навч. посіб. / О. Г. Корченко, Ю. О. Дрейс. – Житомир: ЖВІ НАУ, 2011. – 172 с.
14. Лісовська Ю. Кібербезпека. Ризики та заходи. - К.: Кондор, 2019. - 272 с.
15. Математичні основи криптоаналізу [Текст]: навч. посібник / С. О. Сушко, Г. В. Кузнецов, Л. Я. Фомичова, А. В. Корабльов. – Д.: НГУ, 2004. – 391 с.
16. Основи криптографічного захисту інформації [Текст]: підручник / Г. М. Гулак, В. А. Мухачов, В. О. Хорошко, Ю. Є. Яремчук. – В.: ВНТУ, 2011. – 198 с.
17. Остапов С. Е. Основи криптографії: Навчальний посібник / С.Е. Остапов, Л. О. Валь. – Чернівці: Книги – XXI, 2008. – 188 с.
18. Остапов С. Е., Валь Л. О. Основи криптографії. Чернівці: Книги-XXI, 2008. – 188 с.
19. Стеганографія: Навч. посіб. / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 232 с.
20. Тарнавський Ю.А. Технології захисту інформації [Електронний ресурс]: підручник. – К.: КПІ ім. Ігоря Сікорського, 2018. – 162 с. Режим доступу до ресурсу: https://ela.kpi.ua/bitstream/123456789/23896/1/TZI_book.pdf /