МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Г. Л. Козіна

КРИПТОГРАФІЯ ВІД ІСТОРІЇ ДО СУЧАСНИХ СТАНДАРТІВ

Навчальний посібник

Запоріжжя • НУ «Запорізька політехніка» • 2020

УДК 004.056.55 (075.8)
К59

Рекомендовано до друку вченою радою
Національного університету «Запорізька політехніка»
(протокол № 6/20 від 06 липня 2020 року).

Рецензенти:

М.В. Новожилова – завідувач кафедри комп'ютерних наук та інформаційних технологій Харківського національного університету міського господарства імені О. М. Бекетова, доктор фізіко-математичних наук, професор.

В.І. Корнієнко – завідувач кафедри безпеки інформації та телекомунікацій Національного технічного університету «Дніпровська політехніка», доктор технічних наук, професор.

К59 **Козіна Г.Л.**

Криптографія від історії до сучасних стандартів: навч.посібник / Г. Л. Козіна. – Запоріжжя : НУ «Запорізька політехніка», 2020. – 192 с.

ISBN 978-617-529-278-5

В посібнику відображено розвиток криптографічних перетворень від перших відомих криптоалгоритмів до сучасних чинних стандартів. Необхідний математичний апарат викладено в доступній формі безпосередньо перед описом криптоалгоритмів. Увесь викладений матеріал проілюстровано числовими прикладами. Посібник може бути використаний при написанні курсових та дипломних робіт студентів, стати основою наукових досліджень для магістрів й аспірантів.

УДК 004.056.55 (075.8)

ISBN 978-617-529-278-5

Присвячено Рабенау В.В.

ЗМІСТ

ПЕРЕДМОВА

Посібник написано на основі курсів «Прикладна криптологія», «Протоколи цифрового підпису», «Методи побудови та аналізу криптосистем», які викладаються автором студентам та магістрам, що навчаються за напрямом «Безпека інформаційних і комунікаційних систем» в галузі знань «Інформаційні технології».

Метою автора посібника було показати шлях розвитку криптографічних перетворень від перших відомих криптоалгоритмів до сучасних чинних стандартів. Необхідний для розуміння сучасних стандартів математичний матеріал викладено в доступній формі безпосередньо перед описом криптоалгоритмів.

Посібник складається з 7 розділів.

У першому розділі розглядаються найпростіші криптоперетворення, які не потребують серйозної математичної підготовки. Другій розділ цілком присвячено асиметричному алгоритму RSA з необхідними математичними алгоритмами. Задача дискретного логарифмування також виділена в окремий розділ.

В четвертому розділі розглядаються історично перші схеми цифрового підпису. А в п’ятому розділі розглядаються вже сучасні асиметричні протоколи на еліптичних кривих.

На даний час існують різні схеми цифрового підпису, зокрема мультипідпис, агрегований підпис, сліпий тощо. Ця тематика відображена в шостому розділі.

Сьомий розділ присвячено симетричному шифруванню. В ньому розглянуто сучасні стандарти США, Китаю, України.

Увесь викладений матеріал проілюстровано числовими прикладами. В кінці кожного розділу наведено контрольні питання.

Матеріал посібника може бути використаний при написанні курсових та дипломних робіт студентів, стати основою наукових досліджень для магістрів й аспірантів.

Автор висловлює подяку всім, хто надав підтримку при написанні та виданні посібника.

Зауваження, пропозиції та рекомендації просимо направляти за адресою: 69063, м. Запоріжжя, вул. Жуковського, 64, Національний університет «Запорізька політехніка», кафедра “Захист інформації”.

1 НАЧАЛА КРИПТОГРАФІЇ

В розділі розглядаються перші історичні відомі шифри, найпростіший математичний апарат, пов'язаний з теорією чисел, лінійні рекурентні послідовності, класифікація симетричних шифрів.

1.1 Історія криптографії. Прості шифри

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

Шифр «Сцитала»

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

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

Шифр перестановки

Задається перестановка з n елементів.

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

Розглянемо приклад шифрування для перестановки

1	2	3	4	5
3	2	5	1	4

Вихідний текст:
«СВЯЩЕ ННАЯР ИМСКА ЯИМПЕ РИЯЬЭ»

Зашифрований текст
«ЩВСЕЯЯННРАКМИАСПИЯЕМЬИРЭЯ»

Зашифрований текст виписується без пропусків.

Шифр Цезаря

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

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

16	17	18	19	20	21	22	23	24	25	26	27
P	Q	R	S	T	U	V	W	X	Y	Z	*Space*

Вихідний текст: «SAPIENTI SAT» – УМНОМУ ДОСТАТОЧНО

Зашифрований текст: «VDSLHQWL VDW»

Квадрат Полібія[1]

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

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

Шифр простої заміни

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

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

Частотний криптоаналіз шифрі простої заміни

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

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

---

[1] Полібій - грецький державний діяч, полководець, історик, жив в III в. до н.е.

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

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

Шифр Гронсфельда[2]

Ключем э деяке десяткове число. Наприклад, 13579.

Вихідний текст	G	E	R	M	A	N	Y
Ключ	1	3	5	7	9	1	3
Зашифрований текст	H	H	W	T	J	O	B

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

Біграмні шифри

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

Біграмні шифри були запропоновані Іоганном Трісемусом (Німеччина). У 1508 році він опублікував першу друковану робо-

---

[2] Граф Гронсфельд був начальником першого дешифрувального відділення у Німеччині. У 1734р. він модернізував ідеї Альберті і Вижинера – творців шифрів багатоалфавитної заміни.

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

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

ту з криптології «Поліграфія». Шифр Playfair, заснований на біграмном шифрі, використовувався Великобританією в Першу світову війну.

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

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

Рисунок 1.1 – Квадрат Полібія

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

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

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

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

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

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

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

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

Рисунок 1.2 – Подвійний квадрат Полібія

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

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

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

Шифр Вернама[3]

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

У 1917 році Вернам винаходить перший автоматичне шифрувальний - дешифровальний пристрій, а наступного року, після консультацій з колегами з галузі і військових відомств, пропонує прийом «нескінченного ключа» – те, що ми сьогодні і називаємо «шифром Вернама», або «одноразовим блокнотом». Через три з лишком десятки років Клод Шеннон (1916-2001) строго доведе абсолютну теоретико - інформаційну стійкість шифру Вернама.

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

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

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

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

Шифр Вернама є абсолютно стійким шифром.

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

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

1.2 Поняття теорії чисел

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

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

Для перетворення тексту на число кожна літера або знак замінюється на запис у цифрах, згідно з таблицею кодування. Стандартна таблиця кодування ASCII наведена в таблиці 1.1.

Приклад 1.

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

$$\text{Student} = 53\ 74\ 75\ 64\ 65\ 6E\ 74_{16} = 23490470611349108_{10}.$$

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

Знайдемо, яке повідомлення записується десятковим числом 1515082837: $1515082837_{10} = 5A\ 4E\ 54\ 55_{16} = \text{ZNTU}$

Таблиця 1.1 –Таблиця ASCII-кодів

Основная таблица ASCII

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

Расширенная таблица ASCII (cp866)

	80	90	A0	B0	C0	D0	E0	F0
0	А	Р	а	░	└	╨	р	≡
1	Б	С	б	▒	┴	╤	с	±
2	В	Т	в	▓	┬	╥	т	≥
3	Г	У	г	│	├	╙	у	≤
4	Д	Ф	д	┤	─	╘	ф	⌠
5	Е	Х	е	╡	┼	╒	х	⌡
6	Ж	Ц	ж	╢	╞	╓	ц	÷
7	З	Ч	з	╖	╟	╫	ч	≈
8	И	Ш	и	╕	╚	╪	ш	°
9	Й	Щ	й	╣	╔	┘	щ	∙
A	К	Ъ	к	║	╩	┌	ъ	·
B	Л	Ы	л	╗	╦	█	ы	√
C	М	Ь	м	╝	╠	▄	ь	ⁿ
D	Н	Э	н	╜	═	▌	э	²
E	О	Ю	о	╛	╬	▐	ю	■
F	П	Я	п	┐	╧	▀	я	

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

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

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

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

Алгоритм Евкліда пошуку найбільшого спільного дільника

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

$$b = a\cdot q_0 + r_1$$
$$a = r_1\cdot q_1 + r_2$$
$$r_1 = r_2\cdot q_2 + r_3$$
$$r_2 = r_3\cdot q_3 + r_4$$
$$\ldots\ldots\ldots\ldots$$
$$r_{n-2} = r_{n-1}\cdot q_{n-1} + r_n$$
$$r_{n-1} = r_n\cdot q_n$$
$$НСД(a,b) = r_n.$$

Приклад 2.

Знайдемо $НСД(114,534)$.

$534 = 114\cdot 4 + 78$,

$114 = 78\cdot 1 + 36$,

$78 = 36\cdot 2 + 6$,

$36 = 6\cdot 6$.

$НСД(114,534) = 6$.

Знайдемо $НСД(114,534,22)$.

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

$НСД(114,534,22) = 2$.

1.3 Порівняння та їхні властивості

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

$$a = b \bmod m .$$

Порівняння мають наступні властивості:

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

Доведемо, наприклад, властивість 6.

$$a = b \bmod m \Leftrightarrow a - b = m \cdot t \quad t \in Z$$
$$ac - bc = (a - b)c = m \cdot c \cdot t = m \cdot z, \; z \in Z, \; \Rightarrow ac = bc \bmod m$$

Всі цілі числа, які мають однаковий залишок від ділення на m, порівняні між собою за модулем m і належать до одного *класу залишків* за цим модулем. Клас залишків позначається за найменшим додатнім числом (або 0), яке до нього належить.

Наприклад, за модулем 5 існують наступні класи залишків {0, 1, 2, 3, 4} (стовпці в таблиці 1.2).

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

Таблиця 1.2 –Класи залишків за модулем 5

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

Приклад.

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

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

$\frac{7}{8}\bmod 5=\frac{2}{3}=\frac{2}{-2}=-1=4$.

1.4 Функція Ейлера

Функцією Ейлера[4] $\varphi(n)$ називається число натуральних чисел, що не перевищують n і взаємно простих з n.

Функція Ейлера має наступні властивості:

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

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

---

[4] Леонард Ейлер (1701-1783)

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

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

Приклад.

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

$$\varphi(8) = \varphi(2^3) = 2^2 \cdot (2-1) = 4,$$

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

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

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

$$17^{32} \bmod 11 = 6^{32} = 6^{30} \cdot 6^2 = \left(6^{10}\right)^3 \cdot 6^2 = 1^3 \cdot 6^2 = 36 \bmod 11 = 3$$

1.4 Лінійні рекурентні послідовності

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

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

$C = T \oplus \gamma$ зашифрування

$T = C \oplus \gamma$ розшифрування.

В симетричному шифруванні один і той самий ключ має бути наявний і у відправника, і у отримувача. Гама шифру має

таку ж довжину, як і повідомлення, тому недоцільно передавати її по каналам зв'язку.

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

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

ЛРП задається наступним співвідношенням для :

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

,

де $k = 1,2,...$

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

Величина n – називається *глибиною* послідовності.

Тобто глибина послідовності визначається різницею між найстаршим і наймолодшим індексами елементів в запису послідовності.

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

Характеристичним многочленом лінійної рекурентної послідов-ності
$x_{k+n} = a_1 \cdot x_{k+n-1} \oplus a_2 \cdot x_{k+n-2} \oplus ... \oplus a_{n-1} \cdot x_{k+1} \oplus a_n \cdot x_k$
глибини n називається многочлен степеня n:

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

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

Незвідними многочленами першого і другого степеня є λ, $\lambda + 1$, $\lambda^2 + \lambda + 1$.

Якщо многочлен $f(\lambda)$ степеня n є незвідним, то він є дільником многочлена $g(\lambda) = \lambda^{2^n-1} + 1$.

Лінійній рекурентній послідовності $x_{k+3} = x_{k+1} \oplus x_k$ глибини $n = 3$ відповідає характеристичний многочлен степеня 3 $f(\lambda) = \lambda^3 + \lambda + 1$.

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

Многочлен $f(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + 1$ не є незвідним, оскільки його можна розкласти на множники:

$$f(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + 1 = (\lambda + 1) \cdot (\lambda^3 + \lambda + 1).$$

Многочлен $f(\lambda) = \lambda^3 + \lambda + 1$ є незвідним, оскільки він не має дільників першого і другого степеня.

Тому многочлен $g(\lambda) = \lambda^7 + 1$ діліться на многочлен $f(\lambda)$: $f(\lambda) = \lambda^7 + 1 = (\lambda^3 + \lambda + 1) \cdot (\lambda^4 + \lambda^2 + \lambda + 1)$.

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

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

Порядком незвідного многочлена $f(\lambda)$ називається найменше число s, таке що многочлен $\lambda^s + 1$ ділиться на многочлен $f(\lambda)$.

Якщо характеристичний многочлен послідовності є незвідним, то період послідовності дорівнює порядкові цього многочлена.

Відомо, що многочлен $\lambda^m + 1$ ділиться на многочлен $\lambda^k + 1$ у тому і тільки тому випадку, якщо m ділиться на k. Таким чином, порядок s незвідного многочлена є дільником числа $2^n - 1$.

Звідси якщо характеристичний многочлен лінійної рекурентної послідовності є незвідним, то період послідовності T буде дільником значення $T_{\max} = 2^n - 1$.

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

Послідовність $x_{k+3} = x_{k+1} \oplus x_k$ має максимальне можливий період, рівний 7, при будь-яких початкових значеннях, окрім всіх нулів. Послідовність $x_{k+4} = x_{k+3} \oplus x_{k+2} \oplus x_k$ може мати будь-який період, менший числа 15.

1.6 Симетричні шифри

Всі шифри до 70-х років XX століття були симетричними. Тобто для шифрування і розшифрування в симетричних шифрах використовується один і той же ключ. Симетричні шифри умовно діляться на блочні і потокові. Шифри з послідовним виконанням перетворень над елементами відкритого тексту називаються потоковими. При цьому зазвичай розуміють шифри, які виконують перетворення над елементами невеликого розміру (буква алфавіту, один біт). В блочних шифрах відкритий текст розбивається на блоки фіксованого розміру (в сучасних шифрах декілька байт).

В перших відомих шифрах основними ідеями були заміна та перестановка елементів тексту. В сучасних шифрах як в блочних, так і в потокових обов'язково присутні обидві ці ідеї.

К потоковим шифрам відносяться шифр Цезаря, Полібія, Гронсфельда, зі сучасних – A5, RC4.

К блочним шифрам відносяться шифр Сцитала, решітка Кардано, перестановки, зі сучасних – DES, ГОСТ 28147-89, IDEA, FIPS 197 (Rijndael), ДСТУ 7624-2014.

Блокові шифри вважаються більш дослідженими і тому більш затребуваними як національні стандарти. В розділі 7 описано 3 сучасних стандарти симетричного шифрування.

Контрольні питання

1. Що є ключем в шифрі Полібія? Які вимоги висуваються до ключа?
2. За яким принципом формується квадрат Полібія?
3. Які ще відомі варіанти шифра Полібія?
4. Що таке таблиця замін? Які до неї вимоги?
5. Які різновиди шифру простої заміни існують?
6. Що таке частотний криптоаналіз?
7. Чи є шифр Полібія різновидом шифру простої заміни?
8. Які числа називаються простими? Взаємно простими?
9. Для чого використовується алгоритм Евкліда?
10. Які числа називаються порівняними за модулем m ?
11. За яким модулем всі цілі числа порівняні між собою?
12. Якими властивостями володіють порівняння?
13. Що таке клас залишків за модулем?
14. Що показує функція Ейлера? Як її обчислити?
15. Як формулюється теорема Ейлера?
16. Як формулюється Мала теорема Ферма?
17. Як пов'язані теорема Ейлера і Мала теорема Ферма?
18. Як задається бітова лінійна рекурентна послідовність (ЛРП)?
19. Для чого використовуються ЛРП в криптографії?
20. Що таке глибина ЛРП?
21. Що таке характеристичний многочлен ЛРП?
22. Який многочлен називається незвідним?
23. Як перевірити многочлен на незвідність?
24. Що таке період лінійної рекурентної послідовності і від чого він залежить?
25. Які шифри називаються потоковими?
26. Наведіть приклади потокових шифрів.
27. Які шифри називаються блоковими?
28. Наведіть приклади потокових шифрів.

2 АСИМЕТРИЧНИЙ АЛГОРИТМ ШИФРУВАННЯ RSA

Розділ присвячено першому асиметричному алгоритму RSA. Наведено необхідні математичні алгоритми для реалізації обчислень за алгоритмом RSA.

2.1 Асиметрична криптографія, або криптографія з відкритим ключем

Криптосистеми з відкритим ключем (асиметричні криптосистеми) були розроблені в другій половині сімдесятих років. В асиметричних криптосистемах процедури прямого і зворотного криптоперетворення виконуються на різних ключах і не мають між собою очевидних і легковідсліджуваних зв`язків, що дозволяють за одним ключем визначити інший. В такій схемі знання тільки ключа зашифрування не дозволяє розшифрувати повідомлення, тому він не є секретним елементом шифру і зазвичай публікується учасником обміну для того, щоб будь-хто бажаючий міг послати йому зашифроване повідомлення.

Криптосистеми з відкритим ключем використовуються в трьох напрямках: шифрування (наприклад, алгоритм RSA), створення спільного секретного ключа по відкритих каналах зв’язку, електронний цифровий підпис.

Принцип функціонування асиметричної криптосистеми шифрування полягає в наступному:

користувач А генерує асиметричну пару ключів – відкритий (незасекречений) і секретний – і передає відкритий ключ по незахищеному каналу користувачу Б;

користувач Б шифрує повідомлення, використовуючи відкритий ключ шифрування користувача А;

користувач Б посилає зашифроване повідомлення користувачу А по незахищеному каналу;

користувач А отримує зашифроване повідомлення і дешифрує його, використовуючи свій секретний ключ.

Асиметрична пара ключів {відкритий ключ; секретний ключ} обчислюється за допомогою спеціальних алгоритмів, при чому жоден ключ не може бути виведений з іншого.

2.2 Розв'язання порівняння першого степеня

Порівняннями першого степеня називаються вирази вигляду

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

де $a < m$ – задані цілі числа.

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

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

В інших випадках розв'язків немає.

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

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

Знаходимо залишки від ділення за звичайним алгоритмом Евкліда (див. 1.2):

$$m = a \cdot q_0 + r_1$$
$$a = r_1 \cdot q_1 + r_2$$
$$r_1 = r_2 \cdot q_2 + r_3$$
$$r_2 = r_3 \cdot q_3 + r_4$$
$$\ldots\ldots\ldots\ldots$$
$$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n$$

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

Заповнюємо таблицю 2.1, використовуючи отримані значення.

Таблиця 2.1 – Пошук розв'язку порівняння першого степеня

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

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

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

Якщо таблицю 2.1 заповнено вірно, P_n має дорівнювати m: $P_n = m$.

Розв'язок порівняння отримуємо з таблиці 2.1 за формулою $x = (-1)^n \cdot P_{n-1} \bmod m$.

Для розв'язання порівняння

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

спочатку знаходять розв'язок порівняння

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

тобто $z = \frac{1}{a} \bmod m$, а потім формують розв'язок порівняння $a \cdot x = b \bmod m$:

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

Приклад 1.
Знайдемо розв'язок порівняння $9 \cdot x = 1 \bmod 52$.
Розв'язок порівняння існує, оскільки 9 і 52 взаємно прості.
За алгоритмом Евкліда знаходимо залишки від ділення:
$52 = 9 \cdot 5 + 7$
$9 = 7 \cdot 1 + 2$
$7 = 2 \cdot 3 + 1$
$2 = 1 \cdot 2$
Заповнимо таблицю

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

З отриманої таблиці обчислимо розв'язок:
$x = (-1)^3 \cdot 23 \bmod 52 = -23 \bmod 52 = 29,$
$x = 29$.
Перевірка: $9 \cdot 29 = 261 \bmod 52 = 1$.

Приклад 2.
Знайдемо розв'язок порівняння $9 \cdot x = 10 \bmod 52$.
Спочатку знайдемо розв'язок порівняння $9 \cdot z = 1 \bmod 52$.
Згідно з Прикладом 1, $z = 29$.
Звідси розв'язком порівняння $9 \cdot x = 10 \bmod 52$ є
$x = 29 \cdot 10 = 290 \bmod 52 = 30$,
$x = 30$.
Перевірка: $9 \cdot 30 = 270 \bmod 52 = 10$.

2.3 Алгоритм послідовного зведення в квадрат

В багатьох криптоалгоритмах, зокрема RSA, виникає необхідність обчислювати зведення числа у великий степінь за модулем:

$b = a^k \bmod m$.

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

Алгоритм послідовного зведення в квадрат

1. Розкладемо k по степенях двійки:
   $k = k_1 + k_2 \cdot 2 + k_3 \cdot 2^2 + ... + k_{t+1} \cdot 2^t$
2. $b := 1$.
3. $A := a$.
4. Якщо $k_1 = 1$, тоді $b := a$.
5. Цикл по i от 2 до t
   5.1 $A := A^2 \bmod m$
   5.2 Якщо $k_i = 1$, тоді $b := b \cdot A$
6. Повертаємо b.

Приклад 1.

Обчислимо $b = 3^{42} \bmod 79$.

Розкладемо $k = 42$ по степенях двійки:

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

$b := 1$.

$A := 3$.

$k_1 = 0$.

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

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

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

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

$k_6 = 1$, $A := 16^2 = 256 \bmod 79 = 19$,

$b := 36 \cdot 19 = 684 \bmod 79 = 52$.

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

Приклад 2.

Обчислимо $b = 2^{130} \bmod 35$.

Обчислення b доцільно знайти за допомогою теореми Ейлера: оскільки числа 2 і 35 взаємно прості, то $2^{\varphi(35)} = 1 \bmod 35$.

Знайдемо значення функції Ейлера:

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

Звідси $2^{24} = 1 \bmod 35$.

Оскільки $130 = 24 \cdot 5 + 10$, то

$$b = 2^{130} = \left(2^{24}\right)^5 \cdot 2^{10} \bmod 35 = 1^5 \cdot 2^{10} \bmod 35 = 2^{10} \bmod 35$$

.

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

Розкладемо $k = 10$ по степенях двійки:

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

$b := 1$.

$A := 3$.

$k_1 = 0$.

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

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

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

Таким чином, $b = 2^{10} \bmod 35 = 9$.

Звідси, $b = 2^{130} \bmod 35 = 9$

2.4 Тести на простоту

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

Існують різні методи перевірити, чи є число простим, що базуються на властивостях простих чисел. Так, зокрема, мала теорема Ферма (див. 1.4) стверджує, що для простого числа p виконується співвідношення $a^{p-1} = 1 \bmod p$ для всіх $a < p$.

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

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

Тест Ферма

Нехай дано число n.

1. Вибираємо випадкове ціле a, $1 < a < n$.

2. Знаходимо найбільший спільний дільник чисел a і n – $НСД(a,n)$ – за допомогою алгоритму Евкліда.

3. Якщо $НСД(a,n) > 1$, повертаємо "n – складене".

4. Якщо $НСД(a,n) = 1$, обчислюємо $v = a^{n-1} \bmod n$ за допомо-гою алгоритму послідовного зведення в квадрат (див. 2.3).

5. Якщо $v > 1$, повертаємо "n – складене".

6. Якщо $v = 1$, то число a є свідком простоти.

7. Повторюємо пункти 1-6 доти, поки не наберемо t свідків простоти.

Приклад 1.

Знайти всіх свідків простоти числа 21.

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

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

$$2^{20} \bmod 21 = 4 \neq 1, \qquad 19^{20} \bmod 21 = 4 \neq 1$$
$$4^{20} \bmod 21 = 16 \neq 1, \qquad 17^{20} \bmod 21 = 16 \neq 1$$
$$5^{20} \bmod 21 = 4 \neq 1, \qquad 16^{20} \bmod 21 = 4 \neq 1$$

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

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

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

Приклад 2.

Визначити, чи є число 33 простим.

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

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

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

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

Звідси по тесту Ферма випливає, що число 15 – складене.

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

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

Зокрема, існують складені числа *K*, які проходять тест Ферма, тобто всі числа a менші *K* і взаємно прості з ним є свідками простоти числа *K*. Такі числа називають числами Кармайкла[5].

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

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

Деякі числа Кармайкла можна представити у вигляді добутку

(6j+1)(12j+1)(18j+1), j=1, 6, 35.

j=1 7·13·19= 1729

j=6 37·73·109=294409

---

[5] Robert Daniel Carmichael (1879 –1967)

j=35 $211 \cdot 421 \cdot 631 = 56052361$.

Нехай p - непарне просте число і $p - 1 = 2^s \cdot r$.

Тоді для будь-якого a, $1 \le a \le p - 1$, виконується або

$a^r = 1 \bmod p$, або

$a^{k(j)r} = -1 \bmod p$ для деякого j, $0 \le j \le s - 1$, $k(j) = 2^j$.

Нехай є натуральне непарне число n і $n - 1 = 2^s \cdot r$. Ціле число a, взаємно просте з n, (a, n) =1, для якого виконується або

$a^r = 1 \bmod n$, або

$a^{k(j)r} = -1 \bmod n$ для деякого j, $0 \le j \le s - 1$, $k(j) = 2^j$

називається *сильним свідком простоти* числа *n.*

Приклад 3.

Знайти сильного свідка простоти числа 13.

Представимо число 13-1 у вигляді $13-1=12=2^2 \cdot 3$, s=2, r=3.

Візьмемо a=5: $a^r = 5^3 \bmod 13 = 8 \neq \pm 1$. При s = 1 отримаємо $(5^3)^2 \bmod 13 = -1$. Звідси число a=5 є сильним свідком простоти числа 13.

Тест Міллера - Рабина на простоту

Нехай дане число *n.*

1. Представимо число n - 1 у вигляді $n - 1 = 2^s \cdot r$.
2. Вибираємо випадкове ціле $2 \le a \le n - 2$.
3. Знаходимо найбільший спільний дільник чисел a і n – НСД (a, n) - за допомогою алгоритму Евкліда.
4. Якщо НСД (a, n) > 1, то повертаємо "n - складене".
5. Якщо НСД (a, n) = 1, то обчислюємо
$$y = a^r \bmod n$$
за допомогою алгоритму послідовного зведення в квадрат.
6. Якщо $y \neq 1$ и $y \neq n - 1$, то $j := 1$.

Цикл while ($j \le s-1$ & $y \ne n-1$)

1) $y := y^2 \bmod n$;
2) якщо $y = 1$, то повертаємо «n – складене»;
3) $j := j + 1$

7. Якщо $y \ne n-1$, то повертаємо «n – складене»
8. Якщо $y = n-1$, то число а є сильним свідком простоти .
9. Повторюємо пункти 2-8 доти , поки не наберемо t сильних свідків простоти.

Приклад 4.

Знайти всіх сильних свідків простоти числа 561.

Представимо число 561-1 у вигляді $561-1=560=2^4 \cdot 35$, s=4, r=35.

Візьмемо a=103: $a^r = 103^{35} \bmod 561 = 1$. Звідси число a=103 є сильним свідком простоти числа 561.

Числа 256, 460, 511 також є сильними свідками простоти числа 561.

Нехай для числа n знайдено t сильних свідків простоти. Тоді імовірність того, що число n складене , не перевершує 4^{-t}

$P\{ n$ – складене $\} \le 4^{-t}$.

Якщо число a є сильним свідком простоти числа n , то воно є свідком простоти.

Якщо число n проходить тест Міллера-Рабина на простоту, то воно пройде тест Ферма на простоту.

Числа Мерсенна

Нехай $s \ge 2$ - ціле число. Числа вигляду $2^s - 1$ називаються *числами Мерсенна*.

Якщо число $2^s - 1$ - просте, то воно називається *простим числом Мерсенна*[6].

---

[6] Мерсенн (1588-1648) – французский математик.

Приклад 5.

s=2: $2^2-1=3$ – просте число s=3: $2^3-1=7$ – просте число

s=4: $2^4-1=15$ – не просте число s=5: $2^5-1=31$ – просте число

s=6: $2^6-1=63$ – просте число s=7: $2^7-1=127$ – просте число

s=8: $2^8-1=255$ – не просте число s=9: $2^9-1=511$ – не просте число

Визначити простоту числа Мерсенна можна за критерием Люка:

Нехай $s \geq 3$. Число Мерсенна $n = 2^s - 1$ є простим у тім і тільки тім випадку , коли виконуються наступні умови :

1. s - просте число;
2. послідовність цілих чисел, яка визначена умовами

$$u_0 = 4, \quad u_{k+1} = (u_k{}^2 - 2) \bmod n, \quad k \geq 0,$$

відповідає рівнянню

$$u_{s-2} = 0.$$

Приклад 6.

Перевіримо простоту числа Мерсенна $2^5-1=31$.

1) s=5 – просте число;
2) $u_0=4$ $u_1=4^2-2=14$ $u_2=(14^2-2) \bmod 31 = 194 \bmod 31 = 8$

$u_3==(8^2-2) \bmod 31 = 0$

Число Мерсенна 31 є простим.

Приклад 7.

Чи є число Мерсенна $2^7-1=127$ простим?

3) s=7 – просте число;
4) $u_0=4$ $u_1=4^2-2=14$

$u_2=(14^2-2) \bmod 127 = 67$ $u_3=(67^2-2) \bmod 127 = 42$

$u_4=(42^2-2) \bmod 127 = 111$ $u_5=(111^2-2) \bmod 127 = 0$

Число Мерсенна 127 є простим.

Приклад 8.

Чи є число Мерсенна $2^{11}-1=2047$ простим?

1) s=11 – просте число;

2) $u_0=4$ $u_1=4^2-2=14$

$u_2=(14^2-2) \bmod 2047 = 194$ $u_3==(194^2-2) \bmod 2047 = 788$

$u_4=(788^2-2) \bmod 2047 = 701$ $u_5=(701^2-2) \bmod 2047 = 119$ $u_6=(119^2-2) \bmod 2047 = 1877$ $u_7=(1877^2-2) \bmod 2047 = 240$ $u_8=(240^2-2) \bmod 2047 = 282$ $u_9=(282^2-2) \bmod 2047 = 1736$

Число Мерсенна 2047 не є простим. Дійсно, $2047=23\cdot 89$.

Зауваження. Одним із способів виробництва великих простих чисел могла б бути побудова великих простих чисел Мерсенна заданого порядку. Однак відоме лише кілька простих числа Мерсенна: при s = 2, 3, 5, 7, 13, 17, 19, 31 (1750г.), 61 (1883р.), 89 (1907р.), 107 (1914р.) , 127 (1876р.), 521, 607, 1279, 2203, 2281 (всі п'ять 1952р.), 3217 (1957р.), 4253, 4423 (обидва 1962р.), 9689, 9941, 11213 (1965р.). У 2001 р було знайдено 39-те просте число Мерсенна - при s = 13466917.

На січень 2019 року відомим найбільшим простим числом Мерсенна є число при s = 82589933, знайдене 7 грудня 2018 го-да Патріком Лярош в рамках проекту добровільних обчислень GIMPS.

Апроксимація для простого числа

Нехай p_n – n -е просте число.

Для любого $n \geq 6$ виконується нерівність

$$n \ln n < p_n < n(\ln n + \ln \ln n).$$

Приклад 9.

Знайдемо оцінки 200-ного простого числа.

$n = 200$, $n \ln n = 1059.66$, $n(\ln n + \ln \ln n) = 1393.14$.

Таким чином, $1060 < p_{200} < 1393$. (Зауважимо, що $p_{200}=1217$.)

Число простих чисел, що не перевищують задане

Позначимо через $\pi(x)$ кількість простих чисел, що не перевищують число x.

Нерівність Чебишева:

При $x > 70$ виконується нерівність

$$x / (\ln x - 0.5) < \pi(x) < x / (\ln x - 1.5).$$

Приклад 10.

Знайдемо оцінки кількості простих чисел в діапазоні від 1060 до 1393.

$x = 1060$

$x / (\ln x - 0.5) = 163.93, \qquad x / (\ln x - 1.5) = 193.92;$

$x = 1393$

$x / (\ln x - 0.5) = 206.70, \qquad x / (\ln x - 1.5) = 242.72.$

З нерівності Чебишева отримуємо

$$164 < \pi(1060) < 193 \text{ и}$$

$$207 < \pi(1393) < 242$$

Звідси маємо таки оцінки кількості простих чисел в діапазоні від 1060 до 1393:

$$14 < \pi(1393) - \pi(1060) < 78.$$

Дійсно, $\pi(1060) = 178$, $\pi(1393) = 222$, тобто в зазначеному діапазоні 44 простих числа: 1061, 1063, 1069, 1087, . . . ,1193, 1201, 1213, 1217, 1223, 1229, . . . , 1367, 1373, 1381.

2.5 Розкладання чисел на множники

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

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

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

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

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

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

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

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

Алгоритм ρ-Полларда факторізації цілих чисел

1. Визначаємо початкові значення $a := 2$, $b := 2$.
2. Виконуємо в циклі
$a := (a^2 + 1) \bmod n$; $b := (b^2 + 1) \bmod n$, $b := (b^2 + 1) \bmod n$;
$d := НСД(a - b, n)$.
Якщо $d = 1$, продовжуємо цикл.
Якщо $1 < d < n$, повертаємо «d – дільник числа n».
Якщо $d = n$, алгоритм закінчує роботу невдачею.

Приклад.

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

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

$i = 1$,

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

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

$i = 2$,

$a := (5^2 + 1) \bmod 187 = 26$, $b := (26^2 + 1) = 677 \bmod 187 = 116$,

$b := (116^2 + 1) = 13456 \bmod 187 = 180$;

$d := НСД(26 - 180, 187) = НСД(-154, 187) = 11$.

Звідси $d = 11$ – дільник числа 187.

2.6 Асиметричний алгоритм шифрування RSA

Алгоритм шифрування RSA відноситься до криптографічних систем із відкритим ключем.

Авторами алгоритму RSA, запропонованого в 1977р., є Р.Рівест (Rivest), А.Шамір (Shamir) і А.Адлеман (Adleman). Надійність алгоритму базується на складності факторизації (розкладення на множники) великих чисел.

Алгоритм RSA складається з трьох частин: генерування ключів, зашифрування і дешифрування.

1 Генерування ключів (абонент А).

Оберемо два великих різних простих числа p і q та знайдемо їх добуток

$n = p \cdot q$.

Обчислимо функцію Ейлера $\varphi(n)$ за формулою

$\varphi(n) = \varphi(p \cdot q) = (p - 1) \cdot (q - 1)$.

Закритий ключ d обираємо з умов: $d < \varphi(n)$ і d взаємно просте з $\varphi(n)$, тобто d і $\varphi(n)$ не мають спільних дільників.

Відкритий ключ e обираємо з умов $e < \varphi(n)$ і

$d \cdot e = 1 \bmod \varphi(n)$.

Число e обчислюємо за розширеним алгоритмом Евкліда.

В алгоритмі RSA n – відкритий параметр, e – відкритий ключ, d – секретний ключ.

Секретний ключ d зберігається в секреті, відкритий ключ e та відкритий параметр n надаються користувачеві В для можливості шифрування.

2 Зашифрування (АбонентВ).

Вихідне повідомлення розбивається на блоки M_i однакової довжини. Кожен блок представляється у вигляді великого десяткового числа, менше n, і шифрується окремо. Шифрування блока M (M – десяткове число) здійснюється за наступною формулою

$$M^e = C \bmod n,$$

де C – шифрблок, що відповідає блоку відкритого повідомлення M. Шифрблоки з`єднуються в криптограму.

Криптограма надається користувачеві А.

3 Розшифрування (абонент А).

При розшифруванні криптограма розбивається на блоки відомої довжини і кожен шифрблок розшифрується окремо за наступною формулою

$$C^d = M \bmod n.$$

Доведемо, що $C^d = M \bmod n$.

$$C^d = (M^e)^d = M^{ed} = M^{1+s\cdot\varphi(n)} \bmod n.$$

Покажемо, що

$$M^{1+s\cdot\varphi(n)} = M \bmod n.$$

Якщо M взаємно просте з n, то згідно з теоремою Ейлера

$M^{\varphi(n)} = 1 \bmod n \Rightarrow \quad M^{s\cdot\varphi(n)} = 1 \bmod n \quad \Rightarrow$
$M^{1+s\cdot\varphi(n)} = M \bmod n$.

Нехай M має спільний дільник з n, тобто $M = p \cdot k$, $1 < k < q$. Тоді M взаємно просте з q, і згідно з теоремою Ейлера

$$M^{q-1} = 1 \bmod q \Rightarrow M^{s\cdot(p-1)\cdot(q-1)} = 1 \bmod q \Rightarrow$$

$$M^{s\cdot(p-1)\cdot(q-1)} = 1 + r \cdot q \quad \Rightarrow$$
$$M \cdot M^{s\cdot(p-1)\cdot(q-1)} = M + r \cdot q \cdot (p \cdot k) \quad \Leftrightarrow$$
$$M^{1+s\cdot\varphi(n)} = M \bmod n$$

Співвідношення $C^d = M \bmod n$ доведено.

Приклад 1.

1 Генерування ключів

Оберемо два простих числа $p = 11$ і $q = 17$.

Тоді обчислимо модуль $n = p \cdot q = 11 \cdot 17 = 187$ і функцію Ейлера $\varphi(n)$: $\varphi(n) = (p-1)\cdot(q-1) = 10 \cdot 16 = 160$.

Закритий ключ d обираємо з умов $d < \varphi(n)$ і d взаємно просте з $\varphi(n)$, тобто d і $\varphi(n)$ не мають спільних дільників.

Нехай $d = 83$.

Відкритий ключ e обираємо з умов $e < \varphi(n)$ і $d \cdot e = 1 \bmod \varphi(n)$:

$83 \cdot e = 1 \bmod 160$.

За допомогою розширеного алгоритму Евкліда знаходимо $e = 27$.

В прикладі 1

$n = 187$ – відкритий параметр,

$e = 27$ – відкритий ключ,

$d = 83$ – секретний ключ.

Таблиця 2.2 – Таблиця простих чисел

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997	1009	1013
1019	1021	1031	1033	1039	1049	1051	1061	1063	1069
1087	1091	1093	1097	1103	1109	1117	1123	1129	1151
1153	1163	1171	1181	1187	1193	1201	1213	1217	1223
1229	1231	1237	1249	1259	1277	1279	1283	1289	1291
1297	1301	1303	1307	1319	1321	1327	1361	1367	1373
1381	1399	1409	1423	1427	1429	1433	1439	1447	1451
1453	1459	1471	1481	1483	1487	1489	1493	1499	1511
1523	1531	1543	1549	1553	1559	1567	1571	1579	1583
1597	1601	1607	1609	1613	1619	1621	1627	1637	1657
1663	1667	1669	1693	1697	1699	1709	1721	1723	1733
1741	1747	1753	1759	1777	1783	1787	1789	1801	1811
1823	1831	1847	1861	1867	1871	1873	1877	1879	1889
1901	1907	1913	1931	1933	1949	1951	1973	1979	1987
1993	1997	1999	2003	2011	2017	2027	2029	2039	2053
2063	2069	2081	2083	2087	2089	2099	2111	2113	2129
2131	2137	2141	2143	2153	2161	2179	2203	2207	2213
2221	2237	2239	2243	2251	2267	2269	2273	2281	2287

2 Шифрування

Зашифруємо повідомлення 15, використовуючи відкритий ключ $e = 27$ та відкритий параметр $n = 187$:

$$C = 15^{27} \bmod 187 = 60.$$

Таким чином, вихідному повідомленню 15 відповідає криптограма 60.

3 Розшифрування

Розшифруємо повідомлення 60, користуючись секретним ключем $d = 83$ і відкритим параметром $n = 187$:

$$M = 60^{83} \bmod 187 = 15.$$

В результаті розшифрування було одержано вихідне повідомлення 15.

Таблиця 2.3 – Схема шифрування RSA на прикладі 1

Користувач А	Канал звязку	Користувач В
		$M = 15$
$n = 187$	$n = 187 \rightarrow$	$n = 187$
$e = 27$	$e = 27 \rightarrow$	$e = 27$
$d = 83$		
$C = 60$	$\leftarrow C = 60$	$C = 15^{27} \bmod 187 = 60$
$M = 60^{83} \bmod 187 = 15$		

Приклад 2.

Нехай відомі відкритий ключ $E = 3457$ та відкритий параметр $N = 198691$. Спробуємо знайти секретний ключ D і розшифрувати криптограму $C = 158728$. Розкладемо число $N = 198691$ на множники $N = 431 \cdot 461$ і знайдемо функцію Ейлера $\varphi(N) = 430 \cdot 460 = 197800$.

Секретний ключ $D = 1/3457 \bmod 197800 = 16593$.

Звідси знайдемо відправлене повідомлення $M = 158728^{16593} \bmod 198691 = 41129$. Таким чином, $M = 41129_{10} = A0A9_{16}$. Шістнадцятирічній послідовності '*A*0*A*9' відповідає текст «ай».

При реалізації алгоритму RSA не можна використовувати той самий модуль n для різних користувачів. У випадку спільного модуля n можлива така атака.

2.7 Атака на алгоритм RSA

Нехай в асиметричної криптосистемі з використанням алгоритму шифрування RSA пари ключів – секретний та відкритий – генеруються на спільному модулі n.

Покажемо, що в цьому випадку можлива атака на розкриття секретного повідомлення M.

Нехай користувачі системи A1, A2, …, Ai,… отримали пари ключів – секретний d_i та відкритий e_i, які генеруються на спільному модулі n: $d_i \cdot e_i = 1 \bmod \varphi(n)$.

Нехай далі користувачеві D необхідно передати секретне повідомлення M деякій групі користувачів A1, A2, …, Ai,… асиметричної криптосистеми. Для цього він отримує відкрити ключі e_1, e_2, …, e_i, …. користувачів A1, A2, …, Ai,… відповідно і відкритий параметр – спільний модуль n.

Користувач D зашифровує повідомлення M за алгоритмом RSA: $C_i = M^{e_i} \bmod n$ та відправляє шифрограми C_1, C_2, …, C_i, …. користувачам системи A1, A2, …, Ai,… .

Криптоаналітик має можливість перехопити відкрити ключі e_1, e_2, …, e_i, … користувачів A1, A2, …, Ai,… відповідно і відкритий параметр – спільний модуль n, а також шифрограми C_1, C_2, …, C_i, …..

Нехай ключі e_1 і e_2 взаємно прості: $(e_1, e_2) = 1$. Тоді за алгоритмом Евкліда існують цілі числа x та y такі, що $e_1 \cdot x + e_2 \cdot y = 1$.

Згідно з алгоритмом шифрування RSA маємо порівняння

$$C_1^x \cdot C_2^y \bmod n = M .$$

Дійсно,

$$C1^x \cdot C2^y \bmod n = (M^{e1})^x \cdot (M^{e2})^y \bmod n = M^{e1 \cdot x + e2 \cdot y} = M$$

.

Таким чином, якщо криптоаналітику відомі відкрити ключі, спільний модуль і відповідні шифрограми, то він має можливість провести успішну атаку з розкриттям секретного повідомлення.

Приклад 1.

Нехай криптоаналітику відомі відкрити ключі $e_1 = 5646703$ і $e_2 = 85564645$ користувачів А1 і А2 відповідно, які створені на спільному модулі $n = 12162272489$ за алгоритмом RSA, та відповідні шифрограми $C_1 = 8653343404$ і $C_2 = 3323031838$. Проведемо атаку з розкриттям секретного повідомлення M.

Спочатку перевірімо взаємну простоту ключів e_1 і e_2: (5646703, 85564645) = 1.

Знайдемо x і y, що задовольняють умові $e_1 \cdot x + e_2 \cdot y = 1$:

$x = -22302953,\ y = 1471848$.

Тоді отримуємо повідомлення M в десятковому вигляді згідно з формулою $C_1^x \cdot C_2^y \bmod n = M$:

$$M = 8653343404^{-22302953} \cdot 3323031838^{1471848} \bmod 1216227248\,9 =$$

$$= 5292019530 \cdot 8559378358 \bmod 1216227248\,9 = 1718187621_{10}.$$

$M = 1718187621_{10} = `66697665`_{16}$.

Після декодування маємо

$M = "five"$.

Атака проведена успішно.

Розв'язання рівняння $a \cdot x + b \cdot y = 1$.

Нехай необхідно розв'язати рівняння $a \cdot x + b \cdot y = 1$ для заданих a і b відносно x і y в цілих числах, при цьому $a > b$.

Рівняння $a \cdot x + b \cdot y = 1$ має розв'язок, якщо a і b взаємно прості: $(a, b) = 1$.

Для розв'язання рівняння $a \cdot x + b \cdot y = 1$ використається розширений алгоритм Евкліда.

Згідно з алгоритмом Евкліда знаходимо залишки від ділення

$$a = b \cdot q_0 + r_1$$
$$b = r_1 \cdot q_1 + r_2$$
$$r_1 = r_2 \cdot q_2 + r_3$$
$$r_2 = r_3 \cdot q_3 + r_4$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n$$
$$r_{n-1} = r_n \cdot q_n$$

Заповнюємо дві таблиці за правилами:

положимо $(x_0, x_1) := (1,0)$, $(y_0, y_1) := (0,1)$, далі оновлення значень $(x_0, x_1) := (x_1, x_0 - x_1 \cdot q)$, $(y_0, y_1) := (y_1, y_0 - y_1 \cdot q)$.

Розв'язком рівняння є x_0, y_0.

Розглянемо роботу цього алгоритму на прикладі.

Приклад 2.

Розглянемо рівняння $17 \cdot x + 13 \cdot y = 1$.

Знайдемо залишки від ділення

$$17 = 13 \cdot 1 + 4$$
$$13 = 4 \cdot 3 + 1$$
$$4 = 1 \cdot 4$$

Побудуемо таблиці 2.4 і 2.5.

Таблиця 2.4 – Пошук розв'язку x рівняння $17 \cdot x + 13 \cdot y = 1$

q_i		1	3	4
x_0	1	0	1	**-3**
x_1	0	1	-3	13

Звідси $x = x_0 = -3$.

Звідси $y = y_0 = 4$.

Таким чином,

$x = -3$, $y = 4$.

Таблиця 2.5 – Пошук розв'язку y рівняння $17 \cdot x + 13 \cdot y = 1$

q_i		1	3	4
y_0	0	1	-1	**4**
y_1	1	-1	4	-17

Перевірка : $17 \cdot x + 13 \cdot y = 17 \cdot (-3) + 13 \cdot 4 = -51 + 52 = 1$.

Контрольні питання

1. Що називається розв'язком порівняння першого степеня?
2. За яких умов порівняння першого степеня мають розв'язок?
3. Скільки може бути розв'язків порівняння першого степеня?
4. Якими способами можна розв'язати порівняння першого степеня?
5. В чому полягає розширений алгоритм Евкліда?
6. Для чого використовується алгоритм послідовного зведення в квадрат?
7. Зі скількох кроків складається алгоритм послідовного зведення в квадрат?
8. В яких криптоалгоритмах виникає необхідність підносити числа до степеня за певним модулем?
9. Яке число називається простим?

10. Сформулюйте малу теорему Ферма.
11. Дайте визначення свідка простоти числа.
12. В чому полягає тест Ферма на простоту?
13. Скільки свідків простоти числа необхідно знайти, щоб вважати його простим?
14. Які числа називають числами Кармайкла?
15. Які властивості мають числа Кармайкла?
16. На якому твердженні базується тест Міллера-Рабіна на простоту?
17. Дайте визначення сильного свідка простоти числа.
18. Який зв'язок між сильними свідками простоти і свідками простоти?
19. У чому складається тест Міллера-Рабіна на простоту?
20. Випишіть оцінку числа простих чисел, що не перевищують заданого.
21. Випишіть формулу апроксимації простого числа.
22. Які числа називаються числами Мерсенна, простими числами Мерсенна?
23. У якому випадку число Мерсенна є простим ?
24. У яких криптографічних алгоритмах необхідно використовувати тести на простоту?
25. Що таке факторизація?
26. Стійкість яких криптоалгоритмів базується на задачі факторизації?
27. В чому полягає алгоритм факторизації ρ-Полларда?
28. Яким чином факторизується будь-яке складене число?
29. Чим обмежується діапазон підбору множників в переборному алгоритмі?
30. Опишіть процедуру генерації ключів в RSA.
31. Опишіть процедури зашифрування та розшифрування в RSA.
32. Побудуйте схему криптосистеми з відкритим ключем.
33. Від чого залежить криптостійкість алгоритму RSA?
34. Опишить атаку на алгоритм RSA.

3 ЗАДАЧА ДИСКРЕТНОГО ЛОГАРИФМУВАННЯ

3.1 Елементи теорії груп

Групою $(G,*)$ називається множина G, на якій визначена бінарна операція $*$, що задовольняє умовам:

1) для будь-яких елементів a, b із множини G результат виконання операції $*$ над елементами a, b належить множині G, тобто $a*b \in G$;
2) для будь-яких елементів a, b, c із множини G виконується рівність $a*(b*c)=(a*b)*c$;
3) у множині G існує одиничний (або нейтральний) елемент e: $e*a=a*e=a$;
4) для кожного елемента $a \in G$ існує обернений елемент $a^{-1} \in G$ такий, що результатом виконання операції $*$ над елементами a, a^{-1} є одиничний елемент e: $a*a^{-1}=a^{-1}*a=e$.

З аксіом групи можна вивести, що в групі існує тільки один одиничний елемент. Можна довести також і одиничність оберненого елемента для всякого елемента групи.

Група $(G,*)$ називається *комутативною* (*абелевою*), якщо для будь-яких елементів a, b із множини G виконується рівність

$$a*b=b*a.$$

Якщо в якості бінарної операції визначена операція множення $\times$, то група називається *мультиплікативною*, одиничний елемент $e=1$, обернений елемент позначається a^{-1} або $\frac{1}{a}$: $a\times\frac{1}{a}=1$.

Якщо в якості бінарної операції визначена операція додавання $+$, то група називається *адитивною*, одиничний елемент $e=0$, обернений елемент позначається $-a$: $a+(-a)=0$.

Приклади груп:

1) множина всіх ненульових дійсних чисел щодо операції множення – «мультиплікативна група дійсних чисел» R^*;
2) множина всіх цілих чисел щодо операції додавання – «адитивна група цілих чисел» Z;
3) множина усіх векторів у просторі R^n щодо операції додавання векторів;
4) множина всіх многочленів з дійсними коефіцієнтами щодо операції додавання многочленів;

Групи в прикладах 1-4 мають нескінченно багато елементів.

Група $(G,*)$ називається *скінченою*, якщо число елементів множини G скінчене. У цьому випадку *порядком групи* $(G,*)$ називається число елементів множини G.

Степенем елемента скінченої групи називається результат багатократного виконання операції групи над цім елементом, наприклад, $a^t = \underbrace{a * a * ... * a}_{t}$.

Для адитивної групи степінь a^t елементу a позначається $t \cdot a$.

Порядком елемента скінченої групи називається найменший степінь k, до котрого потрібно піднести елемент, щоб одержати одиничний елемент e: $a^k = e$.

Позначення $ord(a) = k$.

Якщо $ord(a) = k$, то елемент a^t має порядок $k / НСД(k,t)$. Наприклад, якщо $k = 12$, то a^6 має порядок $12 / НСД(12,6) = 2$.

Якщо $ord(a) = k$ і $a^s = e$, то s ділиться на k.

Елемент скінченої групи називається *твірним*, або *примітивним*, елементом групи, якщо його порядок співпадає з порядком групи.

Група називається *циклічною*, якщо в групі є твірний елемент.

В циклічній групі кожен елемент може бути представлений як степінь твірного елемента.

Порядок елемента циклічної групи є дільником порядку групи. Наприклад, якщо $n=18$ є порядком циклічної групи, порядками елементів групи можуть бути тільки числа 1, 2, 3, 6, 9, 18.

Якщо $(G,*)$ – циклічна група порядку n, d – дільник n, то група $(G,*)$ має $\varphi(d)$ елементів порядку d.

Наприклад, якщо $n=30$ і $d=6$, то група $(G,*)$ має $\varphi(6)=2$ елемента порядку 6.

Якщо $(G,*)$ – циклічна група порядку n, то група $(G,*)$ має $\varphi(n)$ твірних елементів. Наприклад, якщо $n=30$, то група $(G,*)$ має $\varphi(30)=\varphi(2\cdot 3\cdot 5)=\varphi(2)\cdot\varphi(3)\cdot\varphi(5)=1\cdot 2\cdot 4=8$ твірних елементів.

Підгрупи

Якщо деяка підмножина H множини G сама утворює групу щодо операції $*$, визначеної в G, то $(H,*)$ називається *підгрупою* групи $(G,*)$.

Наприклад, підмножина парних цілих чисел є підгрупою адитивної групи Z усіх цілих чисел, а підмножина непарних чисел не буде підгрупою цієї групи.

Для того щоб підмножина $H\subset G$ утворювала підгрупу групи $(G,*)$, необхідно и достатньо, щоб виконувалися дві умови:

1) для будь-яких елементів $u, v \in H$ результат виконання операції $*$, визначеної в G, належить множині H, тобто $u * v \in H$;
2) якщо $h \in H$, то й обернений до нього елемент $h^{-1} \in H$.

Зі всяким елементом $a \in G$ можна зв'язати «породжену» ним циклічну підгрупу.

Оскільки операція піднесення елемента групи в степінь не виводить за межі групи, то всі степені елементу належать деякій підгрупі.

Якщо $a \in G$, тоді всі степені a^t елементу a утворюють циклічну підгрупу, породжену елементом a.

Циклічна підгрупа, породжена елементом a порядку k, має порядок k.

Теорема Лагранжа. Порядок будь-якої підгрупи скінченої групи є дільником порядку групи.

Наприклад, якщо $n = 18$ є порядком циклічної групи, порядками підгруп можуть бути тільки числа 1, 2, 3, 6, 9, 18.

Множина Z_n

Нехай n - натуральне число.

Введемо на множині цілих чисел Z операцію порівняння за модулем n: $a \bmod n$, $a \in Z$. Операція порівняння за модулем n розбиває множину Z на класи еквівалентності, що відповідають залишкам від ділення цілих чисел на n: 0, 1, 2, … , $n-1$.

Множина класів еквівалентності за модулем n утворює множину Z_n.

Всі арифметичні операції { +, –, ×, : } в Z_n виконуються за модулем n.

Приклад. Розглянемо множину $Z_{25} = \{0,1,2,3...,24\}$. Для елементів цієї множини маємо $13+16 = 29 \bmod 25 = 4$, $13 \times 4 = 52 \bmod 25 = 2$, $5+8 = 13 \bmod 25 = 13$.

Адитивно оберненим елементом елемента $a \in Z_n$ за модулем n називається елемент $-a$, такий що $a + (-a) = 0 \bmod n$.

Множина Z_n з операцією додавання + за модулем n утворює скінчену адитивну групу порядку n з нейтральним елементом 0.

Для неї виконуються всі умови групи.

Число твірних елементів адитивної групи Z_n дорівнює $\varphi(n)$.

Приклад 1.

Множина Z_9 з операцією додавання + за модулем 9 утворює скінчену адитивну групу порядку 9 з нейтральним елементом 0: $Z_9 = \{0,1,2,3,4,5,6,7,8\}$. Підгрупами цієї групи є {0}, {0,3,6}. Твірними елементами групи є 1, 2, 4, 5, 7 і 8. Елементи 3 і 6 мають порядок 3.

Мультиплікативно оберненим елементом елемента $a \in Z_n$ за модулем n називається елемент $a^{-1} \in Z_n$, такий що $a \times a^{-1} = 1 \bmod n$.

Елемент $a \in Z_n$ називається *оборотним* за модулем n, якщо для нього існує обернений.

Ділення $\frac{a}{b}$ в Z_n визначається як множення на мультиплікативно обернений елемент: $\frac{a}{b} = a \times b^{-1} \bmod n$, якщо дільник b оборотний.

Приклад 2.

У множині Z_{25} елемент $a = 2$ має обернений $\frac{1}{2} = 13$, тому що порівняння $2 \times x = 1 \bmod 25$ дає розв'язок $x = 13$. Звідси $\frac{15}{2} = 15 \times \frac{1}{2} = 15 \times 13 = 195 \bmod 25 = 20$.

Елемент $a \in Z_n$ є мультиплікативно оборотним в тому і тільки в тому випадку, коли a і n взаємно прості: $НСД(a, n) = 1$.

Приклад 3.

Мультиплікативно оборотними елементами в множині Z_9 є елементи 1,2,4,5,7,8. Для визначення, наприклад, $\frac{1}{5} \bmod 9$ розв'яжемо порівняння $5 \times x = 1 \bmod 9$, що дає розв'язок $x = 2$. Таким чином, $\frac{1}{5} \bmod 9 = 2$.

Введемо множину Z_n^*, визначену як підмножина множини Z_n, яка складається тільки із оборотних елементів

$$Z_n^* = \{a \in Z_n \mid (a, n) = 1\}.$$

Множина Z_n^* з операцією множення × за модулем n утворює скінчену мультиплікативну групу порядку $\varphi(n)$ з нейтральним елементом 1.

Приклад 4.

Множина Z_9^* з операцією множення × за модулем 9 утворює скінчену мультиплікативну групу порядку $\varphi(9) = 6$ з нейтральним елементом 1: $Z_9^* = \{1,2,4,5,7,8\}$. Підгрупами цієї групи є $\{1\}$, (1,8), $\{1,4,7\}$. Твірними елементами групи є елементи 2 і 5.

Група Z_n^* має твірний елемент (тобто є циклічною) у тому і тільки тому випадку, коли $n = 1,4, p^k, 2p^k$, p – просте непарне число.

Якщо Z_n^* – циклічна група, то число твірних елементів групи дорівнює $\varphi(\varphi(n))$.

Якщо a – твірний елемент Z_n^*, то

$Z_n^* = \{a^i \bmod n,\ i = 0,1,2,...,\varphi(n)-1\}$.

Якщо a – твірний елемент Z_n^*, то $b = a^i \bmod n$ є твірним елементом у тому і тільки тому випадку, коли i взаємно просте з $\varphi(n)$: $(i,\varphi(n)) = 1$.

Елемент $a \in Z_n^*$ є твірним елементом групи в тому і тільки тому випадку, коли

$$a^{\frac{\varphi(n)}{k}} \neq 1 \bmod n$$

для всіх простих дільників k числа $\varphi(n)$.

Якщо n – просте число, то $\varphi(n) = n-1$ і $Z_n^* = \{1,2,...n-1\}$.

Приклад 5.

Мультиплікативна група Z_{25}^* має $\varphi(25)=20$ елементів. Група циклічна, містить $\varphi(\varphi(25))=\varphi(20)=8$ твірних елементів.

Мультиплікативна група Z_{13}^* має $\varphi(13)=12$ елементів: $Z_{13}^*=\{1,2,3,4,5,6,7,8,9,10,11,12\}$. Група циклічна, містить $\varphi(\varphi(13))=\varphi(12)=4$ твірних елемента.

Елемент $a=2$ є твірним елементом групи Z_{13}^*, тому що виконуються умова $a^{\frac{\varphi(n)}{k}} \neq 1 \bmod n$ для простих дільників 2 і 3 числа 12: $2^{\frac{12}{2}}=2^6=64 \bmod 13=12\neq 1$ і $2^{\frac{12}{3}}=2^4=16 \bmod 13=3\neq 1$.

Знайдемо всі твірні елементи групи Z_{13}^*.

Для цього знайдемо множину індексів $\{i \mid (i,12)=1\}=\{1,5,7,11\}$ і обчислимо за формулою $b=2^i \bmod 13$ твірні елементи $a=2^1 \bmod 13=2$, $b=2^5=32 \bmod 13=6$, $c=2^7=128 \bmod 13=11$, $d=2^{11}=2028 \bmod 13=7$.

Твірними елементами групи Z_{13}^* є 2, 6, 7, 11.

Підгрупами цієї групи є {1}, {1,12}, {1,3,9}, {1,5,8,12}, {1,3,4,9,10,12}. Підгрупи {1,5,8,12}, {1,3,4,9,10,12} також містять підгрупи відповідно {1,12} і {1,3,9}.

Порядок 1 має елемент 1; порядок 2 має елемент 12; порядок 3 мають елементи 3, 9; порядок 4 мають елементи 5, 8; порядок 6 мають елементи 4, 10; порядок 12 мають твірні елементи 2, 6, 7, 11.

3.2 Методи розв'язання задачі дискретного логарифмування

Розглянемо задачу дискретного логарифмування в мультиплікативної групі залишків за простим модулем p:

при заданих значеннях параметрів a, b, p, де p – просте число, a, b – елементи мультиплікативної групі, $1 < a, b < p$, знайти x, що задовольняє порівнянню

$$a^x = b \bmod p.$$

Порядком елемента групи a за модулем p називається найменший степінь, в який потрібно піднести число a, щоб одержати 1 за модулем p.

Позначення $ord(a)$.

Порядок елемента a за простим модулем p є дільником числа $p-1$.

Максимальний порядок елемента a за простим модулем p дорівнює $p-1$.

Задача дискретного логарифмування $a^x = b \bmod p$ має єдиний розв'язок x, $1 \le x \le p-1$, якщо порядок елемента a дорівнює $p-1$, $ord(a) = p-1$.

Якщо $ord(a) < p-1$, задача дискретного логарифмування $a^x = b \bmod p$ може не мати розв'язків.

Алгоритм великих - малих кроків (Baby-step giant-step)

Алгоритм великих - малих кроків (алгоритм Даніеля Шенкса, 1972 р.) призначений для розв'язання задачі дискретного логарифмування в мультиплікативної групі залишків за простим модулем p:

$$a^x = b \bmod p.$$

Нехай n – порядок елемента a.

1. Обчислюємо $m = \left\lceil \sqrt{n} \right\rceil$, де $\lceil u \rceil$ – округлення u з надлишком.
2. Будуємо таблицю

j	0	1	2	…	$m-1$	m
$a^j \bmod p$	1	a	a^2	…	a^{m-1}	a^m

3. Обчислюємо $t = a^{-m} \bmod p$.
4. Задаємо $\gamma := b$.
5. Цикл по i від 0 до $m-1$

   5.1 перевіряємо, чи існує j таке, що $a^j \bmod p = \gamma$;

   5.2 якщо існує j: $a^j \bmod p = \gamma$, то розв'язком є $x = m \cdot i + j$;

   5.3 інакше $\gamma := \gamma \cdot t$.

Приклад 1.

Знайдемо розв'язок задачі дискретного логарифмування $10^x = 4 \bmod 23$, $n = 22$, за допомогою алгоритму Шенкса.

Значення $m = \left\lceil \sqrt{22} \right\rceil = 5$.

Побудуємо таблицю

j	0	1	2	3	4	5
$10^j \bmod 23$	1	10	8	11	18	19

Обчислимо $t = 10^{-5} \bmod 23 = \frac{1}{19} \bmod 23 = \frac{24}{-4} = -6$.

Положимо $\gamma := 4$.

$i = 0$, $\gamma := 4 \cdot (-6) = -24 \bmod 23 = -1$;

$i = 1$, $\gamma := (-1) \cdot (-6) = 6$;

$i = 2$, $\gamma := 6 \cdot (-6) = -36 \bmod 23 = 10$;

$i = 3$, $j = 1$, $x = 3 \cdot 5 + 1 = 16$.

Розв'язком задачі дискретного логарифмування $10^x = 4 \bmod 23$ є $x = 16$.

?

Перевірка. $10^{16} = 4 \bmod 23$

Згідно таблицею в Прикладі 1,

$10^5 \bmod 23 = 19 \bmod 23 = -4$.

Тоді

$10^{10} \bmod 23 = (-4)^2 \bmod 23 = 16 \bmod 23 = -7$,

$10^{15} \bmod 23 = 10^{10} \cdot 10^5 = (-7) \cdot (-4) \bmod 23 = 28 \bmod 23 = 5$,

$10^{16} \bmod 23 = 10^{15} \cdot 10^1 = 5 \cdot 10 \bmod 23 = 50 \bmod 23 = 4$.

Розв'язок задачі знайдено вірно.

Алгоритм Нечаєва - Полліга - Хеллмана

Нехай n – порядок числа a.

1. Число n представимо у вигляді $n = q \cdot r$, q – просте число.

2. Обчислимо значення $c = a^r \bmod p$ і $d = b^r \bmod p$.

3. За допомогою алгоритму великих - малих кроків знайдемо розв'язок y задачі $c^y = d \bmod p$, $ord(c) = q$.

6. Обчислимо $t = a^{-y} \bmod p$.

4. Обчислимо значення $u = a^q \bmod p$ і $v = b \cdot t \bmod p$.

5. За допомогою алгоритму великих - малих кроків знайдемо розв'язок z задачі $u^z = v \bmod p$, $ord(u) = r$.

6. Розв'язком задачі $a^x = b \bmod p$ є $x = z \cdot q + y$.

Дійсно,

$$a^x \bmod p = a^{z \cdot q + y} = a^{z \cdot q} \cdot a^y = (a^q)^z \cdot \frac{1}{t} = u^z \cdot \frac{b}{v} = v \cdot \frac{b}{v} = b.$$

Приклад 2.

Знайдемо розв'язок задачі дискретного логарифмування $10^x = 4 \bmod 23$, $n = 22$,
за допомогою алгоритму Нечаєва - Полліга – Хеллмана.

Розкладемо порядок $n = 22 = 11 \cdot 2$, $q = 11$, $r = 2$.

Обчислимо значення $c = 10^2 \bmod 23 = 8$, $d = 4^2 \bmod 23 = 16$.

Знайдемо розв'язок задачі $8^y = 16 \bmod 23$, $ord(8) = 11$: $y = 5$.

Обчислимо $t = 10^{-5} \bmod 23 = -6$.

Обчислимо значення

$u = 10^{11} \bmod 23 = 22$, $v = 4 \cdot (-6) \bmod 23 = 22$.

Знайдемо розв'язок задачі $22^z = 22 \bmod 23$, $ord(22) = 2$.

Зрозуміло, $z = 1$.

Звідси $x = 1 \cdot 11 + 5 = 16$.

Розв'язком задачі дискретного логарифмування $10^x = 4 \bmod 23$ за допомогою алгоритму Нечаєва – Полліга – Хеллмана є також

$x = 16$.

3.3 Схема Діффі-Хеллмана в мультиплікативній групі

Схема Діффі-Хеллмана призначена для формування спільного секретного ключа по відкритому каналу зв'язку між двома абонентами мережі.

Розглянемо схему Діффі-Хеллмана в групі залишків за простим модулем.

Два абонента А і В обирають спільні відкриті параметри – велике просте число p та генератор g мультиплікативної групи залишків за простим модулем p.

Абоненти А і В обмінюються спільними відкритими параметрами – p, g – по відкритому каналу мережі.

Далі абонент А обирає випадкове таємне число k, $1 < k < p$, та обчислює значення y за формулою:

$$y = g^k \bmod p .$$

Абонент В обирає випадкове таємне число m, $1 < m < p$, та обчислює значення z за формулою:

$$z = g^m \bmod p .$$

Абоненти А і В обмінюються значеннями y та z по відкритому каналу мережі.

Після отримання значення z абонент А обчислює значення U за формулою:

$$U = z^k \bmod p ,$$

а абонент В після отримання значення y обчислює значення V за формулою:

$$V = y^m \bmod p .$$

Знайденні абонентами А і В значення U та V співпадають:

$$U = z^k \bmod p = (g^m)^k = (g^k)^m = y^m \bmod p = V$$

і створюють спільний секретний ключ.

Крипостійкість схеми Діффі-Хеллмана заснована на важкорозв'язаності задачі дискретного логарифмування.

Дійсно, для отримання спільного секретного ключа абонентів А і В по перехопленим значенням y та z необхідно знайти таємні k або m. Для цього достатньо розв'язати одну з задач $g^k = y \bmod p$ або $g^m = z \bmod p$ відносно k або m відповідно. Якщо значення k (або m) знайдено, спільний секретний ключ обчислюється за формулою $U = z^k \bmod p$ ($V = y^m \bmod p$).

Приклад 1.

Нехай відкритими параметрами схеми Діффі-Хеллмана є $p = 23$, $g = 10$. Знайдемо спільний секретний ключ для двох користувачів.

Нехай абонент А обрав випадкове таємне число $k = 5$ та обчислив значення $y = 19$, абонент В обрав випадкове таємне число $m = 12$ та обчислив значення $z = 13$.

Абоненти А і В здійснили обмін значеннями y та z по відкритому каналу мережі.

Після цього абонент А обчислив значення $U = 13^5 \bmod 23 = 4$, а абонент В обчислив значення $V = 19^{12} \bmod 23 = 4$.

Таким чином, кожен із абонентів А і В сформував спільний секретний ключ, рівний 4.

Таблиця 3.1 – Схема формування спільного секретного ключа Діффі-Хеллмана на прикладі 1

Користувач А	Канал звязку	Користувач В
$p = 23$, $g = 10$	$\leftrightarrow$	$p = 23$, $g = 10$
$k = 5$		$m = 12$
$y = 10^5 \bmod 23 = 19$	$y = 19 \rightarrow$	$y = 19$
$z = 13$	$\leftarrow z = 13$	$z = 10^{12} \bmod 23 = 13$
$U = 13^5 \bmod 23 = 4$		$V = 19^{12} \bmod 23 = 4$

Приклад 2.

Нехай відкритими параметрами схеми Діффі-Хеллмана є $p = 23$, $g = 10$. Зловмисник перехопив значення $y = 20$ та $z = 11$ з відкритого каналу мережі і бажає знайти спільний секретний ключ абонентів А і В.

Для отримання спільного секретного ключа абонентів А і В по перехопленим значенням y та z зловмисник розв'язує задачу

дискретного логарифмування $10^k = 20 \bmod 23$: $k = 9$. Звідси $U = 11^9 \bmod 23 = 19$.

Знайдено спільний секретний ключ абонентів А і В, рівний 19.

Якщо зловмисник знаходив значення m, тобто розв'язував задачу $10^m = 11 \bmod 23$, $m = 3$, то знайдене значення $V = 20^3 \bmod 23$ також дорівнює 19.

Таблиця 3.2 – Схема перехоплення та обчислення секретного ключа користувачів А і В на прикладі 2

Користувач А	Канал звязку	Користувач В
$p = 23 \;\; g = 10$	$\leftarrow p = 23 \;\; g = 10 \rightarrow$	$g = 10 \;\; p = 23$
$y = 10^k \bmod 23 = 20$	$y = 20 \rightarrow$	$y = 20$
$z = 11$	$\leftarrow z = 11$	$z = 10^m \bmod 23 = 11$
	$10^k = 20 \bmod 23$: $k = 9$	
	$U = 11^9 \bmod 23 = 19$	
	$\uparrow$ Зловмисник Z	

Контрольні питання

1. Дайте визначення групи, адитивної, мультиплікативної.
2. Дайте визначення порядку групи.
3. Що таке порядок елементу групи?
4. Яка група називається комутативною?
5. Дайте визначення циклічної групи.
6. Дайте визначення підгрупи.
7. Дайте визначення циклічної підгрупи.
8. Який елемент групи називається твірним?

9. Які властивості твірних мультиплікативної групи залишків за модулем простого числа?
10. Сформулюйте теорему Лагранжа.
11. Сформулюйте задачу дискретного логарифмування.
12. Сформулюйте умови розв'язання задачі дискретного логарифмування.
13. Опишіть алгоритм великих - малих кроків.
14. В яких криптографічних алгоритмах необхідно розв'язувати задачу дискретного логарифмування?
15. Назвіть методи розв'язання задачі дискретного логарифмування.
16. Опишіть схему Діффі-Хеллмана генерування спільного секретного ключа.
17. На чому заснована криптографічна стійкість схеми Діффі-Хеллмана?
18. Як впливає розмір простого числа N на криптостійкість схеми Діффі-Хеллмана?
19. Як визначити бітовий розмір спільного ключа по вхідних відкритих параметрах?

4 ЦИФРОВИЙ ПІДПИС

В розділі розглядаються одні з перших алгоритмів цифрового підпису, а саме: на базі алгоритму RSA та ЕльГамаля. Розглянуто також види атак на цифровий підпис.

В перших алгоритмах цифрового підпису формування підпису залежало від електронного документу безпосередньо, в сучасних стандартах перед підписанням електронний документ хешується. В розділі розглянуто поняття функції хешування і атака на базі парадокса днів народження.

4.1 Функції хешування

Функцією хешування (хеш-функцією) називається перетворення даних, що переводить рядок бітів M довільної довжини в рядок бітів $H(M)$ деякої фіксованої довжини (кілька десятків чи сотень біт).

Хеш-функція $H(M)$ повинна задовольняти наступним умовам:

- хеш-функція $H(M)$ повинна бути чутливою до будь-яких змін вхідної послідовності M;
- для даного значення $H(M)$ повинно бути неможливо знайти значення M;
- для даного значення $H(M)$ повинно бути неможливо знайти значення $M' \neq M$ таке, що $H(M') = H(M)$.

Ситуація, за якої для різних вхідних послідовностей M, M' співпадають значення їх хеш-образів: $H(M') = H(M)$, називається колізією.

При побудові хеш-образу, вхідна послідовність M розбивається на блоки M_i, $i = 1,...,k$, фіксованої довжини і оброблюється поблокове за формулою

$h_i = F(h_{i-1}, M_i)$,

h_0 – довільний початковий вміст, $F(\bullet,\bullet)$ – задана функція.

Хеш-значення, що обчислюється при введенні останнього блоку повідомлення, стає хеш-значенням всього повідомлення M:

$H(M) = h_k$.

В якості прикладу розглянемо спрощений варіант хеш-функції із рекомендацій МККТТ X.509:

$$h_i = \left(h_{i-1} + M_i\right)^2 \bmod n,\ H(M) = h_k.$$

де $n = p \cdot q$, p і q – великі прості числа, h_0 – довільний початковий вміст, M_i – i-тий блок повідомлення $M = M_1 \| M_2 \| ... \| M_k$.

В сучасних стандартах цифрового підпису використовуються таки функції хешування MD5 (128 біт), RIPEMD-160, SHA-1 (160 біт), SHA-256, SHA-384, SHA-512, ГОСТ Р 34.11-94, ГОСТ 34.311-95, ГОСТ Р 34.11-2012, ДСТУ 7564-2014, СТБ 34.101.31 та інші.

4.2 Парадокс днів народження

Задача. «Яке повинна бути мінімальна кількість осіб у групі, щоб, принаймні, у двох з них збіглися дні народження з імовірністю, більшої $\frac{1}{2}$?»

Нехай n – число різних днів народжень (n = 365), r – кількість осіб у групі.

Усього комбінацій для r осіб дорівнює n^r. З них комбінацій з незбіжними значеннями

$$n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-(r-1)).$$

Звідси ймовірність того, що в групі не буде осіб з однаковими днями народження:

$$\text{P}\{\text{не збігу}\} = \frac{n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(n-(r-1))}{n^r} =$$

$$= 1\cdot\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{r-1}{n}\right).$$

При великих n

$$\text{P}\{\text{не збігу}\} \approx$$

$$\approx 1-\frac{1}{n}-\frac{2}{n}\ldots-\frac{r-1}{n} = 1-\frac{1+2+\ldots+(r-1)}{n} = 1-\frac{(r-1)r}{2n}$$

Звідси

$$\text{P}\{\text{збігу}\} = \frac{(r-1)r}{2n},$$

За умовою задачі маємо $r > \sqrt{n}$:

$$\frac{(r-1)r}{2n} > 1/2 \Rightarrow (r-1)r > n \Rightarrow r > \sqrt{n}.$$

При $n = 365$ кількість осіб у групі $r > 20$.
Більш точно $r \geq 23$:
За допомогою коду

```
n:=365;
for r from 20  to 50 do
  P:=1.:
  for i from 1  to r-1 do P:=P*(1-i/n) end
do:
  print(r,1-P):
end do:
```

можна обчисліть точне значення ймовірності збігу днів народження в групі зі *r* осіб:

r	P{збігу}
20	0.411
21	0.443
22	0.475
23	0.507
24	0.538
25	0.568

4.3 Атака на геш-функцію на базі «Парадокса днів народження»

Криптоаналітик підготовлює два документи M і M'.

За допомогою певного генератора він формує дві множини модифікованих документів. Перша множина відповідає за сенсом документу M, друга – M'.

Криптоаналітик обчислює геш-образи документів першій та другій множин відповідно.

Якщо знайдено збіг значень геш-образів документів із різних множин, задача криптоаналітика виконано.

Кількість r необхідних документів обчислюється за формулою

$$r \geq 2^{l/2},$$

де l – кількість двійкових розрядів відповідної геш-функції.

4.4 Електронний цифровий підпис

Цифровий підпис у цифрових документах грає ту ж роль, що і підпис, поставлений від руки в документах на папері: це дані, що приєднуються до повідомлення, котре передається, та підтверджують, що власник підпису склав чи завірив це повідомлення. Отримувач повідомлення за допомогою цифрового підпису може перевірити, що автором повідомлення є саме власник

підпису і що в процесі передачі не було порушено цілісність отриманих даних.

При розробці механізму цифрового підпису виникають наступні задачі:

- створити підпис таким чином, щоб її неможливо було підробити;
- мати можливість перевірки того, що підпис дійсно належить вказаному власнику;
- мати можливість запобігти відмові від підпису.

Класична схема створення цифрового підпису полягає в наступному.

Підписант має асиметричну пару ключів: секретний та відкритий. Цифровий підпис формується за допомогою хеш-образа електронного документу з використанням секретного ключа. Свій відкритий ключ підписант відправляє на замовлення особі, що має перевірити підпис.

Отримувач підписаного повідомлення застосовує до нього хеш-функцію, далі з використанням відкритого ключа підписанта обчислює перевірочне співвідношення. Якщо перевірочне співвідношення виконується, підпис признається справжнім

4.5 Схема цифрового підпису на базі RSA

Криптосистема з відкритим ключем RSA (див. 2.6) може використовуватись не лише для шифрування, але і для побудови схеми цифрового підпису.

Для створення підпису повідомлення M відправник

1. обчислює хеш-образ $r = h(M)$ повідомлення M за допомогою деякої хеш-функції;
2. зашифровує отриманий хеш-образ r на своєму секретному ключі (d, n), тобто обчислює значення $s = r^d \bmod n$, що і є підписом.

Для перевірки підпису отримувач

1. розшифровує підпис s на відкритому ключі (e,n) відправника, тобто обчислює $r' = s^d \bmod n$ і таким чином відновлює імовірний хеш-образ r' повідомлення M;
2. обчислює хеш-образ $h(M) = r$ повідомлення M за допомогою тієї ж самої хеш-функції, котру використовував відправник;
3. порівнює отримані значення r і r'. Якщо вони співпадають, то підпис правильний, відправник дійсно є тим, за кого себе видає, і повідомлення не було змінено при передачі.

4.6 Алгоритм підпису ЕльГамаля

Алгоритм ЕльГамаля є одним зі перших алгоритмів цифрового підпису. Цифровий підпис створюється за допомогою операцій в мультиплікативної групи залишків за модулем великого простого числа. Алгоритм ЕльГамаля не передбачає хешування повідомлення.

Відкритими параметрами алгоритму є випадкове просте число p та число g – генератор мультиплікативної групи залишків за модулем p. Секретним ключем підписанта А є число x, його відкритим ключем є число $y = g^x \bmod p$. Криптостійкість алгоритму підпису ЕльГамаля базується на обчислювальної складності задачі дискретного логарифмування. Для забезпечення криптостійкості цифрового підпису параметри алгоритму має бути досить великими: 1024 або 2048 значущих бітів.

Цифровий підпис формується абонентом А за допомогою його секретного ключа, перевірка підпису здійснюється абонентом В з використанням відкритого ключа абонента А.

Загальносистемні параметри

p – випадкове просте число; g – генератор мультиплікативної групи залишків за модулем p.

Генерація ключів

Згенеруємо секретний та відкритий ключі абонента А.

Оберемо випадкове число x, $2 \le x \le p-2$.

Обчислимо $y = g^x \bmod p$.

Секретним ключем абонента А є число x.

Відкритим ключем абонента А є число y.

Формування цифрового підпису

Підписанту А необхідне підписати повідомлення M.

Він обирає випадкове число k, $2 \le k \le n-2$.

Далі підписант А обчислює перший

$r = g^k \bmod p$

та другий елементи підпису

$$s = \frac{M - x \cdot r}{k} \bmod (p-1).$$

(Число s не повинно бути 0.)

Цифровим підписом є пара чисел $\langle r, s \rangle$.

Перевірка цифрового підпису

Для перевірки підписаного абонентом А повідомлення $\{M, \langle r, s \rangle\}$ отримувач – абонент В – використовує відкрити загальносистемні параметри алгоритму ЕльГамаля – p, g – та відкритий ключ y підписанта А.

Якщо виконуються умови:

$r < p$,

$y^r \cdot r^s = g^M \bmod p$,

підпис $\{M, \langle r, s \rangle\}$ признається справжнім.

Приклад.

Нехай відкритими параметрами алгоритму ЕльГамаля є $p = 23$, $g = 5$.

Для генерування асиметричної пари ключів абонента А виберемо випадкове число $x = 12$ та обчислимо число

$$y = g^x \bmod p = 5^{12} \bmod 23 = 18.$$

Відкритим ключем абонента А є $x = 12$.

Секретним ключем абонента А є $y = 18$.

Нехай абоненту А необхідне підписати повідомлення $M = 17$.

Абонент А обирає випадкове число $k = 19$ та обчислює перший

$$r = g^k \bmod p = 5^{19} \bmod 23 = 7$$

та другий елементи підпису

$$s = \frac{M - x \cdot r}{k} \bmod (p-1) = \frac{17 - 12 \cdot 7}{19} \bmod 52 = 15.$$

з використанням секретного ключа $x = 12$.

Цифровим підписом є пара чисел $\langle r = 7, s = 15 \rangle$.

Для перевірки підписаного абонентом А повідомлення $\{M = 17, \langle r = 7, s = 15 \rangle\}$ використовуються відкрити загальносистемні параметри алгоритму ЕльГамаля $p = 23$, $g = 5$ та відкритий ключ $y = 18$ абонента А.

Оскільки виконуються умови:

$r < p$, тобто $7 < 23$,

$y^r \cdot r^s = g^M \bmod p$,

тобто $y^r \cdot r^s = 18^7 \cdot 7^{15} \bmod 23 = 15$ і $g^M = 5^{17} \bmod 23 = 15$;

підпис $\{M = 17, \langle r = 7, s = 15 \rangle\}$ признається справжнім.

4.7 Основні види атак та загроз цифрового підпису

Атаки на цифровий підпис та загрози наведено в порядку

складності. Найбільш надійними є схеми електронного цифрового підпису, стійкі проти екзистенційної підробки на основі адаптивної атаки.

1. *Атака на основі відомого відкритого ключа* (key-only attack) – сама слабка з атак, практично завжди доступна криптоаналітику;
2. *Атака на основі відомих підписаних повідомлень* (known-message attack) – у розпорядженні криптоаналітика є якесь (поліноміальне від k) число пар (повідомлення, підпис), при цьому криптоаналітик не може впливати на вибір повідомлення;
3. *Проста атака з вибором підписаних повідомлень* (generic chosen-message attack) – криптоаналітик має можливість вибрати деяку кількість підписаних повідомлень, при цьому відкритий ключ він одержує після цього вибору;
4. *Спрямована атака з вибором повідомлень* (directed chosen-message attack) – вибираючи підписані повідомлення, криптоаналітик знає відкритий ключ;
5. *Адаптивна атака з вибором повідомлень* (adaptive chosen-message attack) – криптоаналітик знає відкритий ключ, вибір кожного наступного підписаного повідомлення він може робити на основі знання припустимого підпису попереднього обраного повідомлення.
6. *Атака на зв'язаних ключах* (related-key attack) – атакуючий відомий деякий математичний зв'язок між ключами.

Основні види загроз

1. *Екзистенціальна підробка* (existential forgery) – створення криптоаналітиком підпису для какого-н., можливо безглуздого, повідомлення, відмінного від перехопленого.
2. *Селективна підробка* (existential forgery) – створення підпису для заздалегідь обраного повідомлення.
3. *Універсальна підробка* (universal forgery) – знаходження ефективного алгоритму формування підпису, функціонально еквівалентного запропонованій схемі.

4. *Повне розкриття* (total break) – обчислення секретного ключа, що дає можливість формувати підпису для будь-яких повідомлень.

4.8 Атаки на слабкі підписи

Під слабкими схемами електронного цифрового підпису розуміють ті з них, які допускають підробку підпису. Це означає формування справжнього підпису деякого апріорі заданого повідомлення без знання секретного ключа. Розглянуті нижче приклади слабких схем електронного цифрового підпису складені в близькій аналогії до побудови стійких схем підпису. Однак деякі зовнішні незначні відмінності вносять неприпустиму слабість – можливість формування потенційним порушником (який не знає секретного ключа) справжнього підпису до заданого значення h.

Розглянемо приклади схем цифрового підпису, для яких можна зробити підробку.

Приклад 1.

Схема підпису

Загальносистемні параметри

Просте число p, g – генератор мультиплікативної групи залишків за модулем p, H – функція хешування.

Генерація ключів

Підписант має асиметричну пару ключів: особистий $x : 1 < x < p$ та відкритий y: $y = g^x \bmod p$.

Формування цифрового підпису

Нехай підписант має підписати електронний документ M з хеш-образом $H(M)$. Молодші $|p|-1$ розряди хеш-образу $H(M)$ формують десяткове число h, яке використається при обчисленні цифрового підпису.

Підписант обирає одноразовий випадковий секретний ключ k $1 < k < p-1$, обчислює перший елемент підпису $r = g^k \bmod p$. Число r повинно бути взаємно простим з $(p-1)$. Якщо r і $(p-1)$ не взаємно прості, обирають інше k.

Другій елемент підпису $s = \dfrac{k - x \cdot h}{r} \bmod (p-1)$.

Справжній підпис $\langle r, s \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle r, s \rangle$ під електронним документом M здійснюється за допомогою відкритого ключа y підписанта.

Для перевірки обчислюються хеш-образ документу $H(M)$ та відповідне десяткове число h_M.

Якщо $r = g^{r \cdot s} \cdot y^{h_M} \bmod p$, цифровий підпис електронного документу M признається справжнім.

Підробка

«Справжній» підпис $\langle R, S \rangle$ під повідомленням T з хеш-образом $H(T)$ та відповідним значенням h_T від імені підписанта, якій має відкритий ключ y, формується таким чином.

Криптоаналітик обирає довільне ціле число z, $1 < z < p-1$, таке що $\gcd(z, p-1) = 1$, де $\gcd(a, b)$ – найбільш спільний дільник чисел a й b, та обчислює $R = g^z \cdot y^{h_T} \bmod p$.

Якщо $\gcd(R, p-1) = 1$, підпис обчислюється за формулою $S = \dfrac{z}{R} \bmod (p-1)$.

Підробкою підпису є пара $\langle R, S \rangle$.

Якщо умова $\gcd(R, p-1) = 1$ не виконується, Криптоаналітик обирає інше число z.

Покажемо, що підроблений підпис $\langle R, S \rangle$ задовольняє формулі перевірки підпису:

$$R = g^z \cdot y^{h_T} \bmod p = g^{R \cdot S} \cdot y^{h_T} \bmod p$$

В цьому прикладі наведена атака на основі відомого відкритого ключа. Криптоаналітик здійснив універсальну підробку підпису.

Приклад 2.

Схема підпису

Загальносистемні параметри

Просте число p, таке що $p = cq + 1$, q – просте число, $q > 2^{64}$, g – генератор підгрупи порядку q мультиплікативної групи залишків за модулем p, H – функція хешування.

Генерація ключів

Підписант має асиметричну пару ключів: особистий $x : 1 < x < p$ та відкритий y: $y = g^{-x} \bmod p$.

Формування цифрового підпису

Нехай підписант має підписати електронний документ M з хеш-образом $H(M)$. Молодші $|p|-1$ розряди хеш-образу $H(M)$ формують десяткове число h_M, яке використається при обчисленні цифрового підпису.

Підписант обирає одноразовий випадковий секретний ключ k, $1 < k < q$, обчислює число

$$r = (g^k \bmod p) \bmod q.$$

Підпис $s = (k + x \cdot r \cdot h_M) \bmod q$.

Справжній підпис $\langle r, s \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle r, s \rangle$ під електронним документом M здійснюється за допомогою відкритого ключа y підписанта.

Для перевірки обчислюються хеш-образ документу $H(M)$ та відповідне десяткове число h_M .

Якщо $(y^{r \cdot h_M} \cdot g^s \bmod p) \bmod q = r$, цифровий підпис електронного документу M признається справжнім.

Підробка

Передбачається, що в розпорядженні криптоаналітика є два справжніх підпису $\langle r, s_1 \rangle$, $\langle r, s_2 \rangle$ повідомлень $M1$, $M2$ з хеш-образами $H(M1)$, $H(M2)$ та відповідними значеннями h_1, h_2, $h_1 \neq h_2$, які сформовано підписантом А.

Для значень s_1 та s_2 виконується співвідношення

$$s_1 = (k + x \cdot r \cdot h_1) \bmod q,$$
$$s_2 = (k + x \cdot r \cdot h_2) \bmod q.$$

Звідси знайдемо секретний ключ підписанта А

$$x = \frac{s_1 - s_2}{r \cdot (h_1 - h_2)} \bmod q.$$

В цьому прикладі наведена атака на основі відомих підписаних повідомлень та на зв'язаних ключах.

Криптоаналітик здійснив загрозу повне розкриття.

4.9 Атака на алгоритм підпису ЕльГамаля

Атака була запропонована на конференції Eurocrypt'96 швейцарським криптографом Даниелем Блайхенбахером (Daniel Bleichenbacher, 1964 р.н.).

Атакуючий обирає спеціальним образом загальносистемні параметри – просте число p і твірний елемент g мультиплікативної групи залишків за модулем p – та надає їх підписанту А для реалізації алгоритму ЕльГамаля. Це в

подальшему дозволіть атакуючому легко розв'язати задачу дискретного логарифмування для формування підпису по відкритому ключу y абонента А без знання його секретного ключа.

Атака на алгоритм підпису ЕльГамаля

Атакуючий обирає загальносистемний параметр p за умовою

$$p-1=q\cdot u,$$

де q – велике просте число, а число u має невеликі прості дільники.

Далі атакуючий обирає числа $\alpha < p$, $c < u$ і $\beta = c \cdot q$ таким чином, щоб елемент g, обчислений за формулою

$$g=\beta^{\alpha} \bmod p,$$

був генератором мультиплікативної групи залишків за модулем p.

Загальносистемні параметри – p і g – атакуючий надає підписанту А.

Зауважимо, що елемент $g^q \bmod p$ мультиплікативної групи залишків за модулем p має порядок, менший або рівний u:

$$\left(g^q\right)^u = g^{q\cdot u} = g^{p-1} = 1 \bmod p.$$

Тоді легко розв'язується задача дискретного логарифмування відносно z

$$\left(g^q\right)^z = y^q \bmod p,$$

де y – відомий атакуючому відкритий ключ абонента А.

За допомогою параметра z атакуючий може сформувати справжній підпис будь-якого повідомлення M без знання секретного ключа абонента А:

$$r=\beta,$$

$$s=\alpha(M-\beta\cdot z)\bmod(p-1).$$

Покажемо, що сформований підпис пройде перевірку підпису за алгоритмом ЕльГамаля:

$$y^r \cdot r^s = y^{c \cdot q} \cdot \beta^{\alpha \cdot (M - c \cdot q \cdot z)} = g^{c \cdot q \cdot z} \cdot g^{M - c \cdot q \cdot z} = g^M \bmod p .$$

Таким чином, атакуючий має можливість підробляти підпис абонента А для будь-якого документа за допомогою його відкритого ключа.

Приклад.

Нехай атакуючий обрав загальносистемний параметр $p = 53$.

Тоді $p - 1 = 52 = 13 \cdot 4$ і $q = 13$, $u = 4$.

Нехай атакуючий обрав $\alpha = 7$, $c = 3$ та обчислив $\beta = c \cdot q = 3 \cdot 13 = 39$ і $g = \beta^{\alpha} \bmod p = 39^7 \bmod 53 = 51$.

Перевіримо, що $g = 51$ є генератором мультиплікативної групи залишків за модулем $p = 53$.

Число $p - 1 = 52$ має прості дільники 13 і 2.

Оскільки

$$51^{\frac{52}{13}} = 51^4 \bmod 53 = 16 \neq 1,$$

$$51^{\frac{52}{2}} = 51^{26} \bmod 53 = 52 \neq 1,$$

то $g = 51$ є генератором мультиплікативної групи залишків за модулем $p = 53$

Загальносистемні параметри – $p = 53$ і $g = 51$ – атакуючий надає підписанту А.

Нехай атакуючому відомий відкритий ключ $y = 37$ абонента А.

Зауважимо, що елемент $g^q \bmod p = 51^{13} \bmod 53 = 23$ мультиплікативної групи залишків за модулем $p = 53$ має порядок, менший або рівний $u = 4$.

Тоді легко розв'язується задача дискретного логарифмування відносно z

$$23^z = 52 \bmod 53 ,$$

де $y^q \bmod p = 37^{13} \bmod 53 = 52$.

Обчислимо $23^2 \bmod 53 = 52$, $23^3 \bmod 53 = 30$, $23^4 \bmod 53 = 1$.

Звідси $z = 2$.

За допомогою параметра $z = 2$ атакуючий може сформувати справжній підпис будь-якого повідомлення M, наприклад, $M = 40$, без знання секретного ключа абонента А:

$r = \beta = 39$,

$s = \alpha(M - \beta \cdot z) \bmod(p-1) = 7(40 - 39 \cdot 2) \bmod 52 = 46$.

$\langle r = 39, s = 46 \rangle$.

Покажемо, що сформований підпис $\langle r = 39, s = 46 \rangle$ пройде перевірку підпису за алгоритмом ЕльГамаля:

$y^r \cdot r^s \bmod p = 37^{39} \cdot 39^{46} \bmod 53 = 46$,

$g^M \bmod p = 51^{40} \bmod 53 = 46$.

Таким чином, атакуючий має можливість підробляти підпис абонента А для будь-якого документа M за допомогою його відкритого ключа.

Контрольні питання

1. Що є геш-функція?
2. Опишить «Парадокс днів народження».
3. Опишить атаку на геш-функцію на базі парадокса.
4. Як розрахувати мінімальне значення модифікованих документів для гешування?
5. Яки схеми цифрового підпису називаються слабкими?
6. Опишіть атаки на цифровий підпис.
7. Що називається екзистенційною підробкою цифрового підпису?
8. Що називається селективною підробкою цифрового підпису?
9. Чім відрізняється універсальна підробка від підробки «повне розкриття»?

10. Які параметри є загальносистемними в алгоритмі цифрового підпису ЕльГамаля?
11. Опишіть атаку на алгоритм цифрового підпису ЕльГамаля.

5 АСИМЕТРИЧНА КРИПТОГРАФІЯ НА ЕЛІПТИЧНИХ КРИВИХ

В розділі розглядаються сучасні стандарти протоколів обміну ключами (Діффі-Хеллмана, Ескер), шифрування (алгоритм Смарта), стандартів цифрового підпису над простим (ECDSA, EC-GDSA) і розширеними полями (EC-KCDSA, ДСТУ 4145-2002).

5.1 Елементи теорії полей

Полем $(F,+,\times)$ називається множина F, на якій визначені дві бінарні операції додавання «$+$» і множення «$\times$», що задовольняють умовам:

5) $(F,+)$ є комутативної адитивною групою з нейтральним елементом нуль O;
6) $(F\setminus\{\mathrm{O}\},\times)$ є комутативної мультиплікативною групою з нейтральним елементом одиниця I.
7) для будь-яких елементів a, b, c із множини F виконується дистрибутивний закон $(a+b)\times c = a\times c + b\times c$.

Найпростішими прикладами числових полів є поле раціональних чисел Q і поле дійсних чисел R щодо операцій додавання і множення чисел. Можна довести, що при будь-якому простому p (і лише в цьому випадку) множина Z_p є полем. Воно називається полем залишків за модулем p.

Нехай задано поле $(E,+,\times)$. Підмножина $F \subseteq E$ з операціями поля E називається *підполем* поля E, якщо воно є полем щодо цих операцій. Поле E в цьому випадку називається *розширенням поля* F.

Наприклад, поле дійсних чисел R є розширенням поля раціональних чисел Q.

Скінчені поля – поля Галуа[7]

Поле, що містить скінчене число елементів, називається *скінченим*, або полем Галуа.

Порядок скінченого поля – це число його елементів.

Якщо $(F,+,\times)$ – скінчене поле, то воно містить $q = p^n$ елементів, де p – просте число, $n \geq 1$.

Позначення скінчених полів – $GF(q)$.

Якщо $n = 1$, поле $GF(p)$ називається *простим*.

Якщо $n > 1$, поле $GF(p^n)$ називається *розширеним*.

Ненульові елементи поля Галуа $GF(q)$ утворюють мультиплікативну групу, яка позначається через $GF^*(q)$.

Мультиплікативна група поля $GF^*(q)$ є циклічною групою порядку $q-1$.

Твірний елемент мультиплікативної групи поля Галуа називають *примітивним елементом* поля.

Прикладом простого поля є множина Z_p з операціями додавання і множення цілих чисел за модулем p.

У полі Z_7 примітивним елементом є, наприклад, $a = 3$.

Його степені $3^1 \bmod 7 = 3$, $3^2 \bmod 7 = 2$, $3^3 \bmod 7 = 6$, $3^4 \bmod 7 = 4$, $3^5 \bmod 7 = 5$, $3^6 \bmod 7 = 1$ вичерпують всі ненульові елементи поля.

Елементи розширеного поля, наприклад, $GF(2^3)$ можуть бути представлені у вигляді множини многочленів: 0, 1, t, $t+1$, t^2, t^2+1, t^2+t, t^2+t+1 або набору бінарних векторів {000}, {00}, {010}, {011}, {100}, {101}, {110}, {111}. При цьому опера-

---

[7] Эварист Галуа (1811 – 1832) – французский математик Galois Evariste.

ція додавання здійснюється за модулем 2, а операція множення за модулем обраного незвідного многочлена $f(t)$ третього ступеня, наприклад, $f(t) = t^3 + t + 1$.

5.2 Розв'язання квадратного рівняння в простому полі

Розглянемо квадратне рівняння в вигляді

$y^2 = c \bmod m$

відносно y, де m – просте число, $0 < c < m - 1$.

Число c називається *квадратичним залишком* за модулем m, якщо існує число a, таке що $a^2 = c \bmod m$.

Число c є квадратичним залишком за модулем m, якщо виконується умова

$$c^{\frac{m-1}{2}} = 1 \bmod m.$$

Число c називається *квадратичним незалишком* за модулем m, якщо не існує числа a, такого що $a^2 = c \bmod m$.

Число c є *квадратичним незалишком* за модулем m, якщо виконується умова

$$c^{\frac{m-1}{2}} = -1 \bmod m.$$

Нехай $m = 3 \bmod 4$.

Якщо $c^{\frac{m-1}{2}} = 1 \bmod m$, розв'язком рівняння $y^2 = c \bmod m$ є $y = c^{\frac{m+1}{4}} \bmod m$; якщо $c^{\frac{m-1}{2}} = -1 \bmod m$, то рівняння $y^2 = c \bmod m$ розв'язків не має.

Нехай $m = 5 \bmod 8$.

Якщо $c^{\frac{m-1}{2}} = 1 \bmod m$, то

якщо $c^{\frac{m-1}{4}} = 1 \bmod m$, розв'язком рівняння $y^2 = c \bmod m$ є $y = c^{\frac{m+3}{8}} \bmod m$;

якщо $c^{\frac{m-1}{4}} = -1 \bmod m$, розв'язком рівняння $y^2 = c \bmod m$ є

$$y = c^{\frac{m+3}{8}} \cdot d^{\frac{m-1}{4}} \bmod m,$$

де число d є квадратичним незалишком за модулем m.

Якщо $c^{\frac{m-1}{2}} = -1 \bmod m$, то рівняння $y^2 = c \bmod m$ розв'язків не має.

Схема розв'язання рівняння $y^2 = c \bmod m$ відображена в таблиці 5.1

Таблиця 5.1 – Розв'язання рівняння $y^2 = c \bmod m$

<table>
<tr><td colspan="4">$y^2 = c \bmod m$</td></tr>
<tr><td colspan="3">$c^{\frac{m-1}{2}} = 1 \bmod m$</td><td>$c^{\frac{m-1}{2}} = -1 \bmod m$</td></tr>
<tr><td>$m = 3 \bmod 4$</td><td colspan="2">$m = 5 \bmod 8$</td><td rowspan="3">розв'язків не має</td></tr>
<tr><td rowspan="2">$y = c^{\frac{m+1}{4}} \bmod m$</td><td>$c^{\frac{m-1}{4}} = 1 \bmod m$</td><td>$c^{\frac{m-1}{4}} = -1 \bmod m$</td></tr>
<tr><td>$y = c^{\frac{m+3}{8}} \bmod m$</td><td>$y = c^{\frac{m+3}{8}} \cdot d^{\frac{m-1}{4}} \bmod m$
$d^{\frac{m-1}{2}} = -1 \bmod m$</td></tr>
</table>

Приклад 1.

Розв'язати квадратне рівняння $y^2 = 37 \bmod 71$.

Визначимо, чи є число $c = 37$ квадратичним залишком за модулем 71: $c^{\frac{m-1}{2}} \bmod m = 37^{35} \bmod 71 = 1$.

Оскільки $c = 37$ є квадратичним залишком за модулем 71 і $71 = 3 \bmod 4$, то розв'язком рівняння $y^2 = 37 \bmod 71$ є $y = 37^{18} \bmod 71 = 45$, $y = 45$.

Перевірка: $45^2 \bmod 71 = 2025 \bmod 71 = 37$.

Приклад 2.

Розв'язати квадратне рівняння $y^2 = 28 \bmod 71$.

Визначимо, чи є число $c = 28$ квадратичним залишком за модулем 71: $c^{\frac{m-1}{2}} \bmod m = 28^{35} \bmod 71 = -1$.

Оскільки $c = 28$ є квадратичним незалишком за модулем 71, то рівняння $y^2 = 28 \bmod 71$ розв'язків не має.

Приклад 3.

Розв'язати квадратне рівняння $y^2 = 10 \bmod 53$.

Визначимо, чи є число $c = 10$ квадратичним залишком за модулем 53: $c^{\frac{m-1}{2}} \bmod m = 10^{26} \bmod 53 = 1$.

Число $c = 10$ є квадратичним залишком за модулем 53. Оскільки $53 = 5 \bmod 8$, обчислимо значення $c^{\frac{m-1}{4}} \bmod m$: $10^{13} = 1 \bmod 53$.

Таким чином, розв'язком рівняння є $y = 10^7 \bmod 53 = 13$.

$y = 13$.

Перевірка: $13^2 \bmod 53 = 169 \bmod 53 = 10$.

Приклад 4.

Розв'язати квадратне рівняння $y^2 = 2 \bmod 53$.

Визначимо, чи є число $c = 2$ квадратичним залишком за модулем 53: $c^{\frac{m-1}{2}} \bmod m = 2^{26} \bmod 53 = -1$.

Оскільки $c = 2$ є квадратичним незалишком за модулем 53, то рівняння $y^2 = 2 \bmod 53$ розв'язків не має.

Приклад 5.

Розв'язати квадратне рівняння $y^2 = 7 \bmod 53$.

Визначимо, чи є число $c = 7$ квадратичним залишком за модулем 53: $c^{\frac{m-1}{2}} \bmod m = 7^{26} \bmod 53 = 1$.

Число $c = 7$ є квадратичним залишком за модулем 53.

Оскільки $53 = 5 \bmod 8$, обчислимо значення $c^{\frac{m-1}{4}} \bmod m$: $7^{13} = -1 \bmod 53$.

Знайдемо число d – квадратичний незалишок за модулем 53.

Згідно з прикладом 4, число $d = 2$ є квадратичним незалишком за модулем 53.

Звідси розв'язком рівняння $y^2 = 7 \bmod 53$ є

$y = c^{\frac{m+3}{8}} \cdot d^{\frac{m-1}{4}} \bmod m = 7^7 \cdot 2^{13} \bmod 53 = 22$.

$y = 22$.

Перевірка: $22^2 \bmod 53 = 484 \bmod 53 = 7$.

5.3 Еліптичні криві над простим полем

Еліптична крива над простим полем $GF(p)$ задається рівнянням

$$y^2 = x^3 + ax + b \bmod p,$$

де p – просте число, a, b – задані коефіцієнти кривої, при цьому $4a^3 + 27b^2 \neq 0 \bmod p$, $a, b, x, y \in GF(p)$.

Пара елементів (x, y) поля $GF(p)$, що задовольняє рівнянню $y^2 = x^3 + ax + b \bmod p$, є точкою еліптичної кривої. У множину точок кривої також включається нескінченно віддалена точка O.

Порядком кривої називається число її точок (з нескінченно віддаленою).

Теорема Хасе. Значення порядку n еліптичної кривої над полем Галуа $GF(p)$ має верхню і нижню межі:

$$p + 1 - 2\sqrt{p} \le n \le p + 1 + 2\sqrt{p}$$

Нехай точка $P(x, y)$ належить еліптичній кривій.

Точка $-P(x,-y)$ називається *оберненою* (або *протилежною*) точки $P(x, y)$.

На множині точок еліптичної кривої введемо *операцію додавання* «+».

Нехай точки $P_1(x_1, y_1)$ і $P_2(x_2, y_2)$ належать еліптичній кривій.

Визначимо суму точок $P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2)$.

Якщо $P_1 = \mathrm{O}$, то $P_3 = P_2$.

Якщо $P_2 = \mathrm{O}$, то $P_3 = P_1$.

Якщо $x_1 \ne x_2$, то

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p,\ y_3 = \lambda(x_1 - x_3) - y_1 \bmod p,$$

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2} \bmod p.$$

Якщо $x_1 = x_2$, то

якщо $y_1 = 0$ або $y_1 = -y_2$, то $P_3 = \mathrm{O}$;

інакше $x_3 = \lambda^2 - x_1 - x_2 \bmod p$, $y_3 = \lambda(x_1 - x_3) - y_1 \bmod p$,

$$\lambda = \frac{3x_1^2 + a}{2y_1} \bmod p$$

При певних умовах усі точки еліптичної кривої утворюють циклічну групу (існує генератор групи).

Теорема Кассельса. Якщо дільники порядку кривої не мають квадратів, то множина точок еліптичної кривої з операцією додавання «+» утворює циклічну групу з нейтральним елементом O.

При цьому виконуються умови групи:

1) для будь-яких точок кривої сума точок також належить кривій;
2) для будь-яких точок кривої P_1, P_2, P_3 виконується рівність $P_1 + (P_2 + P_3) = (P_1 + P_2) + P_3$;
3) у множині точок кривої існує нейтральний елемент O: $P + \mathrm{O} = \mathrm{O} + P = P$;
4) для кожної точки кривої P існує протилежна точка $-P$ така, що $P + (-P) = \mathrm{O}$.

В криптографії використовують еліптичні криви, які мають циклічні групи точок кривої або їхні підгрупи великого простого порядку.

Генератор групи точок кривої або її підгрупи прийнято називати *базовою точкою*.

Порядком точки P еліптичної кривої називається найменше число k, таке що $k \cdot P = \mathrm{O}$.

Порядок точки є дільником порядку еліптичної кривої.

Приклад 1.

Дано еліптична крива над простим полем $GF(7)$

$y^2 = x^3 + x + 6 \bmod 7$.

Точка $P = (1,1)$ належить еліптичній кривій.

Знайдемо $2P$, $3P$…

$2P = (1,1) + (1,1)$.

$$\lambda = \frac{3 \cdot 1 + 1}{2} \bmod 7 = \frac{4}{2} \bmod 7 = 2$$

$$x_3 = 2^2 - 1 - 1 \bmod 7 = 2$$

$$y_3 = 2(1 - 2) - 1 \bmod 7 = 4$$

Таким чином, $2P = (2,4)$.

$$3P = P + 2P = (1,1) + (2,4)$$

$$\lambda = \frac{1 - 4}{1 - 2} \bmod 7 = \frac{-3}{-1} \bmod 7 = 3$$

$$x_3 = 3^2 - 1 - 2 \bmod 7 = 6$$

$$y_3 = 3(1 - 6) - 1 \bmod 7 = 5$$

Таким чином, $3P = (6,5)$.

$$4P = P + 3P = (1,1) + (6,5)$$

$$\lambda = \frac{1 - 5}{1 - 6} \bmod 7 = \frac{-4}{-5} \bmod 7 = \frac{-4}{2} \bmod 7 = -2 \bmod 7 = 5$$

$$x_3 = 5^2 - 1 - 6 \bmod 7 = 4$$

$$y_3 = 5(1 - 4) - 1 \bmod 7 = 5$$

Таким чином, $4P = (4,5)$.

$P = (1,1)$, $2P = (2,4)$, $3P = (6,5)$, $4P = (4,5)$, $5P = (3,1)$, $6P = (3,6)$, $7P = (4,2)$, $8P = (6,2)$, $9P = (2,3)$, $10P = (1,6)$, $11P = \mathrm{O}$.

Пари протилежних точок:

P і $10P$, $2P$ і $11P$, $3P$ і $8P$, $4P$ і $7P$, $5P$ і $6P$.

Порядок еліптичної кривої $y^2 = x^3 + x + 6 \bmod 7$ дорівнює 11.

Оскільки порядок кривої є простим числом $n = 11$, то будь-яка точка кривої $y^2 = x^3 + x + 6 \bmod 7$, окрім O, може бути базовою.

Алгоритм обчислення точки еліптичної кривої

Розглянемо еліптичну криву над простим полем $GF(p)$

$$y^2 = x^3 + ax + b \bmod p .$$

Оберемо довільний елемент x_0 простого поля $GF(p)$ та обчислимо значення $c = x_0^3 + ax_0 + b \bmod p$.

Отримуємо рівняння $y^2 = c \bmod p$.

Якщо c є квадратичним залишком за модулем p, тобто для числа c виконується умова $c^{\frac{p-1}{2}} = 1 \bmod p$, то існує розв'язок квадратного рівняння $y^2 = c \bmod p$ (див. 5.2). Цей розв'язок y_0 і є другою координатою точки еліптичної кривої: $Q=(x_0,y_0)$.

Якщо c не є квадратичним залишком за модулем p, то необхідно повторювати процедуру пошуку для іншого елемента x_0.

Приклад 2.

Розглянемо еліптичну криву $y^2 = x^3 + 2x + 4 \bmod 7$ над простим полем $GF(7)$. Знайдемо точку кривої.

Оберемо $x_0 = 3$ та обчислимо

$$c = x_0^3 + 2x_0 + 4 \bmod p = (27 + 6 + 4) \bmod 7 = 2 .$$

Визначимо, чи є число $c = 2$ квадратичним залишком за модулем 7: $c^{\frac{p-1}{2}} \bmod p = 2^3 \bmod 7 = 1$.

Оскільки $c = 2$ є квадратичним залишком за модулем 7 і $7 = 3 \bmod 4$, то розв'язком рівняння $y^2 = 2 \bmod 7$ є $y_0 = c^{\frac{p+1}{4}} \bmod p = 2^2 \bmod 7 = 4$:

$x_0 = 3$, $y_0 = 4$.

Знайдено точку $Q = (3,4)$ еліптичної кривої

$y^2 = x^3 + 2x + 4 \bmod 7$.

Алгоритм послідовного подвоєння точок еліптичної кривої

Розглянемо еліптичну криву над простим полем $GF(p)$

$$y^2 = x^3 + ax + b \bmod p$$

і точку P, що належить еліптичній кривій.

Нехай необхідно обчислити точку $k \cdot P$, k – натуральне число. Для цього використаємо алгоритм послідовного подвоєння точок еліптичної кривої.

7. Розкладемо k по ступенях двійки:

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

8. $V := \mathrm{O}$.
9. $A := P$.
10. Якщо $k_1 = 1$, тоді $V := P$.
11. Цикл по i от 2 до t
    a. $A := A + A$
    b. Якщо $k_i = 1$, тоді $V := V + A$
12. Повертаємо V.

Приклад 3.

Розглянемо еліптичну криву над простим полем $GF(7)$

$y^2 = x^3 + x + 6 \bmod 7$

і точку кривої $P = (1,1)$.

Знайдемо точку кривої $V = 5 \cdot P$.

Розкладемо $k = 5$ по ступенях двійки: $k = 1 + 0 \cdot 2 + 1 \cdot 2^2$.

$V := \mathrm{O}$.

$A := (1,1)$.

$k_1 = 1$, $V := (1,1)$.

$k_2 = 0$, $A := (1,1) + (1,1) = (2,4)$.

$k_3 = 1$, $A := (2,4) + (2,4) = (4,5)$, $V := (1,1) + (4,5) = (3,1)$.

Таким чином, $V = 5 \cdot P = (3,1)$.

5.4 Дискретний логарифм в групі точок еліптичної кривої

Задача дискретного логарифмування в групі точок еліптичної кривої: при заданої кривої $y^2 = x^3 + ax + b \bmod p$, базової точки P відомого порядку n, заданої точки Q знайти d, що задовольняє рівнянню

$$d \cdot P = Q.$$

(Адитивний аналог задачі дискретного логарифмування в мультиплікативної групі).

Застосуємо модіфікований алгоритм великих-малих кроків Шенкса..

1. Обчислюємо $m = \left\lceil \sqrt{n} \right\rceil$, де $\lceil u \rceil$ – округлення u з надлишком.

2. Будуємо таблицю пар

j	1	2	…	$m-1$	m
$j \cdot P$	P	$2 \cdot P$	…	$(m-1) \cdot P$	$m \cdot P$

3. Обчислюємо $T = -m \cdot P$.

4. Положимо $S := Q$.

5. Цикл по i від 0 до $m-1$

5.1 перевіряємо, чи є існує j таке, що $j \cdot P = S$;

5.2 якщо існує j: $j \cdot P = S$, то розв'язком є $d = m \cdot i + j$;

5.3 інакше $S := S + T$.

Алгоритм великих-малих кроків Шенкса може бути застосований і для обчислення порядку точки P кривої, тобто для розв'язанні задачі

$n \cdot P = \mathrm{O}$

відносно n.

Приклад 1.

Нехай задано еліптична крива

$y^2 = x^3 + 15x + 24 \bmod 43$,

базова точка кривої $P = (0,14)$, порядок базової точки $n = 41$, точка кривої $Q = (23,18)$.

Знайдемо розв'язок задачі дискретного логарифмування

$d \cdot (0,14) = (23,18)$

в групі точок еліптичної кривої за допомогою алгоритму Шенкса.

Значення $m = \left| \sqrt{41} \right| = 7$.

Побудуємо таблицю

j	1	2	3	4	5	6	7
$j \cdot P$	(0,14)	(1,30)	(40,34)	(14,22)	(3,15)	(21,22)	(20,5)

Обчислимо $T = -(20,5) = (20,38)$.

Положимо $S := (23,18)$.

$i = 0$, $S := (23,18) + (20,38) = (11,31)$;

$i = 1$, $S := (11,31) + (20,38) = (28,37)$;

$i = 2$, $S := (28,37) + (20,38) = (36,7)$;

$i = 3$, $S := (36,7) + (20,38) = (1,30)$;

$i = 4$, $j = 2$, $x = 7 \cdot 4 + 2 = 30$.

Розв'язком задачі $d \cdot (0,14) = (23,18)$ є $d = 30$.

?

Перевірка. $30 \cdot (0,14) = (23,18)$

Розкладемо число 30 по ступенях двійки:

$30 = 0 + 1 \cdot 2 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4$.

$V := \mathrm{O}$, $A := (0,14)$.

$k_1 = 0$;

$k_2 = 1$,

$A := (0,14) + (0,14) = (1,30)$, $V := \mathrm{O} + (1,30) = (1,30)$;

$k_3 = 1$,

$A := (1,30) + (1,30) = (14,22)$,

$V := (1,30) + (14,22) = (21,22)$;

$k_4 = 1$,

$A := (14,22) + (14,22) = (16,19)$,

$V := (21,22) + (16,19) = (27,26)$;

$k_5 = 1$,

$A := (16,19) + (16,19) = (28,37)$,

$V := (27,26) + (28,37) = (23,18)$.

Таким чином, $30 \cdot (0,14) = (23,18)$.

Розв'язок задачі знайдено вірно.

Приклад 2.

Нехай задано еліптична крива $y^2 = x^3 + 15x + 24 \bmod 43$ і точка кривої $P = (0,14)$.

Знайдемо порядок точки P: число n, таке що $n \cdot P = \mathrm{O}$.

Оцінку порядку точки можна зробити за допомогою теореми Хасе при $q = 43$:

$$43 + 1 - 2\sqrt{43} \le n \le 43 + 1 + 2\sqrt{43}.$$

Максимальна оцінка порядку точки кривої дорівнює 57.

Використуємо алгоритм Шенкса для розвязання задачи $n \cdot P = \mathrm{O}$

Значення $m = \left\lfloor \sqrt{57} \right\rfloor = 8$.

Побудуємо таблицю

j	1	2	3	4	5	6	7	8
$j \cdot P$	0,14	1,30	40,34	14,22	3,15	21,22	20,5	16,19

Обчислимо $T = -(16,19) = (16,24)$.

Положимо $S := \mathrm{O}$.

$i = 0$, $S := \mathrm{O} + (16,24) = (16,24)$;

$i = 1$, $S := (16,24) + (16,24) = (28,6)$;

$i = 2$, $S := (28,6) + (16,24) = (12,13)$;

$i = 3$, $S := (12,13) + (16,24) = (36,7)$;

$i = 4$, $S := (36,7) + (16,24) = (0,14)$;

$i = 5$, $j = 1$, $n = 8 \cdot 5 + 1 = 41$.

$n = 41$.

Таким чином, порядок n точки $P = (0,14)$ еліптичної кривої $y^2 = x^3 + 15x + 24 \bmod 43$ дорівнює 41.

5.5 Схема Діффі-Хеллмана на еліптичних кривих

Схема Діффі-Хеллмана призначена для одержання спільного секретного ключа по відкритому каналу зв'язку між двома абонентами мережі.

Два абонента А і В обирають спільні відкриті параметри: еліптичну криву над простим полем і базову точку P еліптичної кривої, яка має великий порядок n.

Абоненти А і В обмінюються спільними відкритими параметрами по відкритому каналу мережі.

Далі абонент А обирає випадкове таємне число c, $1 < c < n$ та обчислює точку Q за формулою $Q = c \cdot P$.

Абонент В обирає випадкове таємне число d, $1 < d < n$ та обчислює точку R за формулою $R = d \cdot P$.

Абоненти А і В обмінюються значеннями Q та R по відкритому каналу мережі.

Після отримання значення R абонент А обчислює значення S за формулою $S = c \cdot R$, а абонент В після отримання значення Q обчислює значення T за формулою $T = d \cdot Q$.

Знайдені абонентами А і В значення S та T співпадають:

$$S = c \cdot R = c \cdot (d \cdot P) = d \cdot (c \cdot P) = d \cdot Q = T$$

і утворюють спільний секретний ключ.

Приклад 1.

Нехай відкритими параметрами схеми є еліптична крива $y^2 = x^3 + 15x + 24 \bmod 43$, базова точка кривої $P = (0,14)$, порядок базової точки $n = 41$.

Абонент А обирає випадкове таємне число $c = 12$ та обчислює точку $Q = 12 \cdot P = (26,39)$.

Абонент В обирає випадкове таємне число $d = 5$ та обчислює точку $R = 5 \cdot P = (3,15)$.

Абоненти А і В обмінюються значеннями Q та R по відкритому каналу мережі.

Після отримання значення $R = (3,15)$ абонент А обчислює значення $S = 12 \cdot R = (10,23)$, а абонент В після отримання значення $Q = (26,39)$ обчислює значення $T = 5 \cdot Q = (10,23)$.

Знайдені абонентами А і В значення $S = (10,23)$ та $T = (10,23)$ співпадають і створюють спільний секретний ключ.

Приклад 2.

Нехай відкритими параметрами схеми є еліптична крива $y^2 = x^3 + 15x + 24 \bmod 43$, базова точка кривої $P = (0,14)$, порядок базової точки $n = 41$.

Криптоаналітик супротивника перехопив значення точок $Q = (23,18)$, $R = (14,22)$ – обміну за схемою Діффі-Хеллмана та бажає обчислить спільний секретний ключ абонентів А і В.

Для отримання таємного числа c абонента А криптоаналітику необхідно розв'язати задачу дискретного логарифмування в групі точок еліптичної кривої $c \cdot P = Q$:

$$c \cdot (0,14) = (23,18)$$

відносно c.

За допомогою алгоритму Шенкса (див. Приклад 1 Додатку Г) знайдемо

$$c = 30.$$

Для отримання спільного секретного ключа абонентів А і В достатньо обчислити $S = c \cdot R$:

$$S = 30 \cdot R = (40,9).$$

Таким чином, спільний секретний ключ $S = (40,9)$ абонентів А і В знайдено.

5.6 Протокол ECKEP

Протокол обчислення ключа парного зв'язку ECKEP (Elliptic Curve Key Establishment Protocol) вважається більш захищеним від зламу, ніж протокол Діффі-Хеллмана. Для зламу

протоколу Діффі-Хеллмана достатньо розв'язати одну задачу дискретного логарифмування в групі точок еліптичної кривої.

Протокол обчислення ключа парного зв'язку ЕСКЕР реалізовано в групі точок еліптичної кривої великого порядку.

Два абонента А і В обирають спільні відкриті параметри: еліптичну криву над простим полем і базову точку P еліптичної кривої, яка має великий порядок n.

Абоненти А і В обмінюються спільними відкритими параметрами по відкритому каналу мережі.

Кожний абонент А і В має асиметричну пару ключів: секретні u, v та відкриті $U = u \cdot P$, $V = v \cdot P$, $1 < u, v < n$, відповідно.

На першому етапі абоненти обмінюються відкритими ключами.

Далі абонент А обирає випадкове таємне число l, $1 < l < n$ та обчислює точку Q за формулою $Q = l \cdot P$.

Абонент В обирає випадкове таємне число m, $1 < m < n$ та обчислює точку R за формулою $R = m \cdot P$.

Абоненти А і В обмінюються значеннями Q та R по відкритому каналу мережі.

Після отримання значення R абонент А обчислює допоміжне значення c за формулою

$$c = l + u \cdot (U)_x (l \cdot V)_x \bmod n,$$

а абонент В після отримання значення Q обчислює допоміжне значення d за формулою

$$d = m + v \cdot (V)_x (m \cdot U)_x \bmod n.$$

Далі абоненти А і В за допомогою чисел c і d обчислюють секретний ключ кожен на своєї стороні відповідно:

$$\mathcal{K}_A = c \cdot \left(R + (V)_x \cdot (u \cdot R)_x \cdot V \right),$$

$$\mathcal{K}_B = d \cdot \left(Q + (U)_x \cdot (v \cdot Q)_x \cdot U \right).$$

Знайдені абонентами А і В значення $\mathcal{K}_A$ та $\mathcal{K}_B$ співпадають і утворюють спільний секретний ключ.

Покажемо, що $\mathcal{K}_A = \mathcal{K}_B$:

$$\begin{aligned}
&\mathcal{K}_A = c \cdot \left(R + (V)_x \cdot (u \cdot R)_x \cdot V \right) = \\
&= \left(l + u \cdot (U)_x (l \cdot V)_x \right) \cdot \left(R + (V)_x \cdot (u \cdot R)_x \cdot V \right) = \\
&= l \cdot R + u \cdot (U)_x (l \cdot V)_x \cdot R + l \cdot (V)_x \cdot (u \cdot R)_x \cdot V + \\
&+ u \cdot (U)_x (l \cdot V)_x \cdot (V)_x \cdot (u \cdot R)_x \cdot V = \\
&= \left(m \cdot Q + m \cdot (U)_x (v \cdot Q)_x \cdot U \right) + v \cdot (V)_x \cdot (m \cdot U)_x \cdot Q + \\
&+ v \cdot (V)_x \cdot (m \cdot U)_x \cdot (U)_x \cdot (v \cdot Q)_x \cdot U = \\
&= m \cdot \left(Q + (U)_x \cdot (v \cdot Q)_x \cdot U \right) + \\
&+ v \cdot (V)_x (m \cdot U)_x \cdot \left(Q + (U)_x \cdot (v \cdot Q)_x \cdot U \right) = \\
&= \left(m + v \cdot (V)_x (m \cdot U)_x \right) \cdot \left(Q + (U)_x \cdot (v \cdot Q)_x \cdot U \right) = \\
&= d \cdot \left(Q + (U)_x \cdot (v \cdot Q)_x \cdot U \right) = \mathcal{K}_B
\end{aligned}$$

Оскільки при формуванні спільного ключа в протоколі ЕСКЕР використовуються допоміжні параметри, які не передаються по мережах зв'язку, то зловмисник не має можливості їх обчисліть без знання асиметричних секретних ключів абонентів. До того ж йому необхідно також розв'язати задачу дискретного логарифмування в групі точок еліптичної кривої.

Приклад.

Нехай відкритими параметрами схеми є еліптична крива $y^2 = x^3 + 15x + 24 \bmod 43$, базова точка кривої $P = (0,14)$, порядок базової точки $n = 41$.

Кожний абонент А і В має асиметричну пару ключів: секретні $u = 27$, $v = 38$ та відкриті $U = u \cdot P = 27 \cdot P = (27,17)$, $V = v \cdot P = 38 \cdot P = (40,9)$ відповідно.

Абонент А обирає випадкове таємне число $l = 12$ та обчислює точку $Q = 12 \cdot P = (26,39)$.

Абонент В обирає випадкове таємне число $m = 5$ та обчислює точку $R = 5 \cdot P = (3,15)$.

Абоненти А і В обмінюються значеннями Q та R по відкритому каналу мережі.

Після отримання значення $R = (3,15)$ абонент А обчислює допоміжне значення $c = l + u \cdot (U)_x (l \cdot V)_x \bmod n = 12 + 27 \cdot 27 \cdot 3 \bmod 41 = 26$, а абонент В після отримання значення $Q = (26,39)$ обчислює допоміжне значення $d = m + v \cdot (V)_x (m \cdot U)_x \bmod n = 5 + 38 \cdot 26 \bmod 41 = 1$.

Далі абоненти А і В за допомогою чисел c і d обчислюють секретний ключ кожен на своєї стороні відповідно:

$$\mathcal{K}_A = c \cdot \left(R + (V)_x \cdot (u \cdot R)_x \cdot V \right) = 26 \cdot \left((3,15) + 40 \cdot 26 \cdot (40,9) \right) = (30,13)$$

,

$$\mathcal{K}_B = d \cdot \left(Q + (U)_x \cdot (v \cdot Q)_x \cdot U \right) = 1 \cdot \left((26,39) + 27 \cdot 3 \cdot (27,17) \right) = (30,13)$$

.

Знайдені абонентами А і В значення $\mathcal{K}_A$ та $\mathcal{K}_B$ співпадають і утворюють спільний секретний ключ.

5.7 Шифрування на еліптичних кривих

Схему шифрування запропонував Nigel Smart в 1999 році.

Відкритими параметрами є: еліптична крива над полем $GF(q)$, базова точка P, порядок базової точки n.

Нехай абоненту В необхідно передати зашифроване повідомлення абоненту А.

Абонент А генерує пару ключів : секретний d, $2 \le d \le n-2$ та відкритий $Q = d \cdot P$, якій передає абоненту В.

Шифрування (Абонент В) :

Генерується випадкове число k, $2 \le k \le n-2$.

Обчислюються точки $R = k \cdot P$ та $T = k \cdot Q = (x_T, y_T)$

Блок повідомлення M перетворюється на два елемента поля $M = (m_1, m_2)$, $m_i \in GF(q)$, $i = 1,2$.

Зашифроване повідомлення обчислюється за формулами: $c_1 = m_1 + x_T$, $c_2 = m_2 + x_T$.

Абоненту А передається набір $\{R, c_1, c_2\}$.

Розшифрування (Абонент А) :

Обчислюються точка $T^* = d \cdot R = \left(x_{T^*}, y_{T^*}\right)$ і повідомлення: $m_1 = c_1 - x_{T^*}$, $m_2 = c_2 - x_{T^*}$.

Зазначимо, що $T^* = d \cdot R = d \cdot k \cdot P = k \cdot Q = T$.

Приклад.

Відкритими параметрами алгоритму є еліптична крива

$$y^2 = x^3 + 2x + 6 \bmod 17,$$

базова точка кривої $P = (2,1)$, порядок точки P – $n = 11$.

$P = (2,1)$, $2P = (11,4)$, $3P = (6,9)$, $4P = (13,11)$, $5P = (1,3)$, $6P = (1,14)$, $7P = (13,6)$, $8P = (6,8)$, $9P = (11,13)$, $10P = (2,16)$, $11P = \mathrm{O}$.

Секретним ключем розшифрування оберемо число $d = 6$.

Обчислимо відкритий ключ зашифрування: $Q = 6 \cdot P = (1,14)$.

Зашифруємо повідомлення $M = (15,3)$, використовуючи відкритий ключ $Q = (1,14)$.

Оберемо випадкове число $k = 4$ і обчислимо точки

$R = 4 \cdot P = (13,11)$ і $T = 4 \cdot Q = (11,4)$.

Зашифрування повідомлення $M = (15,3)$:

$c_1 = m_1 + x_T = 15 + 11 \bmod 17 = 9$,

$c_2 = m_2 + x_T = 3 + 11 \bmod 17 = 14$.

Таким чином, вихідному повідомленню $(15,3)$ відповідає криптограма $\{(13,11),9,14\}$.

Розшифруємо повідомлення $\{(13,11),9,14\}$, користуючись секретним ключем $d = 6$ та відкритими параметрами алгоритму.

За допомогою секретного ключа d відновлюється точка T:

$T = 6 \cdot R = (11,4)$.

Звідси

$m_1 = c_1 - x_T = 9 - 11 \bmod 17 = 15$

$m_2 = c_2 - x_T = 14 - 11 \bmod 17 = 3$.

В результаті розшифрування було одержано вихідне повідомлення $M = (15,3)$.

5.8 Американський цифрового стандарт підпису ECDSA над простим полем

Алгоритм ECDSA є одним зі національних стандартів електронного цифрового підпису. Цифровий підпис створюється за допомогою операцій над точками еліптичної кривої після хешування повідомлення. В якості хеш-функції використовуються алгоритми класу SHA (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512), наприклад, SHA-1, хеш-образ якого дорівнює 160 біт.

Відкритими параметрами алгоритму є еліптична крива, базова точка кривої P з відомим простим порядком n. Відкритим

ключем підписанта є точка еліптичної кривої Q, $Q = d \cdot P$, його секретним ключем є число d. Для забезпечення криптостійкості цифрового підпису порядок базової точки має бути досить великим: $2^{160} < n < 2^{511}$ або $n > 2^{512}$.

Цифровий підпис формується абонентом А за допомогою його секретного ключа, перевірка підпису здійснюється абонентом В з використанням відкритого ключа абонента А.

Загальносистемні параметри

Еліптична крива над простим полем $GF(p)$

$y^2 = x^3 + ax + b \bmod p$;

P – базова точка кривої з великим простим порядком n.

Генерація ключів

Згенеруємо секретний та відкритий ключі абонента А.

Оберемо випадкове число d, $2 \le d \le n-2$.

Секретним ключем абонента А є число d.

Відкритим ключем абонента А є точка кривої $Q = d \cdot P$.

Формування цифрового підпису

Підписант А обчислює хеш-образ повідомлення M за допомогою хеш-функції SHA. Отримане двійкове число $H(M)$ конвертується в десяткове число H. Потім обчислюється значення $h = H \bmod n$.

Нехай хеш-образу повідомлення M відповідає число h.

Оберемо випадкове число k, $2 \le k \le n-2$.

Обчислимо точку $C = k \cdot P = (x_C, y_C)$ та число $r = x_C \bmod n$. (Число r не повинно бути 0.)

З використанням секретного ключа d та хеш-образу повідомлення h обчислимо значення s із співвідношення $s \cdot k = h + d \cdot r \bmod n$. (Число s не повинно бути 0.)

Цифровим підписом є пара чисел $\langle r, s \rangle$.

Підписане повідомлення має вигляд $\{M,\ \langle r,s\rangle,\ \text{текст}\}$. Поле «текст» є довільним, може містити ідентифікатори підписанта, або помітку часу, наприклад.

Перевірка цифрового підпису

Для перевірки підписаного абонентом А повідомлення $\{M,\ \langle r,s\rangle,\ \text{текст}\}$ використовуються хеш-образ повідомлення M, відкрити загальносистемні параметри алгоритму ECDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка P, її порядок n, десяткове число h, що відповідає хеш-образу повідомлення M, відкритий ключ абонента А – точка кривої Q.

Обчислимо два параметри

$$u = \frac{h}{s} \bmod n \qquad \text{та} \qquad v = \frac{r}{s} \bmod n.$$

Знайдемо точку еліптичної кривої $u \cdot P + v \cdot Q = (x_0, y_0)$.

Параметр $r' = x_0 \bmod n$ повинен співпадати з параметром r.

Якщо $r' = r$, підпис признається справжнім.

Приклад.

Нехай відкритими параметрами алгоритму ECDSA є еліптична крива $y^2 = x^3 + 15x + 24 \bmod 43$, базова точка кривої $P = (0,14)$, порядок точки $n = 41$.

Для генерування асиметричної пари ключів абонента А оберемо випадкове число $d = 5$ та обчислимо точку $Q = 5 \cdot P = (3,15)$.

Відкритим ключем абонента А є точка кривої $Q = (3,15)$.

Секретним ключем абонента А є число $d = 5$.

Для формування підпису абонент А хешує повідомлення M та отримує відповідне десяткове число $h = 4$.

Далі абонент А обирає випадкове число $k = 2$ та обчислює точку $C = 2 \cdot P = (1,30)$. Звідси число $r = 1 \bmod 41 = 1$.

З використанням секретного ключа d та числа h абонент А обчислює параметр s: $s \cdot 2 = 4 + 5 \cdot 1 \bmod 41 = 9 = -32$. Звідси $s = -16 \bmod 41 = 25$.

Цифровим підписом є пара чисел $\langle 1,25 \rangle$.

Підписане повідомлення має вигляд $\{M,\ \langle 1,25 \rangle,\ \text{текст}\}$.

Для перевірки підписаного абонентом А повідомлення $\{M,\ \langle r,s \rangle,\ \text{текст}\}$ використовуються хеш-образ повідомлення M, відкрити загальносистемні параметри алгоритму ECDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка $P = (0,14)$, її порядок $n = 41$, десяткове число $h = 4$, що відповідає хеш-образу повідомлення M, відкритий ключ абонента А – точка кривої $Q = (3,15)$

Обчислимо значення $u = \frac{4}{25} \bmod 41 = \frac{4}{-16} = \frac{-1}{4} = 10$ і $v = \frac{1}{25} \bmod 41 = \frac{-40}{25} = \frac{-8}{5} = \frac{-90}{5} = -18 = 23$.

Знайдемо точку еліптичної кривої $10 \cdot P + 23 \cdot Q = (1,30)$ та параметр $r' = 1 \bmod 41 = 1$.

Оскільки $r' = r$, підпис признається справжнім.

(P=(0,14), 2P=(1,30), 3P=(40,34), 4P=(14,22), 5P=(3,15), 6P=(21,22), 7P=(20,5), 8P=(16,19), 9P=(36,7), 10P=(8,21), 11P=(23,25), 12P=(26,39), 13P=(33,32), 14P=(27,26), 15P=(30,30), 16P=(28,37), 17P=(12,13), 18P=(11,12), 19P=(10,23), 20P=(5,3), 21P=(5,40), 22P=(10,20), 23P=(11,31), 24P=(12,30), 25P=(28,6), 26P=(30,13), 27P=(27,17), 28P=(33,11), 29P=(26,4), 30P=(23,18),

31P=(8,22), 32P=(36,36), 33P=(16,24), 34P=(20,38), 35P=(21,21), 36P=(3,28), 37P=(14,21), 38P=(40,9), 39P=(1,13), 40P=(0,29), 41P=O.)

5.9 Німецький стандарт цифрового підпису EC-GDSA над простим полем

Протоколи підпису ECGDSA над простим та розширеним полем вперше були запропоновані в 1990 році. Після модифікації вони є чинним стандартом цифрового підпису Німеччини.

Цифровий підпис створюється за допомогою операцій над точками еліптичної кривої після хешування повідомлення.

В якості хеш-функцій можуть використовуватися функції RIPEMD-160, SHA-1 (= SHA-160), SHA-224, SHA-256, SHA-384 и SHA-512, хеш-образ яких дорівнює відповідно 160, 224, 256, 384 и 512 бітів.

Відкритими параметрами алгоритму є еліптична крива, базова точка кривої P з відомим простим порядком n. Відкритим ключем підписанта є точка еліптичної кривої Q, $Q = (d^{-1} \bmod n) \cdot P$, його секретним ключем є число d.

Для забезпечення криптостійкості цифрового підпису порядок базової точки має бути досить великим.

Цифровий підпис формується абонентом А за допомогою його секретного ключа, перевірка підпису здійснюється абонентом В з використанням відкритого ключа абонента А.

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(p)$

$$y^2 = x^3 + ax + b \mod p,$$

де $a, b \in GF(p), b \neq 0,$ разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}, 0 < k < n$, $|n|$ – число двійкових розрядів в n; H – обрана функція хешування.

Генерація ключів

Згенеруємо секретний та відкритий ключі абонента А.

Оберемо випадкове число d, $2 \le d \le n-2$.

Обчислимо точку кривої $Q = (d^{-1} \bmod n) \cdot P$.

Секретним ключем абонента А є число d.

Відкритим ключем абонента А є точка кривої Q.

Формування цифрового підпису

Підписант А обчислює хеш-образ повідомлення M за допомогою обраної хеш-функції. Отримане двійкове число $H(M)$ конвертується в десяткове число H. Потім обчислюється значення $h = H \bmod n$.

Нехай хеш-образу повідомлення M відповідає число h.

Оберемо випадкове число k, $2 \le k \le n-2$.

Обчислимо точку $C = k \cdot P = (x_C, y_C)$ та число $r = x_C \bmod n$. (Число r не повинно бути 0.)

З використанням секретного ключа d та хеш-образу повідомлення h обчислимо значення $s = (k \cdot r - h) \cdot d \bmod n$. (Число s не повинно бути 0.)

Цифровим підписом є пара чисел $\langle r, s \rangle$.

Підписане повідомлення має вигляд $\{M, \langle r, s \rangle\}$.

Перевірка цифрового підпису

Для перевірки підписаного абонентом А повідомлення $\{M, \langle r, s \rangle\}$ використовуються хеш-образ повідомлення M, відкрити загальносистемні параметри алгоритму ECGDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка P, її порядок n, десяткове число h, що відповідає хеш-образу повідомлення M, відкритий ключ абонента А – точка кривої Q.

Обчислимо два параметри

$u = \frac{h}{r} \bmod n$ та $v = \frac{s}{r} \bmod n$.

Знайдемо точку еліптичної кривої $u \cdot P + v \cdot Q = (x_0, y_0)$.

Параметр $\tilde{r} = x_0 \bmod n$ повинен співпадати з параметром r.

Якщо $\tilde{r} = r$, підпис признається справжнім.

Покажемо коректність пропонованого алгоритму формування і перевірки підпису:

$$\bar{R} = u \cdot P + v \cdot Q = \frac{h}{r} \cdot P + \frac{s}{r} \cdot \frac{1}{d} \cdot P =$$
$$= \left(\frac{h \cdot d + k \cdot r \cdot d - h \cdot d}{r \cdot d} \right) \cdot P = k \cdot P = R = (x_0, y_0)$$

Оскільки $\bar{R} = R$, то і $\tilde{r} = r$.

Приклад.

Нехай відкритими параметрами алгоритму ECGDSA є еліптична крива над простим полем $y^2 = x^3 + 5x + 9 \bmod 59$, базова точка кривої $P = (6,45)$, порядок точки $n = 73$.

Для генерування асиметричної пари ключів абонента А оберемо випадкове число $d = 18$, тоді $d^{-1} \bmod n = \frac{1}{18} \bmod 73 = 69$, $d^{-1} = 69$.

Обчислимо точку $Q = 69 \cdot P = (33,48)$.

Відкритим ключем абонента А є точка кривої $Q = (33,48)$.

Секретним ключем абонента А є число $d = 18$.

Для формування підпису абонент А хешує повідомлення M та отримує відповідне десяткове число $h = 44$.

Далі абонент А обирає випадкове число $k = 20$ та обчислює точку $C = 20 \cdot P = (17,13)$.

Звідси число $r = 17 \bmod 73 = 17$.

З використанням секретного ключа d та числа h абонент А обчислює параметр $s = (20 \cdot 17 - 44) \cdot 18 \bmod 73 = 72$.

Цифровим підписом є пара чисел $\langle r = 17, s = 72 \rangle$.

Підписане повідомлення має вигляд $\{M, \langle r = 17, s = 72 \rangle\}$.

Для перевірки підписаного абонентом А повідомлення $\{M, \langle r = 17, s = 72 \rangle\}$ використовуються хеш-образ повідомлення M, відкрити загальносистемні параметри алгоритму ECGDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка $P = (6,45)$, її порядок $n = 73$, десяткове число $h = 44$, що відповідає хеш-образу повідомлення M, відкритий ключ абонента А – точка кривої $Q = (33,48)$.

Обчислимо значення $u = \frac{44}{17} \bmod 73 = 67$ і $v = \frac{72}{17} \bmod 73 = 30$.

Знайдемо точку еліптичної кривої $67 \cdot P + 30 \cdot Q = (17,13)$ та параметр $\tilde{r} = 17 \bmod 73 = 17$.

Оскільки $\tilde{r} = r = 17$, підпис признається справжнім.

5.10 Мультиплікативне обертання многочленів

Розглянемо многочлени з коефіцієнтами, рівними 0 або 1. Складання і множення коефіцієнтів многочленів будемо здійснювати за модулем 2. Оскільки $-1 = 1 \bmod 2$, то сума і різниця многочленів дають однаковий результат. Наприклад, $x - x = x + x = 2 \cdot x = 0$.

Для многочленів $a(x) = x + 1$ і $b(x) = x^2 + x + 1$ маємо

$a(x) + b(x) = x^2$,

$a^2(x) = (x+1) \cdot (x+1) = x^2 + 2 \cdot x + 1 = x^2 + 1$.

Визначимо порівняння многочленів за модулем $m(x)$.

Два многочлени $a(x)$ і $b(x)$ *порівнянні* за модулем многочлена $m(x)$, якщо їхня різниця ділиться на многочлен $m(x)$:

$a(x) = b(x) \bmod m(x)$.

Наприклад, для модуля $m(x) = x^3 + x + 1$ многочлени $a(x) = x^3 + x^2$ і $b(x) = x^2 + x + 1$ є порівняними за цим модулем.

Два многочлени *взаємно прості*, якщо у них немає спільних дільників.

Нехай заданий модуль $m(x)$ і многочлен $a(x)$, степінь якого менше степеню многочлена $m(x)$.

Розглянемо порівняння відносно многочлена $g(x)$

$$a(x) \cdot g(x) = 1 \bmod m(x).$$

Розв'язок порівняння позначається через $a^{-1}(x)$ або $\frac{1}{a(x)}$.

Многочлен $g(x) = a^{-1}(x)$ називається *оберненим* за модулем $m(x)$ для многочлена $a(x)$.

Якщо многочлени $a(x)$ і $m(x)$ взаємно прості, то порівняння (Ж.1) має єдиний розв'язок.

Якщо многочлени $a(x)$ і $m(x)$ не взаємно прості, то порівняння (Ж.1) не має розв'язків.

Таким чином, якщо многочлен $m(x)$ є незвідним, то для будь-якого многочлена $a(x)$, степінь якого менше степеню многочлена $m(x)$, порівняння $a(x) \cdot g(x) = 1 \bmod m(x)$ має єдиний розв'язок.

Многочлен $g(x)$, що є розв'язком порівняння $a(x) \cdot g(x) = 1 \bmod m(x)$, взаємно простий з модулем $m(x)$.

Порівняння $a(x) \cdot g(x) = 1 \bmod m(x)$ може бути розв'язано за допомогою модифікованого алгоритму Евкліда.

Розширений алгоритм Евкліда

Розглядається порівняння

$$a(x)\cdot g(x)=1 \bmod m(x),$$

де многочлен $m(x)$ – незвідний.

1 Знаходимо залишки від ділення многочленів:

$$m(x)=a(x)\cdot q_0(x)+r_1(x)$$
$$a(x)=r_1(x)\cdot q_1(x)+r_2(x)$$
$$r_1(x)=r_2(x)\cdot q_2(x)+r_3(x)$$
……
$$r_{n-2}(x)=r_{n-1}(x)\cdot q_{n-1}(x)+r_n(x)$$
$$r_{n-1}(x)=r_n(x)\cdot q_n(x)$$

Якщо $r_n(x)\neq 1$, то $a(x)$ і $m(x)$ не взаємно прості, і порівняння (Ж.1) не має розв'язків.

2 Складаємо таблицю 5.2

Таблиця 5.2 – Модифікований розширений алгоритм Евкліда

i		0	1	2	…	$n-1$	n
$q_i(x)$		$q_0(x)$	$q_1(x)$	$q_2(x)$	⋯	$q_{n-1}(x)$	$q_n(x)$
$P_i(x)$	1	$P_0(x)$	$P_1(x)$	$P_2(x)$	⋯	$P_{n-1}(x)$	$P_n(x)$

$$P_0(x)=q_0(x), \qquad P_1(x)=q_1(x)\cdot P_0(x)+1,$$
$$P_i(x)=q_i(x)\cdot P_{i-1}(x)+P_{i-2}(x), \quad i\geq 2,$$
$$P_n(x)=m(x).$$

3 Розв'язком порівняння $a(x)\cdot g(x)=1 \bmod m(x)$ є многочлен

$$g(x)=P_{n-1}(x) \bmod m(x).$$

Приклад 1.

Знайдемо розв'язок порівняння відносно многочлена $g(x)$ за модулем незвідного многочлена $m(x) = x^5 + x^2 + 1$:

$$(x^4 + x + 1) \cdot g(x) = 1 \bmod (x^5 + x^2 + 1).$$

Застосуємо модифікований алгоритм Евкліда.

1 Знаходимо залишки від ділення многочленів:

$$x^5 + x^2 + 1 = (x^4 + x + 1) \cdot x + (x + 1)$$

$$x^4 + x + 1 = (x + 1) \cdot (x^3 + x^2 + x) + 1$$

$$x + 1 = 1 \cdot (x + 1)$$

2 Складаємо таблицю 5.3:

Таблиця 5.3 –Приклад застосування розширеного алгоритму Евкліда

i		0	1	2
$q_i(x)$		x	$x^3 + x^2 + x$	$x + 1$
$P_i(x)$	1	x	$x^4 + x^3 + x^2 + 1$	$x^5 + x^2 + 1$

3 Розв'язком порівняння

$$(x^4 + x + 1) \cdot g(x) = 1 \bmod (x^5 + x^2 + 1)$$

є многочлен $g(x) = x^4 + x^3 + x^2 + 1$.

Перевірка.

$$(x^4 + x + 1) \cdot (x^4 + x^3 + x^2 + 1) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + x + 1 =$$

$$= (x^5 + x^2 + 1)(x^3 + x^2 + x) + 1 = 1 \bmod (x^5 + x^2 + 1).$$

Приклад 2.

Знайдемо $\dfrac{x+1}{x^4+x^3+x^2+1} \bmod (x^5+x^2+1)$. За Прикладом 1 маємо $\dfrac{1}{x^4+x^3+x^2+1} \bmod (x^5+x^2+1) = x^4+x+1$.

Звідси

$$\frac{x+1}{x^4+x^3+x^2+1} = (x^4+x+1)\cdot(x+1) \bmod (x^5+x^2+1) = x^4.$$

5.11 Розв'язання квадратного рівняння в розширеному полі

Нехай задане скінченне поле $GF\left(2^m\right)$ і елементи поля $u, w \in GF\left(2^m\right)$.

Розглянемо квадратне рівняння відносно z:

$$z^2 + uz = w$$

Алгоритм розв'язання квадратного рівняння $z^2 + uz = w$ в розширеному полі

Вхідні дані алгоритму: квадратне рівняння $z^2 + uz = w$; $u, w \in GF\left(2^m\right)$; m –непарне число.

Результат виконання алгоритму – кількість розв'язків k квадратного рівняння й один з розв'язків цього рівняння, якщо $k > 0$.

Алгоритм розв'язування квадратного рівняння:

1. Якщо $u = 0$, то приймають $z = w^{2^{m-1}} = \sqrt{w}$, k=1 і переходять до кроку 8.
2. Якщо $w = 0$, то приймають $z = 0$, $k = 2$, і переходять до кроку 8.
3. Обчислюють елемент основного поля $v = wu^{-2}$.
4. Обчислюють слід елемента $tr(v)$.

5. Якщо $tr(v)=1$, то приймають $k=0$, $z=0$, і переходять до кроку 8.

6. Обчислюють напівслід елемента *v*, $ht=htr(v)$.

7. Обчислюють елемент основного поля $z=ht*u$, приймають $k=2$.

8. Результат виконання алгоритму: кількість розв'язків квадратного рівняння k та один з розв'язків z, якщо $k>0$.

Приклад 1.

Основне поле $GF(2^7)$ з примітивним многочленом $f(t)=t^7+t^5+t^3+t+1$.

Еліптична крива над основним полем

$$y^2+xy=x^3+x^2+B \bmod(f(t),2), \quad B=t+1.$$

Знайдемо точку еліптичної кривої.

Візьмемо випадковий елемент поля $X=1+t^3+t^4$ та обчислимо значення $W=X^3+X^2+B \bmod(f(t),2)$:

$W=t+t^2+t^4+t^5$.

Для отримання другої компоненти Y точки кривої розв'яжемо квадратне рівняння

$$Y^2+XY=W \bmod(f(t),2).$$

Обчислимо елемент основного поля $v=WX^{-2}$:

$v=t^2+t^4+t^5$

та його слід $tr(v)$:

$tr(v)=0.$

Обчислимо напівслід елемента *v*

$ht=htr(v)=t^2+t^3+t^4+t^5$.

Обчислимо елемент основного поля $Y=ht*X$:

$Y=1+t+t^3+t^4$.

Таким чином, знайдено точка еліптичної кривої

$(X,Y)=(1+t^3+t^4,1+t+t^3+t^4)$.

Приклад 2.

Візьмемо випадковий елемент поля $X = t$ та обчислимо значення $W = X^3 + X^2 + B \bmod(f(t),2)$:

$W = 1 + t + t^2 + t^3$.

Для отримання другої компоненти Y точки кривої розв'яжемо квадратне рівняння

$Y^2 + XY = W \bmod(f(t),2)$.

Обчислимо елемент основного поля $v = WX^{-2}$:

$v = 1 + t^3 + t^5$

та його слід $tr(v)$:

$tr(v) = 1$.

Оскільки $tr(v) = 1$, то квадратне рівняння розв'язків немає.

5.12 Еліптичні криві над розширеним полем

Еліптична крива над розширеним полем $GF(2^m)$ задається рівнянням

$$y^2 + xy = x^3 + ax^2 + b \bmod(2, f(t)),$$

де $f(t)$ – незвідний многочлен ступеня m над полем $GF(2)$; a, b – задані коефіцієнти кривої, $a, b \in GF(2^m)$, $b \neq 0$, $a \in \{0,1\}$, $x, y \in GF(2^m)$.

Пара елементів (x, y) поля $GF(2^m)$, що задовольняє рівнянню $y^2 + xy = x^3 + ax^2 + b \bmod(2, f(t))$, є точкою еліптичної кривої.

У множину точок кривої також включається нескінченно віддалена точка O.

Нехай точка $P(x, y)$ належить еліптичній кривій.

Точка $-P(x, x + y)$ називається *оберненою* (або *протилежною*) точки $P(x, y)$.

На множині точок еліптичної кривої над розширеним полем $GF(2^m)$ введемо *операцію додавання* «+».

Нехай точки $P_1(x_1, y_1)$ і $P_2(x_2, y_2)$ належать еліптичній кривій.

Визначимо суму точок $P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2)$.

Якщо $P_1 = \mathrm{O}$, то $P_3 = P_2$.

Якщо $P_2 = \mathrm{O}$, то $P_3 = P_1$.

Якщо $x_1 \neq x_2$, то

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a,$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1,$$

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2}.$$

Якщо $x_1 = x_2$, то

якщо $x_1 = 0$ або $y_1 = x_2 + y_2$, то $P_3 = \mathrm{O}$;

інакше

$$x_3 = x_1^2 + \frac{b}{x_1^2}, \qquad y_3 = x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right) \cdot x_3 + x_3.$$

Множина точок еліптичної кривої з операцією додавання «+» утворює адитивну абелеву групу з нейтральним (нульовим) елементом O, тобто виконуються умови групи:

5) для будь-яких точок кривої сума точок також належить кривій;
6) для будь-яких точок кривої P_1, P_2, P_3 виконується рівність $P_1 + (P_2 + P_3) = (P_1 + P_2) + P_3$;
7) у множині точок кривої існує одиничний (або нейтральний) елемент O: $P + \mathrm{O} = \mathrm{O} + P = P$;
8) для кожної точки кривої P існує обернений елемент, тобто протилежна точка $-P$ така, що $P + (-P) = \mathrm{O}$.

Порядком кривої називається число її точок..

Оскільки усі точки еліптичної кривої утворюють групу, то порядок цієї групи співпадає з порядком кривої.

В криптографії використовуються групи точок кривої або їхні підгрупи великого простого порядку.

Твірний елемент групи точок кривої або її підгрупи прийнято називати *базовою точкою*.

Якщо P є базовою точкою групи точок еліптичної кривої, то для неї виконується рівність $n \cdot P = \mathrm{O}$, де число n є порядком групи.

Теорема Хасе. Значення порядку n групи точок еліптичної кривої над полем Галуа $GF(2^m)$ має верхню і нижню межі:

$$2^m + 1 - 2\sqrt{2^m} \le n \le 2^m + 1 + 2\sqrt{2^m}$$

Порядком точки P еліптичної кривої називається найменше число k, таке що $k \cdot P = \mathrm{O}$.

Порядок точки є дільником порядку еліптичної кривої.

Приклад 1.

Дано еліптичну криву над розширеним полем $GF(2^3)$

$y^2 + xy = x^3 + x^2 + t \bmod(2, f(t) = t^3 + t + 1)$.

Еліптична крива містить 9 точок: $(t + t^2, 1)$, $(1, t^2)$, (t, t^2), $(1, 1 + t^2)$, $(1 + t, 1 + t^2)$, $(0, t + t^2)$, $(t, t + t^2)$, $(1 + t, t + t^2)$, $(t + t^2, 1 + t + t^2)$ і додаємо точку O. Отже, порядок кривої дорівнює 10.

Для виконання операцій в полі $GF(2^3)$ скористаємося таблицею степенів многочлена t за модулем $f(t) = t^3 + t + 1$:

t^1	t^2	t^3	t^4	t^5	t^6	t^7
t	t^2	$1 + t$	$t + t^2$	$1 + t + t^2$	$1 + t^2$	1

Перевіримо належність точки $P=(1,t^2)$ кривій $y^2+xy=x^3+x^2+t \bmod(2,t^3+t+1)$.

Оскільки значення

$y^2+xy=(t^2)^2+1\cdot t^2=t^2+t+t^2=t$ і

$x^3+x^2+t=1^3+1^2+t=t$

співпадають, точка $P=(1,t^2)$ належить кривій.

Знайдемо $2P$, $3P$…

$2P=(1,t^2)+(1,t^2)$.

$$x_3=1+\frac{t}{1}=1+t$$

$$y_3=1+\left(1+\frac{t^2}{1}\right)\cdot(1+t)+1+t=1+(1+t^2)\cdot(1+t)+1+t=t+t^2$$

Таким чином, $2P=(1+t,t+t^2)$.

$3P=P+2P=(1,t^2)+(1+t,t+t^2)$

$$\lambda=\frac{t^2+t+t^2}{1+1+t}=1,$$

$$x_3=1+1+1+1+t+1=1+t,$$

$$y_3=1(1+1+t)+1+t+t^2=1+t^2.$$

Таким чином, $3P=(1+t,1+t^2)$.

$4P=P+3P=(1,t^2)+(1+t,1+t^2)$

$$\lambda=\frac{t^2+1+t^2}{1+1+t}=\frac{1}{t}=\frac{t^7}{t}=t^6=1+t^2,$$

$$x_3=(1+t^2)^2+1+t^2+1+1+t+1=1+t^4+t^2+t=1,$$

$$y_3=(1+t^2)(1+1)+1+t^2=1+t^2.$$

Таким чином, $4P=(1,1+t^2)$.

Оскільки точки P і $4P$ протилежні, то $5P = P + 4P = \mathrm{O}$.

Точки $2P$ і $3P$ також є протилежними.

Звідси порядок точки P дорівнює 5.

Точка $P = (1, t^2)$ утворює циклічну підгрупу точок еліптичної кривої простого порядку 5: $(1, t^2)$, $(1+t, t+t^2)$, $(1+t, 1+t^2)$, $(1, 1+t^2)$, O.

Приклад 2. Обчислення порядку точки еліптичної кривої

Дано еліптичну криву над розширеним полем $GF(2^5)$

$y^2 + xy = x^3 + x^2 + t^2 + t^4 \bmod(2, t^5 + t^3 + 1)$.

Знайдемо порядок точки кривої $P = (1 + t + t^2, t^4)$.

Перевіримо належність точки $P = (1 + t + t^2, t^4)$ кривій $y^2 + xy = x^3 + x^2 + t^2 \bmod(2, t^5 + t^3 + 1)$.

Оскільки значення

$y^2 + xy = (t^4)^2 + (1 + t + t^2) \cdot t^4 \bmod(2, t^5 + t^3 + 1) = 1 + t^4$ і

$x^3 + x^2 + t = (1 + t + t^2)^3 + (1 + t + t^2)^2 + t^2 \bmod(2, t^5 + t^3 + 1) = 1 + t^4$

співпадають, точка $P = (1 + t + t^2, t^4)$ належить кривій.

Знайдемо порядок точки $P = (1 + t + t^2, t^4)$ за допомогою алгоритму великих-малих кроків.

По теоремі Хасе максимальна оцінка порядку групи точок кривої дорівнює $2^5 + 1 + 2\sqrt{2^5} = 44.31$.

Звідси максимальне можливий порядок точки $P = (1 + t + t^2, t^4)$ дорівнює $n = 44$, $m = \left| \sqrt{44} \right| = 7$.

Побудуємо таблицю:

$1 \cdot P$	$(1+t+t^2, t^4)$
$2 \cdot P$	$(1+t^2+t^3, 1+t^2+t^3)$
$3 \cdot P$	$(1+t, t+t^2)$
$4 \cdot P$	$(1+t+t^2+t^3, t+t^3+t^4)$
$5 \cdot P$	$(1+t^2+t^4, 1+t+t^2)$
$6 \cdot P$	$(1, 1+t^2)$
$7 \cdot P$	$(1+t+t^3+t^4, t^2+t^3)$

Обчислимо

$$T = -m \cdot P = -(1+t+t^3+t^4, t^2+t^3) = (1+t+t^3+t^4, 1+t+t^2+t^4)$$

Положимо $S := \mathrm{O}$.

$i = 0$,

$$S := \mathrm{O} + (1+t+t^3+t^4, 1+t+t^2+t^4) = (1+t+t^3+t^4, 1+t+t^2+t^4)$$

$i = 1$,

$$S := (1+t+t^3+t^4, 1+t+t^2+t^4) + (1+t+t^3+t^4, 1+t+t^2+t^4) =$$

$$= (1+t, t+t^2);$$

$i = 2$, $j = 3$, $x = 7 \cdot 2 + 3 = 17$

Порядок точки $P = (1+t+t^2, t^4)$ дорівнює 17.

5.13 Корейський стандарт цифрового підпису EC-KCDSA

Протокол підпису EC-KCDSA над розширеним полем вперше був запропонований в 2001 році. Після модифікації він є чинним стандартом цифрового підпису Південної Кореї.

В якості хеш-функцій можуть використовуватися функції RIPEMD-160, SHA-1 (= SHA-160), SHA-224, SHA-256, SHA-384 и SHA-512, хеш-образ яких дорівнює відповідно 160, 224, 256, 384 и 512 бітів.

Відкритими параметрами алгоритму є еліптична крива, базова точка кривої P з відомим простим порядком n. Відкритим ключем підписанта є точка еліптичної кривої Q, $Q = (d^{-1} \bmod n) \cdot P$, його секретним ключем є число d.

Для забезпечення криптостійкості цифрового підпису порядок базової точки має бути досить великим.

Цифровий підпис формується абонентом А за допомогою його секретного ключа, перевірка підпису здійснюється абонентом В з використанням відкритого ключа абонента А.

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(2^m)$

$$y^2 + xy = x^3 + ax^2 + b \mod(2, f(t)),$$

де $a, b \in GF(2^m)$, разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; $H(\bullet)$ – обрана функція хешування.

Генерація ключів

Згенеруємо секретний та відкритий ключі абонента А.

Оберемо випадкове число d, $2 \le d \le n-2$.

Обчислимо точку кривої $Q = (d^{-1} \bmod n) \cdot P$.

Секретним ключем абонента А є число d.

Відкритим ключем абонента А є точка кривої Q.

Формування цифрового підпису

Підписант А обчислює хеш-образ повідомлення M за допомогою обраної хеш-функції $H = H(M)$.

Підписант А обирає випадкове число k, $2 \le k \le n-2$, обчислює точку $C = k \cdot P = (x_C, y_C)$ та хеш-образ координати x_C : $R = H(x_C)$. Параметр R не повинен бути 0.

Обчислюється параметр $W = R \oplus H$, який конвертується в десяткове число w.

З використанням секретного ключа d підписант А обчислює значення $s = (k - w) \cdot d \bmod n$ (число s не повинно бути 0), яке конвертується в двійкове число S.

Цифровим підписом є пара двійкових чисел $\langle R, S \rangle$.

Підписане повідомлення має вигляд $\{M, \langle R, S \rangle, text\}$.

Перевірка цифрового підпису

Для перевірки підписаного абонентом А повідомлення $\{M, \langle R, S \rangle, text\}$ використовуються хеш-образ $H = H(M)$ повідомлення M, відкрити загальносистемні параметри алгоритму EC-KCDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка P, її порядок n, відкритий ключ абонента А – точка кривої Q.

Обчислюється параметр $W = R \oplus H$, який конвертується в десяткове число w.

Далі обчислюється точка еліптичної кривої

$\tilde{C} = w \cdot P + s \cdot Q = (x_{\tilde{C}}, y_{\tilde{C}})$

та хеш-образ координати $x_{\tilde{C}}$: $\tilde{R} = H(x_{\tilde{C}})$.

Якщо $\tilde{R} = R$, підпис признається справжнім.

Покажемо коректність пропонованого алгоритму формування і перевірки підпису:

$$\tilde{C} = w \cdot P + s \cdot Q = w \cdot P + (k - w) \cdot d \cdot \frac{1}{d} \cdot P = k \cdot P = C = (x_C, y_C)$$

Оскільки $\tilde{C} = C$, то і $\tilde{R} = R$.

Приклад.

Нехай відкритими параметрами алгоритму EC-KCDSA є еліптична крива над розширеним полем $GF(2^5)$

$$y^2 + xy = x^3 + x^2 + t^2 + t^4 \bmod(2, t^5 + t^3 + 1),$$

базова точка кривої $P=(1+t+t^2,t^4)$, її порядок $n=17$, $|n|-1=4$, і для хешування повідомлень обрана хеш-функція SHA-256.

Для генерування асиметричної пари ключів абонента А оберемо випадкове число $d=10$ та обчислимо точку

$$Q=\left(\frac{1}{10}\bmod 17\right)\cdot P=(1+t^2+t^4,t+t^4).$$

Відкритим ключем абонента А є точка кривої $Q=(1+t^2+t^4,t+t^4)$.

Секретним ключем абонента А є число $d=10$.

Нехай абоненту А необхідно сформувати підпис для повідомлення «world».

Для формування підпису абонент А хешує повідомлення «world» та отримує відповідне двійкове число $H=0111$. (Молодші $|n|-1=4$ розряди хеш-образу повідомлення «world» формують двійкове число H.)

Далі абонент А обирає випадкове число $k=9$, обчислює точку $C=9\cdot P=(1+t^2+t^3+t^4,t+t^3)$ та хеш-образ координати x_C: $R=H(1+t\wedge 2+t\wedge 3+t\wedge 4)\to 0001$.

Потім обчислюється параметр W, який конвертується в десяткове число $w=6$: $W=R\oplus H=0001_2\oplus 0111_2=0110_2\to 6_{10}$.

З використанням секретного ключа d підписант А обчислює значення

$$s=(k-w)\cdot d \bmod n=(9-6)\cdot 10 \bmod 17=13_{10}\to S=1101_2.$$

Цифровим підписом є пара двійкових чисел $\langle R=0001,S=1101\rangle$

Підписане повідомлення має вигляд $\{\text{"world"},\langle 0001,1101\rangle,text\}$.

Для перевірки підписаного абонентом А повідомлення $\{\text{"world"},\langle 0001,1101\rangle,text\}$ використовуються хеш-образ

$H = H(M)$ повідомлення «world», відкрити загальносистемні параметри алгоритму EC-KCDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка P, її порядок n, відкритий ключ абонента А – точка кривої Q.

Абонент В обчислює параметр $W = R \oplus H = 0110$, який конвертується в десяткове число $w = 6$. Абонент В конвертує двійкове число $S = 1101$ в десяткове число $s = 13$ та обчислює точку еліптичної кривої

$$\tilde{C} = 6 \cdot P + 13 \cdot Q = (1 + t^2 + t^3 + t^4, t + t^3)$$

та хеш-образ координати $x_{\tilde{C}}$:

$$\tilde{R} = H(x_{\tilde{C}}) = H(1 + t \wedge 2 + t \wedge 3 + t \wedge 4) \to 0001.$$

Оскільки $\tilde{R} = R$, підпис признається справжнім.

5.14 Український стандарт цифрового підпису ДСТУ 4145-2002

Цей стандарт встановлює механізм цифрового підписування, що базується на властивостях груп точок еліптичних кривих над полями $GF\left(2^m\right)$, та правила застосування цього механізму до повідомлень, що пересилаються каналами зв'язку та/або обробляються у комп'ютеризованих системах загального призначення. Застосування цього стандарту гарантує цілісність підписаного повідомлення, автентичність його автора та неспростовність авторства.

Цифровий підпис створюється за допомогою операцій над точками еліптичної кривої після хешування повідомлення. В якості хеш-функції використовується стандарт України ДСТУ 7564:2014 або Міждержавний стандарт ГОСТ 34.311-95, хеш-образ якого дорівнює 256 бітів.

Відкритими параметрами алгоритму є еліптична крива, базова точка кривої P з відомим простим порядком n. Відкритим ключем підписанта є точка еліптичної кривої Q, $Q = -d \cdot P$, його секретним ключем є число d. Для забезпечення криптостійко-

сті цифрового підпису порядок базової точки еліптичної кривої, просте непарне число, має бути досить великим: $n \geq \max(2^{160}, 4(\lfloor\sqrt{2^m}\rfloor + 1))$.

Цифровий підпис формується абонентом А за допомогою його секретного ключа, перевірка підпису здійснюється абонентом В з використанням відкритого ключа абонента А.

Загальносистемні параметри

Еліптична крива над розширеним полем $GF(2^m)$

$y^2 + xy = x^3 + ax^2 + b$;

де $a, b \in GF(2^m)$, $b \neq 0$, а $\in \{0,1\}$, $f(t)$ – незвідний многочлен степеню m ; базова точка еліптичної кривої $P \neq O$ великого простого порядку n, $|n|$ – число двійкових розрядів в n; H – функція хешування ГОСТ 34.311-95.

Генерація ключів

Згенеруємо секретний та відкритий ключі абонента А.

Оберемо випадкове число d, $2 \leq d \leq n-2$.

Обчислимо точку $Q = -d \cdot P$.

Секретним ключем абонента А є число d.

Відкритим ключем абонента А є точка кривої Q.

Формування цифрового підпису

Підписант А обчислює хеш-образ повідомлення M за допомогою хеш-функції ГОСТ 34.311-95. Отримане двійкове число $H(M)$ конвертується в елемент поля $h \in GF(2^m)$. Для цього використовують m молодших бітів $H(M)$.

Оберемо випадкове число k, $1 < k \leq n-1$.

Обчислимо точку $R = k \cdot P = (x_R, y_R)$ та елемент поля $y = h \cdot x_R \bmod f(t)$.

Молодші $|n|-1$ розряди елемента поля y формують десяткове число r. Число r не повинно бути 0.

З використанням секретного ключа d та хеш-образу повідомлення h обчислимо значення $s = k + d \cdot r \bmod n$. Число s не повинно бути 0.

Цифровим підписом є пара чисел $\langle r, s \rangle$.

Перевірка цифрового підпису

Для перевірки підписаного абонентом А повідомлення $\{M, \langle r, s \rangle\}$ використовуються хеш-образ повідомлення M, відкриті загальносистемні параметри алгоритму ДСТУ 4145-2002 та відкритий ключ підписанта, тобто еліптична крива, базова точка P, її порядок n, елемент поля h, що відповідає хеш-образу повідомлення M, відкритий ключ абонента А – точка кривої Q.

Абонент В обчислює точку еліптичної кривої $s \cdot P + r \cdot Q = (x_0, y_0)$ та елемент поля $y = h \cdot x_0 \bmod f(t)$. Молодші $|n|-1$ розряди елемента поля y формують десяткове число r'.

Параметр r' повинен співпадати з параметром r.

Якщо $r' = r$, підпис визнається справжнім.

Приклад.

Нехай відкритими параметрами алгоритму ДСТУ 4145-2002 є еліптична крива над розширеним полем $GF(2^5)$

$$y^2 + xy = x^3 + x^2 + t^2 + t^4 \bmod(2, t^5 + t^3 + 1),$$

базова точка кривої $P = (1 + t + t^2, t^4)$, її порядок $n = 17$, $|n| - 1 = 4$.

Для генерування асиметричної пари ключів абонента А оберемо випадкове число $d = 5$ та обчислимо точку $Q = -5 \cdot P = (1 + t^2 + t^4, t + t^4)$.

Відкритим ключем абонента А є точка кривої $Q = (1 + t^2 + t^4, t + t^4)$.

Секретним ключем абонента А є число $d = 5$.

Для формування підпису абонент А хешує повідомлення M та отримує відповідний елемент поля $h = t + t^3$.

Далі абонент А обирає випадкове число $k = 9$ та обчислює точку $R = 9 \cdot P = (1 + t^2 + t^3 + t^4, t + t^3)$ і елемент поля y: $y = (t + t^3) \cdot (1 + t^2 + t^3 + t^4) \bmod (t^5 + t^3 + 1) = 1 + t^2 + t^3$.

Елементу поля y відповідає двійкове число 01101. Молодші $|n| - 1 = 4$ розряди формують десяткове число r: $r = 1101_2 = 13_{10}$.

Звідси $r = 13$.

З використанням секретного ключа d абонент А обчислює параметр $s = 9 + 5 \cdot 13 \bmod 17 = 6$.

Цифровим підписом є пара чисел $\langle 13,6 \rangle$.

Підписане повідомлення має вигляд $\{M, \langle 13,6 \rangle\}$.

Для перевірки підписаного абонентом А повідомлення $\{M, \langle 13,6 \rangle\}$ використовуються хеш-образ повідомлення M, відкриті загальносистемні параметри алгоритму ДСТУ 4145-2002 та відкритий ключ підписанта, тобто еліптична крива, базова точка $P = (1 + t + t^2, t^4)$, її порядок $n = 17$, елемент поля $h = t + t^3$, що відповідає хеш-образу повідомлення M, відкритий ключ абонента А – точка кривої $Q = (1 + t^2 + t^4, t + t^4)$.

Абонент В обчислює точку еліптичної кривої $6 \cdot P + 13 \cdot Q = (1,1 + t^2) + (1 + t, t + t^2) = (1 + t^2 + t^3 + t^4, t + t^3)$ та елемент поля $y = (t + t^3) \cdot (1 + t^2 + t^3 + t^4) \bmod (t^5 + t^3 + 1) = 1 + t^2 + t^3$. Елементу поля y відповідає двійкове число 01101. Молодші $|n| - 1 = 4$ розряди формують десяткове число r': $r' = 1101_2 = 13_{10}$.

Звідси $r' = 13$.

Оскільки $r' = r$, підпис визнається справжнім.

Контрольні питання

1. Дайте визначення поля.
2. Що таке поле Галуа?
3. Приведіть приклади простого і розширеного полів.
4. Що таке мультиплікативна група поля?
5. Що таке примітивний елемент поля?
6. Наведіть приклади простих чисел m, для яких виконується умова $m = 3 \bmod 4$.
7. Наведіть приклади простих чисел m, для яких виконується умова $m = 5 \bmod 8$.
8. Як задається еліптична крива над простим полем.
9. Як визначити точку, протилежну даній?
10. Які параметри еліптичної кривої необхідно знати для її застосування?
11. Дати визначення порядка групи точок еліптичної кривої.
12. Опишить алгоритм пошуку точки еліптичної кривої над простим полем.
13. Дати визначення порядка точки еліптичної кривої.
14. Як визначити базову точку?
15. Сформулюйте задачу дискретного логарифмування в групі точок еліптичної кривої.
16. Опишіть модіфікований алгоритм великих - малих кроків.
17. В яких криптографічних алгоритмах необхідно розв'язувати задачу дискретного логарифмування в групі точок еліптичної кривої?
18. Опишіть схему Діффі-Хеллмана генерування спільного секретного ключа в групі точок еліптичної кривої.
19. На чому заснована криптографічна стійкість схеми Діффі-Хеллмана?
20. Як визначити бітовий розмір спільного ключа по вхідних відкритих параметрах?
21. Як побудувати схему Діффі-Хеллмана для трьох користувачів?
22. Які параметри є загальносистемними в стандарті підпису ECDSA?
23. Опишіть процедуру генерації ключів у ECDSA.

24. Опишіть процедуру формування і перевірки підпису в ECDSA.
25. Які параметри впливають на криптостійкість підпису ECDSA?
26. Які параметри є загальносистемними в стандарті підпису EC-GDSA?
27. Опишіть процедуру генерації ключів у EC-GDSA.
28. Опишіть процедуру формування підпису в EC-GDSA.
29. Опишіть процедуру перевірки підпису в EC-GDSA.
30. Які параметри впливають на криптостійкість підпису EC-GDSA?
31. Дайте визначення оберненого многочлена за модулем заданого многочлена.
32. При якій умові існує обернений многочлен за модулем.
33. При якій умові не існує оберненого многочлена за модулем.
34. Опишіть модифікований алгоритм Евклида для обертання многочленів за модулем.
35. У яких криптографічних алгоритмах необхідно обертати многочлени за незвідним модулем?
36. Опишить алгоритм розв'язання квадратного рівняння в розширеному полі.
37. Як задається еліптична крива над розширеним полем?
38. Що є точкою еліптичної кривої над розширеним полем?
39. Як перевірить приналежність точки заданої еліптичної кривої?
40. Як знайти точку еліптичної кривої над розширеним полем?
41. Як визначити точку, протилежну даній?
42. Дати визначення порядка групи точок еліптичної кривої.
43. Дати визначення порядка точки еліптичної кривої.
44. У яких криптографічних алгоритмах застосовуються еліптичні криві над розширеним полем ?
45. Які параметри є загальносистемними в стандарті підпису EC-KCDSA?
46. Опишіть процедуру генерації ключів у EC-KCDSA.
47. Опишіть процедуру формування підпису в EC-KCDSA.
48. Опишіть процедуру перевірки підпису в EC-KCDSA.
49. Які параметри впливають на криптостійкість підпису EC-KCDSA?

50. Які параметри є загальносистемними в стандарті підпису ДСТУ 4145-2002?
51. Опишіть процедуру генерації ключів у ДСТУ 4145-2002.
52. Опишіть процедуру формування підпису в ДСТУ 4145-2002.
53. Опишіть процедуру перевірки підпису в ДСТУ 4145-2002.
54. Які параметри впливають на криптостійкість підпису ДСТУ 4145-2002?

6 РІЗНІ СХЕМИ ЦИФРОВОГО ПІДПИСУ

При формуванні електронних документів у ряді випадків виникає необхідність підписування документів декількома учасниками. Підпис, сформований колективом рівноправних учасників підписання під спільним документом, називається *колективним*, або *мультипідписом* (multisignature).

Схеми мультипідпису призначені для розв'язання задачі одночасного підписання контрактів і скорочення розміру підписів до документів, що підписуються двома й більше суб'єктами. Особливо актуальним є питання про скорочення розміру підпису у випадках, коли електронний цифровий підпис вноситься в шрихкод або іншу машиночитаєму мітку, що наноситься на матеріальний об'єкт.

В протоколі мультипідпису здійснюється обмін відкритими параметрами по мережах зв'язку, причому кожен учасник створює свою частину підпису, після чого формується загальний підпис. Для перевірки мультипідпису формується спільний відкритий ключ, який залежить від відкритих ключів учасників підписання електронного документа.

Якщо учасники підписання не є рівноправними, може виникнути необхідність підписання різних документів групою осіб, кожна із котрих має право підписувати тільки свій документ. Наприклад, директор, бухгалтер, завідувач відділу кадрів, технолог підписують кожний свій електронний документ с використанням свого особистого ключа. С метою зменшення довжини підпису пропонується формування єдиного, *композиційного*, або *агрегованого* (aggregate signature), підпису різних документів на базі елементів особистих підписів. Перевірка такого агрегованого підпису потребує знання відкритих ключів кожного із учасників підписання і відповідних кожному електронних документів.

Схеми агрегованого підпису мають призначення, аналогічне призначенню мультипідписів, але надають розширені можливості: одночасне підписання пакета контрактів і підписання різних документів різними підмножинами користувачів, що брали участь у формуванні єдиного пакета документів.

Кільцевий підпис (ring signature) є варіантом реалізації електронного підпису, при якому відомо, що повідомлення підписа-

но одним з членів списку потенційних підписантів, але не розкривається, ким саме. Підписант самостійно формує список з довільного числа різних осіб, включаючи в нього і себе. Для накладення підпису підписанту не потрібні дозвіл, сприяння або допомога з боку включених до списку осіб, використовуються тільки відкриті ключі всіх членів списку і закритий ключ лише самого підписанта.

Перший алгоритм кільцевого підпису був розроблений Рональдом Рівестом, Аді Шаміром і Яелем Тауманом, і представлено в 2001 році на міжнародній конференції Asiacrypt 2001 [7]. За твердженням авторів, вони намагалися в назві підкреслити відсутність центральної або координуючої структури при формуванні такого підпису: «... кільця являють собою геометричні фігури з однорідною периферією і без центру».

В протоколі кільцевого підпису обраний від групи підписант обчислює підписи для кожного учасника групи і формує кінцевий кортеж підписів. Перевірка підпису здійснюється за допомогою спарювання Вейля точок еліптичної кривої. При цьому використовуються відкриті ключі всіх учасників групи. В процедурі перевірки неможливо визначити, хто саме підписав електронний документ від імені групи.

Схеми кільцевого підпису призначені для розв'язання задачі анонімного підписання документів, наприклад, рецензій, від імені деякого колективу.

Недоліком схем кільцевого підпису є великий розмір кінцевого кортежу підписів. Тому їх недоцільно використовувати в ситуаціях, де величина розміру підписів є критичною.

Сліпим (blind signature) називається підпис, який сформовано під замаскованим повідомленням. В процесі підписання підписант не має можливості ознайомиться зі змістом відкритого (незамаскованого) повідомлення.

Схеми сліпого підпису призначені для розв'язання задачі забезпечення анонімності (невідстежуваності) користувачів у системах електронної готівки, в системах таємного електронного голосування.

В типовій схемі сліпого підпису, як правило, приймають участь три сторони – емітент документу, підписант та валідатор. Емітент створює документ, який має підписати підписант. При

цьому, підписант не має знати вмісту документа та вигляду остаточного підпису, тому емітент маскує документ за допомогою певного криптографічного перетворення. Підписант підписує замаскований документ, а емітент на основі його підпису формує остаточний підпис під документом у відкритому вигляді згідно зі схемою сліпого підпису. Валідатор перевіряє правильність підпису за допомогою відкритого ключа емітента.

Наприклад, в схемі електронного голосування в якості емітента виступає виборець, підписанта – дільнична виборча комісія, валідатора – центральна виборча комісія. Виборець заповнює бюлетень, маскує його, щоб зберегти конфіденційність голосу, і передає на підпис у дільничну виборчу комісію. Комісія, засвідчивши особу виборця, підписує його бюлетень, не знаючи, за кого той проголосував. Центральна виборча комісія отримує бюлетень у відкритому вигляді і перевіряє валідність підпису дільничної комісії, не знаючи при цьому, хто заповнив цей бюлетень.

Відомі протоколи сліпого підпису реалізуються на основі алгоритмів електронного цифрового підпису, що використають три обчислювальне складні задачі: факторизація натурального числа, знаходження дискретного логарифма по простому модулі, знаходження дискретного логарифма на еліптичній кривій.

В розглянутих нижче протоколах використається в якості математичної структури група точок еліптичної кривої над скінченним полем. Стійкість протоколів заснована на складності задачі знаходження дискретного логарифма на еліптичній кривій.

Для хешування електронного документу може бути запропоновано використання відомих стандартів.

6.1 Протокол мультипідпису електронного документу на еліптичної кривої над простим полем

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(p)$

$$y^2 = x^3 + ax + b \mod p,$$

де $a, b \in GF(p)$, $b \neq 0$, разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq O$

простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; $H(\bullet)$ – функція хешування; δ – допоміжне просте багаторозрядне двійкове число (введення допоміжного числа δ дозволяє скоротити першу частину цифрового підпису).

Генерація ключів

Кожний i-ий ($i = 1,2,...,t$) користувач має асиметричну пару ключів: особистий d_i $-1 < d_i < n$ – та відкритий $Q_i = d_i P$.

Формування цифрового підпису

Нехай колектив користувачів, $i = 1,2,...,t$, має підписати електронний документ M з хеш-образом $H(M)$. Молодші $|\delta| - 1$ розряди хеш-образу $H(M)$ формують десяткове число h, яке використається при обчисленні цифрового підпису.

Кожний підписант обирає одноразовий випадковий секретний ключ k_i, $1 < k_i < n$, обчислює координати точки

$$R_i = k_i P$$

та надає їх для колективного використання.

Далі обчислюється сума всіх точок R_i, $i = 1,2,...,t$:

$$R = \sum_{i=1}^{t} R_i = (xR, yR),$$

після чого формується число

$$r = h \cdot xR \mod \delta.$$

При $r = 0$ обираються нові випадкові секретні ключі k_i.

Потім кожний користувач i за допомогою свого особистого ключа d_i та значення k_i обчислює свою частину підпису

$$s_i = k_i - d_i \cdot r \mod n,$$

після чого генерується підпис s:

$$s = \sum_{i=1}^{t} s_i \bmod n .$$

Число s не може бути рівним 0. При $s = 0$ процедура підпису повторюється.

Мультипідписом є пара чисел $\langle r, s \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle r, s \rangle$ під електронним документом M здійснюється за допомогою додаткової точки еліптичної кривої

$$Q = \sum_{i=1}^{t} Q_i ,$$

яка залежить від відкритих ключів Q_i учасників підписання.

Обчислюється точка $\tilde{R}$ еліптичної кривої

$$\tilde{R} = sP + rQ = (x\tilde{R}, y\tilde{R})$$

після чого обчислюються хеш-образ документу $H(M)$, відповідне десяткове число h та формується число $\tilde{r} = h \cdot x\tilde{R} \bmod \delta$.

Якщо $\tilde{r} = r$, мультипідпис електронного документу M признається справжнім.

Покажемо коректність пропонованого алгоритму формування і перевірки мультипідпису:

$$\tilde{R} = sP + rQ = \left(\sum_{i=1}^{t} s_i \right) P + r \left(\sum_{i=1}^{t} Q_i \right) = \left(\sum_{i=1}^{t} k_i - d_i r \right) P + r \left(\sum_{i=1}^{t} d_i P \right) =$$

$$= \left(\sum_{i=1}^{t} k_i \right) P = \sum_{i=1}^{t} R_i = R.$$

Оскільки $\tilde{R} = R$, то і $\tilde{r} = r$.

Приклад.

Оберемо загальні параметри:

основне поле – скінченне поле $GF(17)$;

еліптична крива над основним полем $y^2 = x^3 + 2x + 6 \mod 17$.

Базова точка еліптичної кривої $P = (2,1)$ має порядок $n = 11$.

Допоміжне просте багаторозрядне двійкове число $\delta = 7$, $|\delta| = 3$.

Нехай число користувачів $t = 2$.

Відповідні особисті ключі є $d_1 = 8$, $d_2 = 5$.

Тоді відкрити ключі $Q_1 = (6,8)$, $Q_2 = (1,3)$.

Нехай хеш-образ електронного документу M дорівнює $h = 2$.

Кожний підписант обирає одноразовий випадковий секретний ключ k_i: $k_1 = 3$, $k_2 = 4$, та обчислює координати точки R_i: $R_1 = (6,9)$, $R_2 = (13,11)$.

Далі обчислюється R – сума всіх точок R_i: $R = (13,6)$, після чого формується число r: $r = 2 \cdot 13 \bmod 7 = 5$, $r = 5$.

Потім кожний користувач i за допомогою свого особистого ключа d_i та значення k_i обчислює свою частину підпису: $s_1 = 3 - 8 \cdot 5 \bmod 11 = 7$, $s_2 = 4 - 5 \cdot 5 \bmod 11 = 1$, $s_1 = 7$, $s_2 = 1$, після чого генерується підпис s: $s = 8$.

Мультипідписом є пара чисел $\langle r, s \rangle = \langle 5,8 \rangle$.

Перевірка підпису $\langle r, s \rangle = \langle 5,8 \rangle$ під електронним документом M здійснюється за допомогою додаткової точки кривої Q, яка залежить від відкритих ключів Q_i учасників підписання: $Q = (11,4)$.

Обчислюється точка $\tilde{R}$ еліптичної кривої: $sP = (6,8)$, $rQ = (2,16)$, $\tilde{R} = (13,6)$. Далі обчислюються хеш-образ документу $H(M)$, відповідне десяткове число $h = 2$ та формується число $\tilde{r} = 2 \cdot 13 \bmod 7 = 5$, $\tilde{r} = 5$.

Оскільки $\tilde{r} = r$, мультипідпис електронного документу M признається справжнім.

6.2 Протокол агрегованого підпису різних документів на еліптичної кривої над простим полем

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(p)$

$$y^2 = x^3 + ax + b \mod p,$$

де $a,b \in GF(p)$, $b \neq 0$, разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; $H(\bullet)$ – функція хешування; δ – допоміжне просте багаторозрядне двійкове число (введення допоміжного числа δ дозволяє скоротити першу частину цифрового підпису).

Генерація ключів

Кожний i-ий ($i = 1,2,...,t$) користувач має асиметричну пару ключів: особистий d_i – $1 < d_i < n$ – та відкритий $Q_i = d_i P$.

Формування цифрового підпису

Нехай колектив із t користувачів має створити агрегований підпис під набором електронних документів $\{M_1, M_2, ..., M_t\}$, причому кожен користувач i, $i = 1,2,...,t$, має підписати свій електронний документ M_i з хеш-образом $H(M_i)$. Молодші $|n|-1$ розряди хеш-образу $H(M_i)$ формують десяткове число h_i, яке використається при обчисленні цифрового підпису.

Кожний підписант обирає одноразовий випадковий секретний ключ k_i, $1 < k_i < n$, обчислює координати точки

$$R_i = k_i P$$

та надає їх для подальшого використання.

Далі обчислюється сума всіх точок R_i, $i = 1,2,...,t$:

$$R = \sum_{i=1}^{t} R_i = (xR, yR),$$

після чого формується число $r = xR \mod \delta$.

При $r = 0$ обираються нові випадкові секретні ключі k_i.

Потім кожний користувач i за допомогою свого особистого ключа d_i, значення k_i, хеш-образу h_i та числа r обчислює свою частину підпису

$$s_i = k_i - d_i \cdot h_i \cdot r \mod n,$$

після чого генерується підпис s:

$$s = \sum_{i=1}^{t} s_i \mod n.$$

Параметр підпису s не може бути рівним 0. При $s = 0$ процедура підпису повторюється.

Агрегованим підписом є пара чисел $\langle r, s \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle r, s \rangle$ під електронними документами $\{M_1, M_2, ..., M_t\}$ з відповідними хеш-образами $\{h_1, h_2, ..., h_t\}$ здійснюється за допомогою додаткової точки еліптичної кривої

$$Q = \sum_{i=1}^{t} h_i \cdot Q_i,$$

яка залежить від відкритих ключів Q_i учасників підписання та хеш-образів електронних документів h_i.

Обчислюється точка $\widetilde{R}$ еліптичної кривої

$$\widetilde{R} = sP + rQ = (x\widetilde{R}, y\widetilde{R})$$

після чого формується число

$$\widetilde{r} = x\widetilde{R} \mod \delta.$$

Якщо $\tilde{r} = r$, агрегований підпис під набором електронних документів $\{M_1, M_2, ..., M_t\}$ признається справжнім.

Покажемо коректність пропонованого алгоритму формування і перевірки агрегованого підпису:

$$\tilde{R} = sP + rQ = \left(\sum_{i=1}^{t} s_i\right)P + r\left(\sum_{i=1}^{t} h_i \cdot Q_i\right) =$$

$$= \left(\sum_{i=1}^{t} k_i - d_i h_i r\right)P + r\left(\sum_{i=1}^{t} h_i d_i P\right) = \left(\sum_{i=1}^{t} k_i\right)P = \sum_{i=1}^{t} R_i = R.$$

Оскільки $\tilde{R} = R$, то і $\tilde{r} = r$.

Приклад.

Оберемо загальні параметри:

основне поле – скінченне поле $GF(13)$;

еліптична крива над основним полем

$$y^2 = x^3 + 2x + 4 \mod 13.$$

Базова точка еліптичної кривої $P = (7,6)$ має порядок $n = 17$.

Допоміжне просте багаторозрядне двійкове число $\delta = 7$.

Нехай число користувачів $t = 3$.

Відповідні особисті ключі є $d_1 = 8$, $d_2 = 5$, $d_3 = 15$.

Тоді відкрити ключі $Q_1 = (5,10)$, $Q_2 = (8,8)$, $Q_3 = (9,7)$.

Нехай хеш-образи електронних документів M_1, M_2, M_3 дорівнюють відповідно $h_1 = 9$, $h_2 = 10$, $h_3 = 13$.

Кожний підписант обирає одноразовий випадковий секретний ключ k_i: $k_1 = 5$, $k_2 = 10$, $k_3 = 9$ та обчислює координати точки R_i: $R_1 = (8,8)$, $R_2 = (0,11)$, $R_3 = (5,3)$.

Далі обчислюється R – сума всіх точок R_i: $R = (0,2)$, після чого формується число r:

$$r = 0 \bmod 7 = 0.$$

Оскільки $r=0$, необхідно обрати нові випадкові секретні ключі k_i: $k_1=3$, $k_2=4$, $k_3=12$. Відповідно $R_1=(10,7)$, $R_2=(12,1)$, $R_3=(8,5)$. Тоді $R=(9,6)$, і число $r=9 \bmod 7=2$, $r=2$.

Далі кожний користувач i за допомогою свого особистого ключа d_i, значення k_i, хеш-образу h_i та числа r обчислює свою частину підпису:

$$s_1=3-8\cdot 9\cdot 2 \bmod 17=12,\ s_1=12,$$

$$s_2=4-5\cdot 10\cdot 2 \bmod 17=6,\ s_2=6,$$

$$s_3=12-15\cdot 13\cdot 2 \bmod 17=13,\ s_3=13,$$

після чого генерується підпис s: $s=14$.

Агрегованим підписом є пара чисел $\langle r,s\rangle=\langle 2,14\rangle$.

Перевірка підпису $\langle r,s\rangle=\langle 2,14\rangle$ під набором електронних документів $\{M_1,M_2,M_3\}$ з відповідними хеш-образами $h_1=9$, $h_2=10$, $h_3=13$ здійснюється за допомогою додаткової точки еліптичної кривої

$$Q=9Q_1+10Q_2+13Q_3=(2,9),\ Q=(2,9).$$

Обчислюється точка $\tilde{R}$ еліптичної кривої:

$$sP=(10,6),\ rQ=(8,8),$$

$$\tilde{R}=(9,6).$$

Звідси $\tilde{r}=9 \bmod 7=2$, $\tilde{r}=2$.

Оскільки $\tilde{r}=r$, агрегований підпис під набором електронних документів $\{M_1,M_2,M_3\}$ признається справжнім.

6.3 Спарювання Вейля точок еліптичної кривої

Розглянемо еліптичну криву над скінченним полем $GF(p)$

$$y^2 = x^3 + ax \mod p,$$

де $p = 3 \pmod 4$, $a, b \in GF(p)$, $b \neq 0,$; G – адитивна група точок еліптичної кривої простого порядку n з базовою точкою P, $nP = \mathrm{O}$, O – нескінченно віддалена точка; V – мультиплікативна група простого порядку n з нейтральним елементом **1**.

Білінійним спарюванням точок називається функція

$$e : G \times G \to V,$$

для якої виконуються властивості:

1. $e(P+Q,R) = e(P,R) \cdot e(Q,R)$,
   $e(P,Q+R) = e(P,Q) \cdot e(P,R)$
2. $e(k \cdot P, Q) = e(P,Q)^k$, $e(P, k \cdot Q) = e(P,Q)^k$,
3. $e(k \cdot P, Q) = e(P, k \cdot Q)$
4. $e(k \cdot P, m \cdot Q) = e(P,Q)^{k \cdot m}$,
5. $e(P,P) \neq 1$

Спарювання Вейля $e(P,Q)$ точок P, Q еліптичної кривої задається формулою:

$$e(P,Q) = \frac{F(P,Q+S) \cdot F(Q,-S)}{F(P,S) \cdot F(Q,P-S)} \qquad \forall S \in G,$$

де $F(T,Q)$ – функція Вейля для точок T, Q.

Функцію Вейля для точок T, Q, які мають порядок n, можна обчислить за допомогою рекурсивного алгоритму Міллера:

$$f_{1,T}(Q) = 1 \qquad \forall Q \in G$$

$$f_{i+j,T}(Q) = f_{i,T}(Q) \cdot f_{j,T}(Q) \cdot \left. \frac{l_{i,j}}{v_{i+j}} \right|_Q, \qquad i + j < n,$$

$F(T,Q) = f_{n,T}(Q)$,

де $l_{i,j} = \alpha x + \beta y + \gamma$ – рівняння прямої, яка проходить через точки $i \cdot T$, $j \cdot T$, $v_{i+j} = x - x_R$, $R = (i+j) \cdot T = (x_R, y_R)$.

В криптографії для обчислення спарювання Вейля використається функція спотворення $\phi(x,y) = (-x, y \cdot i)$, яка забезпечує виконання властивостей 1-5:

Спарювання Вейля точок еліптичної кривої обчислюється за правилом: $e(P,Q) \to e(P,\phi(Q))$.

Приклад.

Оберемо загальні параметри:

основне поле – скінченне поле $GF(2383)$;

еліптична крива над основним полем

$$y^2 = x^3 - 3x \mod 2383.$$

Базова точка еліптичної кривої $P = (81,787)$ має простий порядок $n = 149$.

Нехай $Q = 3 \cdot P = (1863,213)$, $R = 5 \cdot P = (1368,1568)$.

Перевіримо виконання властивостей 1-5 спарювання Вейля для точок P, Q, R при використанні функції спотворення $\phi(x,y) = (-x, y \cdot i)$.

Нечай $S = \mathrm{O}$ – нескінченно віддалена точка.

Тоді

$e(P+Q,\phi(R)) = 1283 + 1240 \cdot i$

$e(P,\phi(R)) = 1855 + 2008 \cdot i \qquad e(Q,\phi(R)) = 1416 + 364 \cdot i$

$e(P,\phi(R)) \cdot e(Q,\phi(R)) = 1283 + 1240 \cdot i$

$e(P,\phi(Q+R)) = 25 + 976 \cdot i$

$e(P,\phi(Q)) \cdot e(P,\phi(R)) = 25 + 976 \cdot i$

$e(3 \cdot P,\phi(Q)) = 203 + 1502 \cdot i \qquad e(P,\phi(3 \cdot Q)) = 203 + 1502 \cdot i$

$e(P,\phi(Q))^3 = 203 + 1502 \cdot i$

$e(3 \cdot P,\phi(5 \cdot Q)) = 815 + 298 \cdot i \qquad e(P,\phi(Q))^{15} = 815 + 298 \cdot i$

$e(P,\phi(P)) = 716 + 1466 \cdot i$

Надані обчислення демонструють виконання властивостей 1-5.

6.4 Протокол кільцевого підпису електронного документу на еліптичної кривої над простим полем

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(p)$

$$y^2 = x^3 + ax \mod p,$$

де $p = 3 \pmod 4$, $a, b \in GF(p)$, $b \neq 0$, разом із приєднаною нескінченно віддаленою точкою O ; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; H – функція хешування.

Для обчислення спарювання Вейля використається функція спотворення $\phi(x, y) = (-x, y \cdot i)$.

Генерація ключів

Кожний i-ий $(i = 1, 2, ..., t)$ користувач має асиметричну пару ключів: особистий (c_i, d_i) – $1 < c_i, d_i < n$ – та відкритий (U_i, Q_i), $U_i = c_i P$, $Q_i = d_i P$.

Формування цифрового підпису

Нехай одному A_L з членів списку потенційних підписантів A_i, $i = 1, 2, ..., t$, необхідно підписати електронний документ M з хеш-образом $H(M)$. Молодші $|n| - 1$ розряди хеш-образу $H(M)$ формують десяткове число h, яке використається при обчисленні та перевірки цифрового підпису.

Підписант A_L обирає одноразові випадкові числа r, k_i, $1 < r, k_i < n$, $i \neq L$, та обчислює координати точок

$$S_i = k_i P,$$

$$S_L = \frac{1}{h + c_L + d_L \cdot r}\left(P - \sum_{i \neq L} k_i \cdot \left(h \cdot P + U_i + r \cdot Q_i \right) \right).$$

Кільцевим підписом є набір $\langle r, S_1, S_2, ..., S_t \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle r, S_1, S_2, ..., S_t \rangle$ електронного документу M з відповідним його хеш-образу $H(M)$ числом h здійснюється за допомогою відкритих ключів (U_i, Q_i) підписантів A_i, $i = 1, 2, ..., t$, відповідно.

Якщо виконується

$$\prod_{i=1}^{t} e(h \cdot P + U_i + r \cdot Q_i, \phi(S_i)) = e(P, \phi(P))$$

кільцевий цифровий підпис електронного документу M признається справжнім.

Покажемо коректність пропонованого алгоритму формування і перевірки кільцевого підпису:

$$e(h \cdot P + U_L + r \cdot Q_L, \phi(S_L)) =$$

$$= e\left(\left(h + c_L + r \cdot d_L \right) \cdot P, \phi\left(\frac{1}{h + c_L + r \cdot d_L} \cdot \left(P - \sum_{i \neq L} k_i \cdot \left(h \cdot P + U_i + r \cdot Q_i \right) \right) \right) \right) =$$

$$= e\left(P, \phi\left(P - \sum_{i \neq L} k_i \cdot \left(h \cdot P + U_i + r \cdot Q_i \right) \right) \right) =$$

$$= e(P, \phi(P)) \cdot \prod_{i \neq L} e\left(P, \phi\left(k_i \cdot \left(h \cdot P + U_i + r \cdot Q_i \right) \right) \right)^{-1}$$

Звідси

$$\prod_{i=L} e\left(P, \phi\left(k_i \cdot \left(h \cdot P + U_i + r \cdot Q_i\right)\right)\right) = e(P, \phi(P))$$

Приклад.

Оберемо загальні параметри:

основне поле – скінченне поле $GF(2383)$;

еліптична крива над основним полем $y^2 = x^3 + -3x \bmod 2383$.

Базова точка еліптичної кривої $P = (81,787)$ має порядок $n = 149$

Нехай число користувачів в групі дорівнює $t = 3$: A_1, A_2, A_3.

Відповідними особистими ключами є $(c_1, d_1) = (4,5)$, $(c_2, d_2) = (2,1)$, $(c_3, d_3) = (6,3)$.

Тоді відкритими ключами є
$(U_1, Q_1) = ((213,1462),(1368,1568))$,
$(U_2, Q_2) = ((1602,1137),(81,787))$,
$(U_3, Q_3) = ((14,1046),(1863,213))$.

Нехай підписантові A_1 необхідно підписати електронний документ M з хеш-образом $H(M)$ і відповідним йому числом $h = 4$.

Підписант A_1 обирає одноразові випадкові числа $r = 5$, $k_2 = 3$, $k_3 = 5$ та обчислює координати точок S_2, S_3, та S_1:

$S_2 = k_2 P = (1863,213)$, $S_3 = k_3 P = (1368,1568)$,

$$S_1 = \frac{1}{h + c_1 + d_1 \cdot r}\left(P - k_2 \cdot \left(h \cdot P + U_2 + r \cdot Q_2\right) - k_3 \cdot \left(h \cdot P + U_3 + r \cdot Q_3\right)\right) =$$

$$= \frac{1}{28}\left(P - \left(3 \cdot \left(4 \cdot P + U_2 + 5 \cdot Q_2\right) + 5 \cdot \left(4 \cdot P + U_3 + 5 \cdot Q_3\right)\right)\right) =$$

$$= \frac{1}{28}\left((81,787) - \left((1902,214) + (516,1993)\right)\right) =$$

$$= \frac{1}{28}\left((81,787) - (1940,2215)\right) = \frac{1}{28}\left((81,787) + (1940,168)\right) = (740,521)$$

.

Кільцевим підписом є набір
$\langle r, S_1, S_2, S_3 \rangle = \langle 5, (740,521), (1863,213), (1368,1568) \rangle$.
Перевірка підпису

$$\langle r, S_1, S_2, S_3 \rangle = \langle 5, (740,521), (1863,213), (1368,1568) \rangle$$

електронного документу M з відповідним його хеш-образу $H(M)$ числом h здійснюється за допомогою відкритих ключів $(\left(U_1, Q_1\right) = \left((213,1462), (1368,1568)\right)$,
$\left(U_2, Q_2\right) = \left((1602,1137), (81,787)\right)$,
$\left(U_3, Q_3\right) = \left((14,1046), (1863,213)\right)$, підписантів A_1, A_2, A_3 відповідно.

Обчислимо

$$e(h \cdot P + U_1 + r \cdot Q_1, \phi(S_1)) = 25 + 1407i$$
$$e(h \cdot P + U_2 + r \cdot Q_2, \phi(S_2)) = 2312 + 2028i$$
$$e(h \cdot P + U_3 + r \cdot Q_3, \phi(S_3)) = 467 + 1168i$$
$$e(h \cdot P + U_3 + r \cdot Q_3, \phi(S_3)) \times e(h \cdot P + U_3 + r \cdot Q_3, \phi(S_3)) \times$$
$$\times e(h \cdot P + U_3 + r \cdot Q_3, \phi(S_3)) = 716 + 1466i$$

та
$e(P, \phi(P)) = 716 + 1466i$.

Оскільки перевірочне співвідношення виконується, кільцевий цифровий підпис електронного документу M признається справжнім.

6.5 Протокол сліпого підпису на базі алгоритму ЕльГамаля

Цей протокол є модифікацією алгоритму ЕльГамаля для еліптичних кривих. Користувач В підписує в підписанта А деяке повідомлення m, $0 < m < n$, так щоб А у момент формування підписи не міг ознайомитися із умістом повідомлення m.

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(p)$

$$y^2 = x^3 + ax + b \mod p,$$

де $a, b \in GF(p)$, $b \neq 0$, разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; $H(\bullet)$ – функція хешування.

Генерація ключів

Підписант А має асиметричну пару ключів:
особистий d : $1 < d < n$ та відкритий $Q = d \cdot P$.

Формування цифрового підпису

Підписант А обирає одноразовий випадковий секретний ключ k, $1 < k < n$, обчислює координати точки $E = k \cdot P$ та $H(E)$. Молодші $|n|-1$ розряди хеш-образу $H(E)$ формують десяткове число h_E. Далі підписант А перевіряє умову $h_E \neq 0$ та надає точку E користувачу В. Якщо $h_E = 0$, підписант А обирає інше значення k.

Користувач В перевіряє приналежність точки E еліптичній кривій, обирає випадкове число α, $1 < \alpha < n$, обчислює коорди-

нати точки $R = \alpha \cdot E$ та $H(R)$. Молодші $|n|-1$ розряди хеш-образу $H(R)$ формують десяткове число h_R. Далі користувач В перевіряє умову $h_R \neq 0$ (якщо $h_R = 0$, користувач В обирає інше значення α), обчислює коефіцієнт $\beta = \frac{h_R}{h_E} \bmod n$, осліпляє повідомлення m: $\overline{m} = \frac{\alpha}{\beta} m \bmod n$ та надає $\overline{m}$ підписанту А. Якщо α співпадає з β користувач В має обрати інше значення α.

Підписант А перевіряє умову $\overline{m} \neq 0$, обчислює підпис $\overline{s} = (h_E \cdot d + k \cdot \overline{m}) \bmod n$ та надає його користувачу В.

Користувач В перевіряє сформований підписантом А підпис $\overline{s}$. Якщо $\overline{s}P = h_E \cdot Q + \overline{m} \cdot E$, сліпий цифровий підпис документу $\overline{m}$ признається справжнім.

Далі користувач В обчислює підпис для документа m: $s = \beta \cdot \overline{s} \bmod n$.

Сліпим підписом є пара $\langle R, s \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle R, s \rangle$ під електронним документом m здійснюється за допомогою відкритого ключа Q підписанта А.

Якщо $sP = h_R \cdot Q + m \cdot R$, де h_R – молодші $|n|-1$ розряди хеш-образу $H(R)$, сліпий цифровий підпис документу m признається справжнім.

Покажемо коректність пропонованого алгоритму формування і перевірки сліпого підпису:

$$sP = \beta \cdot \overline{s} \cdot P = \beta \cdot (d \cdot h_E + \overline{m} \cdot k) \cdot P = \beta \cdot h_E \cdot Q + m \cdot \alpha \cdot E =$$
$$= h_R \cdot Q + m \cdot R.$$

Приклад.

Оберемо загальні параметри:

основне поле – скінченне поле $GF(17)$;

еліптична крива над основним полем

$$y^2 = x^3 + 6x + 8 \mod 17.$$

Дана еліптична крива містить 13 точок, тобто будь-яка її точка має порядок $n = 13$, $|n| = 4$.

Базова точка еліптичної кривої $P = (1,7)$.

Нехай користувач В бажає підписати у підписанта А повідомлення $m = 10$.

Нехай підписант А має особистий ключ $d = 8$ та відповідний йому відкритий ключ $Q = (9,3)$.

Підписант А обирає одноразовий випадковий секретний ключ $k = 4$, обчислює координати точки $E = (3,6)$ та

$H(E) = MD5(\text{"(3,6)"}) =$

$= \text{'4A9F50245A33E4429DD7B31E9C1F6240'}.$

Звідси $h_E = 0$. Оскільки $h_E = 0$, підписант А має обрати інше значення k: $k = 5$. Далі підписант А обчислює координати новій точки $E = (9,14)$ та

$H(E) = MD5(\text{"(9,14)"}) =$

$= \text{'33BC083E4ED53C83903460C66655BBAD'}.$

Звідси $h_E = 5$ (молодші $|n| - 1 = 3$ розряди хеш-образу $H(E)$ становлять 101).

Підписант А надає точку $E = (9,14)$ користувачу В.

Користувач В перевіряє приналежність точки E еліптичній кривій, обирає випадкове число $\alpha = 9$, обчислює координати точки $R = (16,16)$ та

$H(R) = MD5(\text{"(16,16)"}) =$

$= \text{'C6F039988A718433AB1D5367484E9453'}.$

Звідси $h_R = 3$.

Далі користувач В обчислює коефіцієнт $\beta = 11$, осліпляє повідомлення m: $\overline{m} = 7$ та надає $\overline{m}$ підписанту А.

Підписант А обчислює підпис $\overline{s} = 10$ та надає його користувачу В.

Користувач В перевіряє сформований підписантом А підпис $\overline{s}$: обчислює $\overline{s}P = (0,12)$ і $h_E \cdot Q + \overline{m} \cdot E = (1,7) + (3,11) = (0,12)$. Оскільки $\overline{s}P = h_E \cdot Q + \overline{m} \cdot E$, сліпий цифровий підпис документу $\overline{m}$ признається справжнім. Далі користувач В обчислює підпис для документа m: $s = 6$.

Сліпим підписом є пара $\langle R, s \rangle = \langle (16,16), 6 \rangle$.

Перевірка підпису $\langle R, s \rangle = \langle (16,16), 6 \rangle$ під електронним документом $m = 10$ здійснюється за допомогою відкритого ключа $Q = (9,3)$ підписанта А.

Обчислюється хеш-образ $H(R)$ точки R та відповідне число $h_R = 3$. Далі обчислюються $sP = (16,16)$,

$h_R \cdot Q + m \cdot R = (7,6) + (9,3) = (16,16)$.

Оскільки $sP = h_R \cdot Q + m \cdot R$, сліпий цифровий підпис документу m признається справжнім.

6.6 Анонімність схеми сліпого підпису на базі алгоритму ЕльГамаля

У випадку сліпого підпису до критеріїв захищеності схеми підпису додається анонімність – неможливість відстежити за підписаним документом його автора і однозначно їх пов’язати.

Втім, в деяких схемах у підписанта може виявитись можливість порушити анонімність, оскільки в процесі формування остаточного підпису він обмінюється з емітентом документа додатковими параметрами, передбаченими схемою підпису. Якщо підписант збереже ці параметри, пов’язавши їх з конкретним емітентом, а в подальшому зможе отримати доступ до документу із власним підписом у відкритому вигляді, то він зможе спробувати вирахувати його автора за допомогою збережених параметрів.

Обчисливши маскуючи параметри, які використовував емітент, підписант зможе однозначно пов'язати його із документом, що призведе до порушення анонімності.

Наприклад, в схемі електронного голосування це означитиме, що буде відомо, за кого проголосував конкретний виборець.

Приклад перевірки на анонімність схеми сліпого підпису

Нехай загальні параметри підпису: еліптична крива $y^2 = x^3 + 5x + 9 \mod 59$, базова точка еліптичної кривої $P = (0,3)$ з простим порядком $n = 73$, $|n| = 7$. Підписант А має особистий ключ $d = 15$ та відповідний йому відкритий ключ $Q = (34,22)$.

Нехай підписант А отримав електронний документ $m = 5$ з підписом $\langle R, s \rangle = \langle (1,29), 30 \rangle$, згідно з протоколом, що наведено в 6.5.

Обчислимо h_R. Для отримання хеш-образу $H(R)$ підписант А обрав функцію хешування MD5. Значення h_R сформовано зі молодших $|n| - 1 = 6$ розрядів 128-бітного значення функції MD5.

Хеш-функція

MD5("(1,29)") = 3729424dd8272d04deea8aefe33242d6=….11010110.

Звідси $h_R = 010110_2 = 22_{10}$.

Перевіримо приналежність підпису $R = (1,29)$, $s = 30$ підписанту А – $sP = h_R \cdot Q + m \cdot R$:

$$sP = 30 \cdot (0,3) = (12,33),$$
$$h_R \cdot Q + m \cdot R = 22 \cdot (34,22) + 5 \cdot (1,29) = (12,33).$$

Підписант А переконався, що саме він підписав документ m з підписом $\langle R, s \rangle$.

За допомогою бази параметрів обміну з користувачами (наборів даних k, E, h_E, $\overline{m}$, $\overline{s}$) підписант А спробував визначити, якій із користувачів був емітентом документа m.

Для цього він обчислив значення $\beta = \frac{s}{\overline{s}} \bmod n$, $\alpha = \frac{\beta \cdot \overline{m}}{m} \bmod n$ та точки $\alpha \cdot E$ по усіх наборах даних.

Результати обчислень зведені в таблицю 6.1.

Оскільки $\alpha \cdot E = R$ для користувача B_{11}, то саме він надав документ m для підпису.

Таблиця 6.1 – Результати обчислень підписанта А

	k	E	h_E	$\overline{m}$	$\overline{s}$	$\beta = \frac{s}{\overline{s}}$	$\alpha = \frac{\beta \cdot \overline{m}}{m}$	$\alpha \cdot E$
B_1	8	(35,44)	8	60	16	11	59	(24,18)
B_2	24	(46,15)	25	21	3	10	42	(26,30)
B_{11}	**19**	**(17,13)**	**23**	**53**	**38**	**20**	**66**	**(1,29)**
B_2	29	(56,12)	23	19	20	38	13	(20,12)
B_{10}	16	(37,44)	2	54	18	26	18	(30,14)
B_3	57	(37,15)	53	15	44	4	12	(32,30)
B_1	20	(45,33)	8	8	61	34	69	(39,46)
B_5	67	(23,14)	63	60	1	30	68	(12,33)

6.7 Протокол сліпого підпису на базі німецького стандарту EC-GDSA

Цей протокол засновано на алгоритмі EC-GDSA для еліптичних кривих над простим полем.

Користувач В підписує в підписанта А деяке повідомлення M, так щоб А у момент формування підписи не міг ознайомитися із умістом повідомлення M.

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(p)$

$$y^2 = x^3 + ax + b \mod p,$$

де $a, b \in GF(p)$, $b \neq 0$, разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; $H(\bullet)$ – функція хешування.

Генерація ключів

Підписант А має асиметричну пару ключів:

особистий $d : 1 < d < n$ та

відкритий $Q = (d^{-1} \bmod n) \cdot P$.

Формування цифрового підпису

Підписант А після отримання запиту на підписання повідомлення від користувача В обирає одноразовий випадковий секретний ключ k, $1 < k < n$, обчислює координати точки $C' = k \cdot P$ та відправляє точку C' користувачу В.

Користувач В перевіряє приналежність точки C' еліптичній кривій, обирає випадкові числа α, β, γ, $1 < \alpha, \beta, \gamma < n$, обчислює координати точки $C = \alpha \cdot C' + \beta \cdot Q + \gamma \cdot P = (x_C, y_C)$.

Користувач В обчислює першу складову підпису за формулою $r = x_C \bmod n$.

Користувач В обчислює хеш-образ повідомлення M за допомогою обраної хеш-функції. Отримане двійкове число $H(M)$ конвертується в десяткове число H.

Потім обчислюється значення $h = H \bmod n$.

Користувач В маскує першу складову підпису r та відповідне хеш-образу повідомлення M число h за допомогою обраних параметрів α, β, γ за формулами

$$r' = \alpha \cdot (r + h) \bmod n, \quad h' = (r + h)\left(\frac{h}{r} - \gamma\right) \bmod n$$

та відправляє отримані значення r', h' підписанту А.

Підписант А обчислює за допомогою секретного ключа d сліпий підпис

$$s' = (k \cdot r' - h') \cdot d \bmod n$$

та надає його користувачу В.

Користувач В перевіряє сформований підписантом А підпис s'. Для цього він обчислює два параметри

$$u' = \frac{h'}{r'} \bmod n, \qquad v = \frac{s}{r} \bmod n,$$

та точку $\overline{C} = u' \cdot P + v' \cdot Q$.

Якщо $\overline{C} = C'$, підпис s' признається справжнім.

Далі користувач В обчислює підпис для повідомлення M:

$$s = \left(\frac{s'}{r + h} + \beta\right) \cdot r \bmod n.$$

Підписом для повідомлення M є пара $\langle r, s \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle r, s \rangle$ для повідомлення M здійснюється за допомогою відкритого ключа Q підписанта А.

Верифікатор обчислює хеш-образ повідомлення M за допомогою обраної хеш-функції. Отримане двійкове число $H(M)$ конвертує в десяткове число H і обчислює значення $h = H \bmod n$.

Далі верифікатор обчислює два параметри

$$u = \frac{h}{r} \bmod n \qquad \text{та} \qquad v = \frac{s}{r} \bmod n,$$

точку $\tilde{C} = u \cdot P + v \cdot Q = (x_{\tilde{C}}, y_{\tilde{C}})$ та число $\tilde{r} = x_{\tilde{C}} \bmod n$.

Якщо $\tilde{r} = r$, цифровий підпис $\langle r, s \rangle$ для повідомлення M признається справжнім.

Приклад.

Нехай відкритими параметрами є еліптична крива над простим полем $y^2 = x^3 + 5x + 9 \bmod 59$, базова точка кривої $P = (6, 45)$, порядок точки $n = 73$, $|n| = 7$, і для хешування повідомлень обрана хеш-функція SHA-256.

Нехай користувач B бажає отримати сліпий підпис від підписанта A для повідомлення «world».

Нехай секретним ключем підписанта A є число $d = 18$, відкритим відповідно – точка кривої $Q = (33, 48)$.

Підписант A після отримання запиту на підписання повідомлення від користувача B обирає одноразовий випадковий секретний ключ $k = 64$, обчислює точку $C' = 64 \cdot P = (49, 32)$ та відправляє точку C' користувачу B.

Користувач B перевіряє приналежність точки C' еліптичній кривій, обирає випадкові числа $\alpha = 35$, $\beta = 51$, $\gamma = 10$, обчислює координати точки $C = 35 \cdot C' + 51 \cdot Q + 10 \cdot P = (45, 26)$.

Користувач B обчислює першу складову підпису: $r = 45 \bmod 73 = 45$, $r = 45$.

Користувач B для отримання хеш-образу повідомлення «world» використав програму hash.exe.

В якості функції хешування обрана функція SHA-256.

Молодші $|n| - 1 = 6$ розрядів 256-бітного значення функції SHA-256 формують параметр схеми сліпого підпису h: $\ldots a7_{16} = 10100111_2 \to 100111_2 = 39_{10}$, $h = 39$.

Користувач B маскує першу складову підпису r та відповідне хеш-образу повідомлення «world» число h за допомогою обраних параметрів α, β, γ:

$$r' = 35 \cdot (45 + 39) \bmod 73 = 20,$$

$$h' = (45+39)\left(\frac{39}{45} - 10\right) \bmod 73 = 65$$

та відправляє отримані значення $r' = 20$, $h' = 65$ підписанту А.

Підписант А обчислює за допомогою секретного ключа $d = 18$ сліпий підпис $s' = (64 \cdot 20 - 65) \cdot 18 \bmod 73 = 43$, $s' = 43$, та надає його користувачу В. Користувач В перевіряє сформований підписантом А підпис $s' = 43$. Для цього він обчислює два параметри

$$u' = \frac{65}{20} \bmod 73 = 58, \qquad v' = \frac{43}{20} \bmod 73 = 35,$$

та точку $\overline{C} = 58 \cdot P + 35 \cdot Q = (49,32)$.

Оскільки $\overline{C} = C'$, підпис $s' = 43$ признається справжнім.

Далі користувач В обчислює підпис для повідомлення «world»: $s = \left(\frac{43}{45+39} + 51\right) \cdot 45 \bmod 73 = 42$, $s = 42$.

Підписом для повідомлення «world» є пара $\langle r = 45, s = 42 \rangle$.

Перевірка підпису $\langle r = 45, s = 42 \rangle$ для повідомлення «world» здійснюється за допомогою відкритого ключа $Q = (33,48)$ підписанта А.

Верифікатор обчислює хеш-образ повідомлення «world» за допомогою обраної хеш-функції SHA-256. Для цього він використав програму hash.exe. Молодші $|n| - 1 = 6$ розрядів 256-бітного значення функції SHA-256 формують параметр схеми сліпого підпису h: $\ldots a7_{16} = 10100111_2 \to 100111_2 = 39_{10}$, $h = 39$.

Далі верифікатор обчислює два параметри

$$u = \frac{39}{45} \bmod 73 = 69 \qquad \text{та} \qquad v = \frac{42}{45} \bmod 73 = 35,$$

точку $\tilde{C} = 69 \cdot P + 35 \cdot Q = (45,26)$ та число $\tilde{r} = 45 \bmod 73 = 45$.
$\tilde{r} = 45$.

Оскільки $\tilde{r} = r = 45$, цифровий підпис $\langle r = 45, s = 42 \rangle$ для повідомлення «world» признається справжнім.

6.8 Перевірка на анонімність сліпого підпису на базі німецького стандарту EC-GDSA

Підписант А здійснив наосліп декілька підписів для різних користувачів B_i, згідно з протоколом, наведеним в 6.7. Параметри обміну k, r', h', s' з користувачами він зберіг в базі даних.

В подальшому підписанту А надали документ M з його підписом $\langle R, S \rangle$. Щоб переконатись, що саме він підписав цей документ, підписант А здійснив перевірку підпису згідно з протоколом, наведеним в 6.7.

Далі за допомогою бази параметрів обміну з користувачами підписант А спробував визначити, якій зі користувачів був емітентом документа M.

Для кожної сесії i поставлення підпису підписант А обчислив можливі обрані користувачами параметри:

$$\alpha_i = \frac{r_i'}{R + h} \bmod n,$$

$$\beta_i = \frac{S}{R} - \frac{s_i'}{R + h} \bmod n$$

$$\gamma_i = \frac{h}{R} - \frac{h_i'}{R + h} \bmod n,$$

сформував точку еліптичної кривої

$$\tilde{C} = \alpha_i \cdot k_i \cdot P + \beta_i \cdot Q + \gamma_i \cdot P = (x_{\tilde{C}}, y_{\tilde{C}})$$

та обчислив першу складову підпису $\tilde{r} = x_{\tilde{C}} \bmod n$.

Якщо буде знайдено сесія i, для котрий виконується рівність $\tilde{r} = R$, то підписант А зможе визначити, якій зі користувачів був емітентом документа M.

Однак, за допомогою отриманих значень підписанту не вдасться ідентифікувати емітента, оскільки результат обчислення перевірочної точки для кожної сесії поставлення підпису буде однаковим:

$$\tilde{C} = \alpha_i \cdot k_i \cdot P + \beta_i \cdot Q + \gamma_i \cdot P =$$
$$= \left(\frac{r_i'}{R+h} \cdot k_i + \left(\frac{S}{R} - \frac{s_i'}{R+h} \right) \cdot \frac{1}{d} + \frac{h}{R} - \frac{h_i'}{R+h} \right) \cdot P =$$
$$= \left(\frac{k_i \cdot r_i'}{R+h} + \frac{S}{R} \cdot \frac{1}{d} + \left(- \frac{(k_i \cdot r_i' - h_i') \cdot d}{R+h} \right) \cdot \frac{1}{d} + \frac{h}{R} - \frac{h_i'}{R+h} \right) \cdot P =$$
$$= \left(\frac{S}{R} \cdot \frac{1}{d} + \frac{h}{R} \right) \cdot P$$

Таким чином, наведений протокол є захищеним за критерієм анонімності і може використовуватися в ІВК, заснованих на схемі ECGDSA.

6.9 Протокол сліпого підпису на базі корейського стандарту підпису EC-KCDSA

Цей протокол засновано на алгоритмі EC-KCDSA для еліптичних кривих над розширеним полем.

Користувач B підписує в підписанта A деяке повідомлення M, так щоб A у момент формування підписи не міг ознайомитися із умістом повідомлення M.

Загальносистемні параметри

Еліптична крива над скінченним полем $GF(2^m)$

$$y^2 + xy = x^3 + ax^2 + b \mod(2, f(t)),$$

де $a, b \in GF(2^m)$, разом із приєднаною нескінченно віддаленою точкою O; базова точка еліптичної кривої $P \neq \mathrm{O}$ простого порядку n, така що $nP = \mathrm{O}$ і $kP \neq \mathrm{O}$, $0 < k < n$, $|n|$ – число двійкових розрядів в n; $H(\bullet)$ – обрана функція хешування.

Генерація ключів

Підписант А має асиметричну пару ключів:

особистий $d : 1 < d < n$ та

відкритий $Q = (d^{-1} \bmod n) \cdot P$.

Формування цифрового підпису

Користувач В обчислює хеш-образ повідомлення M за допомогою обраної хеш-функції $H = H(M)$ та відправляє запит на підписання повідомлення підписанту А.

Підписант А після отримання запиту на підписання повідомлення від користувача В обирає одноразовий випадковий секретний ключ k, $1 < k < n$, обчислює координати точки $C' = k \cdot P$ та відправляє точку C' користувачу В.

Користувач В перевіряє приналежність точки C' еліптичній кривій, обирає випадкові числа α, β, $1 < \alpha, \beta < n$, обчислює координати точки $C = \alpha \cdot C' + \beta \cdot Q = (x_C, y_C)$.

Користувач В обчислює першу складову підпису за формулою $R = H(x_C)$ та обчислює параметр $W = R \oplus H$, який конвертується в десяткове число w. Користувач В маскує параметр w за допомогою обраного числа α:

$$w' = \frac{w}{\alpha} \bmod n$$

та відправляє отримане значення w' підписанту А.

Підписант А обчислює за допомогою секретного ключа d сліпий підпис

$$s' = (k - w') \cdot d \bmod n$$

та надає його користувачу В.

Користувач В перевіряє сформований підписантом А підпис s'. Для цього він обчислює точку

$$\overline{C} = w' \cdot P + s' \cdot Q.$$

Якщо $\overline{C} = C'$, підпис s' підписанта А признається справжнім.

Далі користувач В обчислює другу частину s сліпого підпису для повідомлення M:

$$s = \alpha \cdot s' + \beta \bmod n .$$

Десяткове число s конвертується в двійкове число S.

Підписом для повідомлення M є пара $\langle R, S \rangle$.

Перевірка цифрового підпису

Перевірка підпису $\langle R, S \rangle$ для повідомлення M здійснюється за допомогою відкритого ключа Q підписанта А.

Верифікатор обчислює хеш-образ повідомлення M за допомогою обраної хеш-функції: $H = H(M)$ і параметр $W = R \oplus H$, який конвертується в десяткове число w. Двійкове число S також конвертується в десяткове число s.

Далі обчислюється точка еліптичної кривої

$$\tilde{C} = w \cdot P + s \cdot Q = (x_{\tilde{C}}, y_{\tilde{C}})$$

та хеш-образ координати $x_{\tilde{C}}$: $\tilde{R} = H(x_{\tilde{C}})$.

Якщо $\tilde{R} = R$, підпис признається справжнім.

Приклад.

Нехай відкритими параметрами алгоритму EC-KCDSA є еліптична крива над розширеним полем $GF(2^5)$

$$y^2 + xy = x^3 + x^2 + t^2 + t^4 \bmod(2, t^5 + t^3 + 1),$$

базова точка кривої $P = (1 + t + t^2, t^4)$, її порядок $n = 17$, $|n| - 1 = 4$, і для хешування повідомлень обрана хеш-функція SHA-256.

Нехай користувач В бажає отримати сліпий підпис від підписанта А для повідомлення «world».

Користувач В хешує повідомлення «world» та отримує відповідне двійкове число $H = 0111$. (Молодші $|n| - 1 = 4$ розряди хеш-образу повідомлення «world» формують двійкове число H.)

Нехай секретним ключем підписанта А є число $d = 10$, відкритим відповідно – точка кривої $Q = (1 + t^2 + t^4, t + t^4)$.

Підписант А після отримання запиту на підписання повідомлення від користувача В обирає одноразовий випадковий секретний ключ $k = 9$, обчислює точку $C' = 9 \cdot P = (1 + t^2 + t^3 + t^4, t + t^3)$ та відправляє точку C' користувачу В.

Користувач В перевіряє приналежність точки C' еліптичній кривій, обирає випадкові числа $\alpha = 14$, $\beta = 7$, обчислює координати точки $C = \alpha \cdot C' + \beta \cdot Q = 14 \cdot C' + 7 \cdot Q = (1, 1 + t^2)$.

Користувач В обчислює першу складову підпису за формулою $R = H(x_C) = 1011$ та обчислює параметр $W = R \oplus H = 1011 \oplus 0111 = 1100$, який конвертується в десяткове число $w = 12$. Користувач В маскує параметр w за допомогою обраного числа α:

$$w' = \frac{w}{\alpha} \bmod n = \frac{12}{14} \bmod 17 = 13$$

та відправляє отримане значення $w' = 13$ підписанту А.

Підписант А обчислює за допомогою секретного ключа $d = 10$ сліпий підпис

$$s' = (k - w') \cdot d \bmod n = (9 - 13) \cdot 10 \bmod 17 = 11$$

та надає його користувачу В.

Користувач В перевіряє сформований підписантом А підпис $s' = 11$. Для цього він обчислює точку

$$\overline{C} = w' \cdot P + s' \cdot Q = 13 \cdot P + 11 \cdot Q = (1 + t^2 + t^3 + t^4, t + t^3).$$

Оскільки $\overline{C} = C'$, підпис $s' = 11$ підписанта А признається справжнім.

Далі користувач В обчислює другу частину s сліпого підпису для повідомлення M:

$$s = \alpha \cdot s' + \beta \bmod n = 14 \cdot 11 + 7 \bmod 17 = 8.$$

Десяткове число $s = 8$ конвертується в двійкове число $S = 1000$.

Підписом для повідомлення M є пара $\langle R = 1011, S = 1000 \rangle$.

Підписане повідомлення має вигляд $\{\text{"world"}, \langle 1011, 1000 \rangle, text\}$.

Для перевірки підписаного абонентом А повідомлення $\{\text{"world"}, \langle 1011, 1000 \rangle, text\}$ використовуються хеш-образ $H = H(M) = 0111$ повідомлення «world», відкрити загальносистемні параметри алгоритму EC-KCDSA та відкритий ключ підписанта, тобто еліптична крива, базова точка P, її порядок n, відкритий ключ абонента А – точка кривої Q.

Абонент В обчислює параметр $W = R \oplus H = 1100$, який конвертується в десяткове число $w = 12$. Абонент В конвертує двійкове число $S = 1000$ в десяткове число $s = 8$ та обчислює точку еліптичної кривої

$\tilde{C} = 6 \cdot P + 13 \cdot Q = (1 + t^2 + t^3 + t^4, t + t^3)$

$C = 12 \cdot P + 8 \cdot Q = (1, 1 + t^2)$ та хеш-образ координати $x_{\tilde{C}}$:

$\tilde{R} = H(x_{\tilde{C}}) = H(1) \to 1011$.

Оскільки $\tilde{R} = R$, підпис признається справжнім.

6.10 Перевірка на анонімність сліпого підпису на базі корейського стандарту підпису EC-KCDSA

Підписант А здійснив наосліп декілька підписів для різних користувачів B_i, згідно з протоколом, наведеним в 6.9. Параметри взаємодії k, w', s' з користувачами він зберіг в базі даних.

В подальшому підписанту А надали документ M з його підписом $\langle R, S \rangle$. Щоб переконатись, що саме він підписав цей документ, підписант А здійснив перевірку підпису згідно з протоколом, наведеним в 6.9. Для цього двійкове число S конвертується в десяткове число s.

Далі за допомогою бази параметрів обміну з користувачами підписант А спробував визначити, якій зі користувачів був емітентом документа M.

Для кожної сесії i поставлення підпису підписант А обчислив можливі обрані користувачами параметри:

$$\alpha_i = \frac{w}{w_i'} \bmod n, \qquad \beta_i = s - \alpha_i \cdot s_i' \bmod n$$

сформував точку еліптичної кривої

$$\overline{C}_i = \alpha_i \cdot k_i \cdot P + \beta_i \cdot Q = (x_{\overline{C}}, y_{\overline{C}})$$

та обчислив першу складову підпису $\tilde{R} = H(x_{\tilde{C}})$.

Якщо буде знайдено сесія i, для котрий виконується рівність $\tilde{R} = R$, то підписант А зможе визначити, якій зі користувачів був емітентом документа M.

Однак, за допомогою отриманих значень підписанту не вдасться ідентифікувати емітента, оскільки результат обчислення перевірочної точки $\overline{C}_i$ для кожної сесії i поставлення підпису буде однаковим:

$$\tilde{C}_i' = \alpha_i \cdot k_i \cdot P + \beta_i \cdot Q = \left(\frac{w}{w_i'} \cdot k_i + \left(s - \frac{w}{w_i'} \cdot s_i' \right) \cdot \frac{1}{d} \right) \cdot P =$$

$$= \left(\frac{w}{w_i'} \cdot k_i + s \cdot \frac{1}{d} - \left(\frac{w}{w_i'} \cdot \left(k_i - w_i' \right) \right) \right) \cdot P = \left(w + s \cdot \frac{1}{d} \right) \cdot P$$

Таким чином, наведений протокол є захищеним за критерієм анонімності і може використовуватися в ІВК.

Контрольні питання

1. Які параметри схеми підпису є загальносистемними ?
2. Як формуються секретні та відкриті ключі в наведених протоколах?

3. Дайте визначення поняття мультипідпису.
4. Опишіть процедуру формування мультипідпису.
5. Опишіть процедуру перевірки мультипідпису.
6. Чи є обмеження по кількості підписувачів у схемах мультипідпису?
7. Дайте визначення поняття агрегованого підпису.
8. Назвіть властивості агрегованого підпису.
9. Чи є обмеження по кількості підписантів у схемах агрегованого підпису?
10. Чім відрізняється агрегований підпис від мультипідпису?
11. Опишіть процедуру генерації ключів у протоколі агрегованого підпису.
12. Опишіть процедуру формування агрегованого підпису.
13. Опишіть процедуру перевірки агрегованого підпису.
14. Дайте визначення поняття кільцевого цифрового підпису.
15. Опишіть процедуру формування кільцевого цифрового підпису.
16. Опишіть процедуру перевірки кільцевого цифрового підпису.
17. Чи є обмеження по кількості підписантів у схемах кільцевого цифрового підпису?
18. Дайте визначення поняття сліпого цифрового підпису.
19. Яке призначення сліпого підпису?
20. Опишіть процедури формування та перевірки сліпого цифрового підпису.
21. Як перевірить приналежність сліпого підпису підписанту А?
22. Чі можливо встановити емітента підписаного наосліп документу?
23. Сформулюйте визначення поняття анонімності сліпого підпису.
24. Опишіть алгоритм перевірки анонімності підпису електронного документу.
25. Опишіть властивості сліпого підпису.
26. У чому складається перевірка коректності роботи протоколу цифрового підпису?
27. Які параметри впливають на криптостійкість підпису?

7 СУЧАСНІ СТАНДАРТИ СИМЕТРИЧНОГО ШИФРУВАННЯ

В розділі розглядаються три чинні національні стандарти шифрування США, Китаю та України FIPS 197 (RIJNDAEL), SM4 та ДСТУ 7624:2014 відповідно.

Алгоритм RIJNDAEL (вимовляється Рендал) створили бельгійські криптографи Vincent Rijmen і Joan Daemen в 1996 році. З 1997 по 2001 цей алгоритм брав участь у конкурсі AES (Advanced Encryption Standard). Метою проведення конкурсу AES був пошук нового стандарту шифрування США. Переможцем став алгоритм RIJNDAEL. Він був прийнятий американським інститутом NIST в якості стандарту AES-FIPS 197 (Federal Information Processing Standard).

Китайський стандарт шифрування SM4 став відомим в 2006 році. Він призначений для захисту бездротових мереж.

Державна служба спеціального зв'язку та захисту інформації України в 2007-2010 рр. успішно провела національний відкритий конкурс, в результаті якого був відзначений алгоритм «Калина», що став, після додаткових досліджень, основою для національного стандарту.

З першого липня 2015 р. в Україні був введений в дію національний криптографічний стандарт блочного симетричного перетворення ДСТУ 7624: 2014 року, що визначає шифр «Калина» та режими його роботи для забезпечення конфіденційності і цілісності.

7.1 Американський стандарт шифрування FIPS 197 (алгоритм RIJNDAEL)

Алгоритм RIJNDAEL – симетричний блоковий шифр. Довжина блоку і довжина ключа можуть бути незалежно встановлені в 128, 192 або 256 біт. N_b – число 32-бітних слів у блоці: $N_b = 4,6,8$; N_k – число 32-бітних слів в ключі: $N_k = 4,6,8$.

Алгоритм складається з 10, 12 або 14 раундів залежно від довжин ключа і блоку. Число раундів визначається формулою $N_r = \max\{N_b, N_k\} + 6$. Плюс додатковий нульовий раунд.

Таким чином, число раундових ключів дорівнює $N_r + 1$. Раундові ключі RoundKeys виробляються блоками по 32 біта з ключа шифрування ChiperKey. Довжина раундового ключа збігається з довжиною блоку.

Проміжні результати перетворень називаються станами State. Стан можна представити у вигляді матриці байтів. Ця матриця має 4 рядки і N_b стовпців.

Алгоритм RIJNDAEL використовує структуру типу SP-мережа (Substitution-Permutation network, підстановочно-перестановочна мережа).

Базовими операціями в алгоритмі RIJNDAEL є перетворення ByteSub (заміна байтів), ShiftRow (зрушення рядків), MixColumn (перемішування стовпців), AddRoundKey (додавання раундового ключа).

Криптоалгоритм RIJNDAEL включає:

нульовий раунд: AddRoundKey(State, RoundKey);

$N_r - 1$ раунд: на кожному раунді застосовуються 4 перетворення – ByteSub(State), ShiftRow(State), MixColumn(State), AddRoundKey(State, RoundKey);

N_r раунд: ByteSub(State), ShiftRow(State), AddRoundKey(State, RoundKey).

У перетворенні AddRoundKey (State, RoundKey) раундовий ключ RoundKey додається до State за допомогою порозрядного XOR.

Перетворення ByteSub(State) являє собою нелінійну заміну байтів, яка виконується з кожним байтом стану незалежно від його розташування. Перетворення ByteSub може бути зведено в таблицю підстановки S–Box (табл. 7.1). Наприклад, байт A2 перетворюється в байт 3A, 17 в F0. Зворотне перетворення InvByteSub проілюстровано в таблиці підстановки InvS-Box (табл. 7.2).

У перетворенні ShiftRow(State) останні 3 рядка State циклічно зсуваються вправо на різне число байтів: рядок 1 зсувається на C1 байт, рядок 2 - на C2 байт, і рядок 3 – на C3 байт.

Таблиця 7.1 – Таблиця підстановки S –Box

x\y	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

Таблиця 7.2 – Таблиця підстановки InvS-Box

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

Значення зрушень C1, C2 і C3 залежать від довжини блоку N_b. Їхні величини наведено в табл. 7.3. У зворотному InvShiftRow(State) останні 3 рядка State циклічно зсуваються відповідно вліво.

Таблиця 7.3 – Величина зрушення для різної довжини блоків

N_b	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

У перетворенні MixColumn(State) елементи State розглядаються як многочлени над полем $GF(2^8)$. Перетворення стовпця (a_1, a_2, a_3, a_4) здійснюється за правилом:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}.$$

Зворотне перетворення InvMixColumn(State) має вигляд:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} '0E' & '0B' & '0D' & '09' \\ '09' & '0E' & '0B' & '0D' \\ '0D' & '09' & '0E' & '0B' \\ '0B' & '0D' & '09' & '0E' \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

Алгоритм вироблення раундових ключів включає побудову розширеного ключа з ключа шифрування ChiperKey і вибір з нього раундових ключів.

Розширений ключ являє собою лінійний масив 4-байтових слів W. Перші N_b слів розширеного ключа формують раундовий ключ для нульового раунду, наступні N_b слів – для першого раунду тощо.

Побудова розширеного ключа.

Перші N_k слів містять ключ шифрування. Кожне наступне слово $W[i]$ обчислюється за допомогою XOR попереднього слова $W[i-1]$ і слова $W[i-N_k]$. Для слів, позиція яких кратна N_k, перед XOR застосовується перетворення до $W[i-1]$, а потім ще додається раундова константа. Перетворення містить циклічний зсув Rotl на один байт вліво байтів в слові $W[i-1]$, потім застосовується процедура SubByte – побайтова заміна байтів ByteSub.

Раундова константа не залежить від N_k і визначається наступним чином:

$Rcon[i] = (RC[i],'00','00','00')$,

де $RC[i] \in GF(2^8)$, $RC[i] = x^i$, $RC[0] = '01' = 1$, $RC[1] = '02' = x$, $RC[2] = '04' = x^2$ …

Розширений ключ повинен завжди бути обчисленим з ключа шифрування і ніколи не вказується безпосередньо.

Немає жодних обмежень на вибір ключа шифрування.

Алгоритм побудови розширеного ключа на псевдо Сі:

```
KeyExpansion(CipherKey,W)
{for(i = 0,i < Nk;i++) W[i] = CipherKey[i];
 for(j = Nk; j < Nb*(Nr + 1); j = j + Nk)
   {
W[j] = W[j - Nk] ⊕ SubByte(Rotl(W[j - 1])) ⊕ Rcon[j / Nk];
    for(i = 1,(i < Nk)and(i + j < Nb*(Nr + 1);i++)
      W[i + j] = W[i + j - Nk] ⊕ W[i + j - 1];
   }
}
```

Приклади обчислень перетворень.

Перетворення ByteSub(State) і InvByteSub(State).

Нехай дано байт $a = '54'$.

Згідно з таблицею 7.1 підстановки S-Box байт '54' перетворюється в байт '20', згідно з таблицею 7.2 оберненої підстановки InvS-Box байт '20' перетворюється в байт '54'.

Перетворення ShiftRow(State) і InvShiftRow(State)

A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

ShiftRow
$\rightarrow$
InvShiftRow
$\leftarrow$

A	E	I	M
F	J	N	B
K	O	C	G
P	D	H	L

Перетворення MixColumn(State)

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{bmatrix} . \begin{bmatrix} '1B' \\ '10' \\ '23' \\ 'C1' \end{bmatrix} = \begin{bmatrix} 'E4' \\ '9F' \\ '15' \\ '87' \end{bmatrix}.$$

$'01' = 1$, $'02' = x$, $'03' = x+1$, $'1B' = x^4 + x^3 + x + 1$, $'10' = x^4$, $'23' = x^5 + x + 1$, $'C1' = x^7 + x^6 + 1$.

$$'02' \cdot '1B' + '03' \cdot '10' + '01' \cdot '23' + '01' \cdot 'C1' =$$
$$= x \cdot (x^4 + x^3 + x + 1) + (x+1) \cdot x^4 + 1 \cdot (x^5 + x + 1) + 1 \cdot (x^7 + x^6 + 1) =$$
$$= x^5 + x^4 + x^2 + x + x^5 + x^4 + x^5 + x + 1 + x^7 + x^6 + 1 =$$
$$= x^7 + x^6 + x^5 + x^2 \bmod (x^8 + x^4 + x^3 + x + 1) = 11100100_2 = 'E4'.$$

Перетворення InvMixColumn(State)

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} '0E' & '0B' & '0D' & '09' \\ '09' & '0E' & '0B' & '0D' \\ '0D' & '09' & '0E' & '0B' \\ '0B' & '0D' & '09' & '0E' \end{bmatrix} . \begin{bmatrix} 'E4' \\ '9F' \\ '15' \\ '87' \end{bmatrix} = \begin{bmatrix} '1B' \\ '10' \\ '23' \\ 'C1' \end{bmatrix}.$$

$$'0E' = x^3 + x^2 + x, \quad '0B' = x^3 + x + 1, \quad '0D' = x^3 + x^2 + 1,$$
$$'09' = x^3 + 1, \; 'E4' = x^7 + x^6 + x^5 + x^2, \; '9F' = x^7 + x^4 + x^3 + x^2 + x + 1,$$
$$'15' = x^4 + x^2 + 1, \; '87' = x^7 + x^2 + x + 1.$$
$$'0E' \cdot 'E4' + '0B' \cdot '9F' + '0D' \cdot '15' + '09' \cdot '87' =$$
$$= (x^3 + x^2 + x) \cdot (x^7 + x^6 + x^5 + x^2) + (x^3 + x + 1) \cdot (x^7 + x^4 + x^3 + x^2 +$$
$$+ x + 1) + (x^3 + x^2 + 1) \cdot (x^4 + x^2 + 1) + (x^3 + 1) \cdot (x^7 + x^2 + x + 1) =$$
$$= x^{10} + x^6 + x^5 + x^4 + x^2 + x + 1 \bmod (x^8 + x^4 + x^3 + x + 1) = 11011_2 = '1B'.$$

7.2 Китайський стандарт шифрування для захисту бездротових мереж SM4

Алгоритм SM4 – симетричний блоковий шифр. Довжина блоку, а також довжина ключа дорівнюють 128 біт. Алгоритм SM4 використовує структуру мережа Фейстеля. Таким чином, алгоритми шифрування та дешифрування співпадають, однак при дешифруванні раундові ключі подаються в зворотному порядку.

Алгоритм шифрування складається з 32 раундів. Відкритий 128-бітний блок даних розбивається на 4 блоки по 32 біта: (X_0, X_1, X_2, X_3). На кожному раунді i, $i = 0,1,...31$, обчислюється новий 32-бітний блок X_{i+4} з використанням відповідного раундового ключа rk_i довжиною 32 біта. Раундові ключі виробляються з ключа шифрування MK.

Процес шифрування блоку (X_0, X_1, X_2, X_3) виконується таким чином:

$$X_{i+4} = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i), \; i = 0,1,...31.$$

Результатом шифрування є блок

$$(Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32}).$$

Базове перетворення $T(\cdot)$ є суперпозицією нелінійної $\tau(\cdot)$ і лінійної $L(\cdot)$ замін: $T(\cdot) = L(\tau(\cdot))$.

Перетворення $\tau(\cdot)$ являє собою нелінійну заміну байтів

$$B = \tau(a_0, a_1, a_2, a_3) = (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3))$$

і здійснюється за допомогою таблиці підстановки Sbox (табл. 7.4). Наприклад, Sbox('ef')= '84', Sbox('a2')= 'e2'.

Таблиця 7.4 – Таблиця підстановки S –Box

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

Лінійне перетворення $L(\cdot)$ використає операцію $<<< t$ циклічного зсуву 32-бітної послідовності на t біт вліво

$$L(B) = B \oplus (B <<< 2) \oplus (B <<< 10) \oplus (B <<< 18) \oplus (B <<< 24)$$

.

Алгоритм вироблення раундових ключів

Ключ шифрування MK (128 біт) розбивається на 4 блоки по 32 біта: $MK = (MK_0, MK_1, MK_2, MK_3)$.

За допомогою заданих констант $FK_0 = \text{'a3b1bac6'}$, $FK_1 = \text{'56aa3350'}$, $FK_2 = \text{'677d9197'}$, $FK_3 = \text{'b27022dc'}$ обчислюються

$$(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$$

і раундові константи CK_i, $i = 0,1,...31$,

$$CK_i = (4i \cdot 7 \bmod 2^8, (4i+1) \cdot 7 \bmod 2^8, (4i+2) \cdot 7 \bmod 2^8, (4i+3) \cdot 7 \bmod 2^8)$$

Процес обчислення раундових ключів r_i здійснюється за формулою:

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i),\ i = 0,1,...31.$$

Базове перетворення $T'(\cdot)$ є суперпозицією нелінійної $\tau(\cdot)$ і лінійної $L'(\cdot)$ замін: $T'(\cdot) = L'(\tau(\cdot))$.

Лінійне перетворення $L'(\cdot)$ задається формулою

$$L'(B) = B \oplus (B <<< 13) \oplus (B <<< 23).$$

Приклад.

Виконуємо нелінійну заміну $\tau(\cdot)$ для

$A = 10011111001100001110011000010011$.

$A = 10011111001100001110011000010011 \rightarrow$

$\rightarrow (10011111, 00110000, 11100110, 00010011) \rightarrow$

$\rightarrow (\text{'9f'}, \text{'30'}, \text{'e6'}, \text{'13'}) \rightarrow \tau \rightarrow (\text{'e3'}, \text{'e4'}, \text{'77'}, \text{'76'}) \rightarrow$

$\rightarrow (11100011, 11100100, 01110111, 01110110) \rightarrow$

$\rightarrow (11100011111001000111011101110110) = \tau(A)$

Виконуємо лінійне перетворення $L'(\cdot)$ для

$B = 10011111001100001110011000010011$.

$B <<< 13 = 00011100110000100111001111100110$.

$B <<< 23 = 00001001110011111001100001110011$.

$$L'(B) = 10001010001111010000110110000110.$$

Обчислімо раундову константу CK_2.

$$CK_2 = (8 \cdot 7 \bmod 2^8, (8+1) \cdot 7 \bmod 2^8, (8+2) \cdot 7 \bmod 2^8, (8+3) \cdot 7 \bmod 2^8) =$$
$$= (56, 63, 70, 77) = ('38', '3f', '46', '4d') = ('383f464d').$$

7.3 Український стандарт шифрування ДСТУ 7624:2014

Новий національний стандарт є ітеративним шифром і підтримує розмір блоку і довжину ключа шифрування 128, 256 і 512 біт (довжина ключа дорівнює розміру блоку або в два рази перевищує його), забезпечуючи нормальний, високий і надвисокий рівень стійкості (зараз це єдиний в світі стандарт блочного шифрування, що підтримує 512-бітові симетричні ключі). Різні варіанти забезпечують гнучкість вибору параметрів для розробників систем криптографічного захисту, що дозволяє отримати як найвищий рівень швидкодії, так і найбільший запас стійкості перетворення.

Кількість ітерацій шифрування залежить від довжини ключа: 10 циклів для 128-бітового, 14 циклів для 256-бітового і 18 циклів для 512-бітового ключа шифрування (див. табл. 7.5).

Таблиця 7.5 – Таблиця відповідності числових характеристик шифру

Розмір блоку l	Довжина ключа k	Кількість ітерацій t	Кількість стовпців c
128	128	10	2
	256	14	
256	256	14	4
	512	18	
512	512	18	8

Алгоритм «Калина» використовує структуру типу SP-мережа.

Проміжні результати перетворень називаються станами. Стан можна представити у вигляді матриці байтів. Ця матриця має 8 рядків і c стовпців. Початковий стан заповнюється байтами відкритого тексту по стовпцях.

Базове перетворення шифрування блоку даних визначено наступним чином:

$$T_{l,k}^{(K)} = \eta_l^{(K_t)} \circ \psi_l \circ \tau_l \circ \pi_l' \circ \prod_{\nu=1}^{t-1} (\kappa_l^{(K_\nu)} \circ \psi_l \circ \tau_l \circ \pi_l') \circ \eta_l^{(K_0)}.$$

Кожна ітерація шифру складається з п'ятьох перетворень :

$\eta_l^{(K_0)}$ – додавання циклового ключа K_0 і матриці внутрішнього стану G за модулем 2^{64}, виконується над стовпцями, менші значущі байти мають менші індекси;

π' – байтова підстановка, виконує заміну кожного елементу $g_{i,j}$ матриці внутрішнього стану $G = g_{i,j}$ на $\pi_{i \bmod 4}\left(g_{i,j}\right)$, де матриці підстановок $\pi_0 - \pi_3$ наведено в таблицях 7.6 – 7.9;

τ_l – циклічний зсув вправо рядків матриці внутрішнього стану $G = g_{i,j}$, кількість елементів зсуву обчислюється за формулою $\delta_i = \left\lfloor \frac{i \cdot l}{512} \right\rfloor$, де i – номер рядка елемента $g_{i,j}$, l – розмір блоку;

ψ_l – лінійне перетворення стовпців матриці внутрішнього стану $G = g_{i,j}$ (множення матриці лінійного перетворення M на матрицю внутрішнього стану G над розширеним полем $GF(2^8)$, кожен елемент $g_{i,j}$ матриці внутрішнього стану G і матриці лінійного перетворення M розглядається як елемент розширеного поля $GF(2^8)$ за модулем примітивного многочлена $m(x) = x^8 + x^4 + x^3 + x^2 + 1$, матриця лінійного перетворення M наведена в табл. 7.10;

$\kappa_l^{(K_\nu)}$ – побітове додавання циклового ключа K_ν і матриці внутрішнього стану G за модулем 2.

Таблиця 7.6 – Таблиця підстановки π_0

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

Таблиця 7.7 – Таблиця підстановки π_1

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

Таблиця 7.8 – Таблиця підстановки π_2

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

Таблиця 7.9 – Таблиця підстановки π_3

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

Таблиця 7.10 – Матриця лінійного перетворення M в десятковому вигляді

$$M := \begin{bmatrix} 1 & 1 & 5 & 1 & 8 & 6 & 7 & 4 \\ 4 & 1 & 1 & 5 & 1 & 8 & 6 & 7 \\ 7 & 4 & 1 & 1 & 5 & 1 & 8 & 6 \\ 6 & 7 & 4 & 1 & 1 & 5 & 1 & 8 \\ 8 & 6 & 7 & 4 & 1 & 1 & 5 & 1 \\ 1 & 8 & 6 & 7 & 4 & 1 & 1 & 5 \\ 5 & 1 & 8 & 6 & 7 & 4 & 1 & 1 \\ 1 & 5 & 1 & 8 & 6 & 7 & 4 & 1 \end{bmatrix}$$

При розшифрування використовуються зворотні перетворення.

Базове перетворення розшифрування блоку даних визначено наступним чином:

$$U_{l,k}^{(K)} = \eta_l^{(-K_t)} \circ \prod_{\nu=t-1}^{1} (\pi'_{-1} \circ \tau_l^{-1} \circ \psi_l^{-1} \circ \kappa_l^{(K_\nu)}) \circ \pi'_{-1} \circ \tau_l^{-1} \circ \psi_l^{-1} \circ \eta_l^{(-K_0)}$$

.

$\eta_l^{(-K_0)}$ – віднімання циклового ключа K_0 від внутрішнього стану за модулем 2^{64}, виконується над стовпцями, менші значущі байти мають менші індекси;

π'_{-1} – обернена байтова підстановка, виконує заміну кожного елементу $g_{i,j}$ матриці внутрішнього стану $G = g_{i,j}$ на $\pi_{i \bmod 4}^{-1}\left(g_{i,j}\right)$, де матриці підстановок $\pi_0^{-1} - \pi_3^{-1}$ наведено в таблицях 7.11 – 7.14;

τ_l^{-1} – циклічний зсув вліво рядків матриці внутрішнього стану $G = g_{i,j}$, кількість елементів зсуву обчислюється за формулою $\delta_i = \left\lfloor \frac{i \cdot l}{512} \right\rfloor$, де i – номер рядка елемента $g_{i,j}$, l – розмір блоку; наприклад, при $l = 128$:

i	0	1	2	3	4	5	6	7
$\delta_i = \left\lfloor \frac{i \cdot 128}{512} \right\rfloor$	0	0	0	0	1	1	1	1

ψ_l^{-1} – обернене лінійне перетворення стовпців матриці внутрішнього стану $G = g_{i,j}$ (множення матриці лінійного перетворення M^{-1} на матрицю внутрішнього стану G над розширеним полем $GF(2^8)$), кожен елемент $g_{i,j}$ матриці внутрішнього стану G і матриці лінійного перетворення M^{-1} розглядається як елемент розширеного поля $GF(2^8)$ за модулем примітивного многочлена $m(x) = x^8 + x^4 + x^3 + x^2 + 1$, матриця лінійного перетворення M^{-1} наведена в табл. 7.15;

$\kappa_l^{(K_\nu)}$ – побітове додавання циклового ключа K_ν і матриці внутрішнього стану G за модулем 2.

Таблиця 7.11 – Таблиця підстановки π_0^{-1}

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

Таблиця 7.12 – Таблиця підстановки π_1^{-1}

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

Таблиця 7.13 – Таблиця підстановки π_2^{-1}

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

Таблиця 7.14 – Таблиця підстановки π_3^{-1}

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

Таблиця 7.15 – Матриця оберненого лінійного перетворення $M^{-1} = MM$ в десятковому вигляді

$$MM := \begin{bmatrix} 173 & 149 & 118 & 168 & 47 & 73 & 215 & 202 \\ 202 & 173 & 149 & 118 & 168 & 47 & 73 & 215 \\ 215 & 202 & 173 & 149 & 118 & 168 & 47 & 73 \\ 73 & 215 & 202 & 173 & 149 & 118 & 168 & 47 \\ 47 & 73 & 215 & 202 & 173 & 149 & 118 & 168 \\ 168 & 47 & 73 & 215 & 202 & 173 & 149 & 118 \\ 118 & 168 & 47 & 73 & 215 & 202 & 173 & 149 \\ 149 & 118 & 168 & 47 & 73 & 215 & 202 & 173 \end{bmatrix}$$

Приклад. Обчислення перетворень при розшифруванні.

Обчислення функції $\kappa_l^{(K_v)}$ побітового додавання за модулем 2, $l = 128$, циклового ключа

$K_1 = "0102030405060708090A0B0C0D0E0F00"$

і матриці внутрішнього стану

$\bar{G} = "726894B4D6DC4DEC8D0A4A767C7B2E1B"$.

$\overline{G}$	
72	8D
68	0A
94	A4
B4	76
D6	7C
DC	7B
4D	2E
EC	1B

$\oplus$

K_1	
01	09
02	0A
03	0B
04	0C
05	0D
06	0E
07	0F
08	00

=

$\overline{G}_4$	
73	84
6A	00
97	41
B0	7A
D3	71
DA	75
4A	21
E4	1B

Обчислення функції ψ^{-1}.

Нехай дана матриця внутрішнього стану $\overline{G}_4$.

В результаті множення матриці M^{-1} оберненого лінійного перетворення (табл. 7.15) на матрицю внутрішнього стану $\overline{G}_4$ над розширеним полем $GF(2^8)$ отримаємо внутрішній стан $\overline{G}_3$.

Обчислення функції τ_l^{-1}.

Нехай дана матриця внутрішнього стану $\overline{G}_3$.

В результаті множення матриці M^{-1} на матрицю внутрішнього стану $\overline{G}_4$ і циклічного зсуву вліво рядків матриці внутрішнього стану $\overline{G}_3$ отримаємо внутрішній стан $\overline{G}_2$:

$\overline{G}_4$	
73	84
6A	00
97	41
B0	7A
D3	71
DA	75
4A	21
E4	1B

ψ^{-1} $\rightarrow$

$\overline{G}_3$	
F5	3F
06	00
66	27
63	09
14	3F
DA	C9
5E	3F
77	70

τ^{-1} $\rightarrow$

$\overline{G}_2$	
F5	3F
06	00
66	27
63	09
3F	14
C9	DA
3F	5E
70	77

Обчислення функції π'_{-1}.

Нехай дана матриця внутрішнього стану $\overline{G}_2$.

Елементи 0 і 4 рядків замінюються за таблицею підстановки π_0^{-1}, 1 і 5 рядків за таблицею підстановки π_1^{-1}, 2 і 6 рядків за таблицею підстановки π_2^{-1}, 3 і 7 рядків за таблицею підстановки π_3^{-1}.

$\overline{G}_2$	
F5	3F
06	00
66	27
63	09
3F	14
C9	DA
3F	5E
70	77

π'_{-1} $\rightarrow$

$\overline{G}_1$	
7B	8C
7B	83
78	90
86	8F
8C	7C
82	81
85	7E
30	8B

Обчислення функції $\eta_l^{(-K_0)}$ віднімання циклового ключа K_0 від внутрішнього стану $\overline{G}_1$ за модулем 2^{64}. При цьому менші значущі байти мають менші індекси.

$$K_0 = \text{"1716151413121110 1F1E1D1C1B1A1918"}$$

$\overline{G}_1$	
7B	8C
7B	83
78	90
86	8F
8C	7C
82	81
85	7E

$\boxed{-}$

K_0	
17	1F
16	1E
15	1D
14	1C
13	1B
12	1A
11	19

=

$\overline{G}_0$	
64	6D
65	65
63	73
72	73
79	61
70	67
74	65

30	8B

10	18

20	73

Внутрішній стан $\overline{G}_0$ відповідає відкритому тексту повідомлення $T = \text{"decrypt messages"}$:

$\overline{G}_0$	
64	6D
65	65
63	73
72	73
79	61
70	67
74	65
20	73

decoding
$\rightarrow$

T	
d	m
e	e
c	s
r	s
y	a
p	g
t	e
	s

Контрольні питання

1. Як знайти мультиплікативно обернений елемент в розширеному полі?
2. Як перетворити шістнадцятеричне число на елемент розширеного поля?
3. Опишіть раундові перетворення алгоритму RIJNDAEL.
4. В якій послідовності виконуються перетворення в алгоритмі RIJNDAEL?
5. Які параметри алгоритму RIJNDAEL впливають на його криптостійкість?
6. Як многочлен перетворити на десяткове число?

7. Як перетворити шістнадцятеричне число на многочлен?
8. Опишіть раундові перетворення алгоритму SM4.
9. В якій послідовності виконуються перетворення в алгоритмі SM4?
10. Які параметри алгоритму SM4 впливають на його криптостійкість?
11. Яке розширене поле використається в алгоритмі ДСТУ 7624:2014?
12. Як перетворити шістнадцятеричне число на елемент розширеного поля?
13. Опишіть ітераційні перетворення алгоритму ДСТУ 7624:2014.
14. В якій послідовності виконуються перетворення в алгоритмі ДСТУ 7624:2014?
15. Які параметри алгоритму ДСТУ 7624:2014 впливають на його криптостійкість?

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Задірака В.К. Комп'ютерні технології криптографічного захисту інформації на спеціальних цифрових носіях: навч. посіб. / Задірака В.К., Кудін А.М., Людвиченко В.О., Олексюк О.С. – Київ–Тернопіль: Підручники і посібники, 2007. 272 с.
2. Горбенко І.Д. Прикладна криптологія / Горбенко І.Д., Горбенко Ю.І. – Харків: Видавництво «Форт», 2012. – 878 с.
3. Горбенко Ю.І. Інфраструктури відкритих ключів. Електронний цифровий підпис. Теорія та практика : монографія. / Горбенко Ю.І. Горбенко І.Д. – Харків : Видавництво «Форт», 2010. – 608 с.
4. Хорошко В.А., Методы и средства защиты информации / Хорошко В.А., Чекатков А.А.. — К.:Изд-во Юниор, 2003. – 504 с.
5. Кузнецов Г.В. Математичні основи криптографії / Г.В. Кузнецов, В.В. Фомичов, С.О. Сушко, Л.Я. Фомичова. – Дніпропетровськ: НГУ, 2006. – 391 с.
6. Сушко С.О. Математичні основи криптоаналізу: навч. посібник для студ. вищ. навч. закл.: рек. МОНУ / С.О. Сушко, Г.В. Кузнецов, Л.Я. Фомичова, А.В. Корабльов. – Дніпропетровськ: НГУ, 2010.-466с.
7. Новожилова М.В. Математичні моделі захисту інформації: Навчальн. посібник / М.В. Новожилова, С.С. Добротворський, Я.В. Здановський. – Харків: ХДТУБА, 2008. – 80 с.
8. Фергюсон Н. Практическая криптография / Фергюсон Н., Шнайер Б. – М.: Диалектика, 2005. – 424 с.
9. Смарт Н. Криптографія. – Москва: Техносфера, 2005. -528 с.
10. Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях. М.: КУДИЦ-ОБРАЗ, 2001 – 386с.
11. Баричев С.Г. Основы современной криптографии / Баричев С.Г., Гончаров В.В. – М.: Горячая линия - Телеком, 2001.
12. Романец Ю.В. Защита информации в компьютерных системах и сетях / Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. ; под ред. В.Ф. Шаньгина. – 2-е изд., перераб. и доп. – М.: Радио и связь, 2001. – 376 с.

13. Введение в криптографию / Под общ. Ред. В.В. Ященко. – 2-е изд., испр. – М.:МЦНМО: «ЧеРо», 1999. – 272с.
14. Нечаев В.И. Элементы криптографии (Основы теории защиты информации): Учеб. пособие для ун-тов и пед. вузов / Под ред. В.А.Садовничего – М.: Высш.шк., 1999. – 109с.
15. Вербіцький О.В. Вступ до криптології. – Львів: ВНТЛ, 1998. – 248.
16. Саломаа А. Криптография с открытым ключом: Пер. с англ. – М.: Мир, 1995. – 318с.
17. Бессалов А.В. Криптосистемы на эллиптических кривых: учеб.пособие / Бессалов А.В., Телиженко А.Б. – К.: ІВЦ "Видавництво «Політехніка»", 2004. – 224 с.
18. Молдовян Н.А. Теоретический минимум и алгоритмы цифровой подписи. СПб: БХВ-Петербург, 2010. – 304 с.
19. Menezes A. Handbook of applied cryptography / Menezes A., van Oorschot P., Vanstone S. – CRC Press, 1997.
20. Козіна Г.Л. Криптопротоколи: схеми цифрового підпису: навч.посіб./ Г.Л. Козіна, М.А. Молдов'ян, Г.В. Неласа. – Запоріжжя: ЗНТУ, 2014. – 170 с.
21. Виноградов И.М. Основы теории чисел. – М.: Наука, 1981. – 168 с.
22. Курош А.Г. Курс высшей алгебры. – М.:Наука, 1971. – 423 с.
23. О. Н. Василенко, «О вычислении спариваний Вейля и Тейта», Тр. по дискр. матем., 10, Физматлит, М., 2007.
24. Ростовцев А.Г. Алгебраические основы криптографии. – СПб.: НПО «Мир и семья, ООО « Интерлайн», 2000. – 354 с.
25. Штанько С.В. Эллиптические кривые в криптографии // Проблемы информационной безопасности. Компьютерные системы. 2003. № 2. С. 65 – 74.
26. Тарабанько А.В. Реализация протокола проверки знания с использованием эллиптических кривых / Тарабанько А.В., Кучеров М.М. // Безопасность информационных технологий. 2001. № 2. С.61 – 63.
27. Shamir A. How to share a secret // Communications of the ACM. – New York, NY, USA: ACM, 1979. – В. 11. - Т. 22. – С. 612 - 613.
28. Фонвизина Д.В. Защита информации в телекоммуникационных сетях / Фонвизина Д.В., Корниенко В.И. – 2012.

http://ir.nmu.org.ua/bitstream/handle/123456789/1829/34.pdf?sequence=3&isAllowed=y
29. GM/T 0002-2012 SM4 Block Cipher Algorithm (English) http://www.codeofchina.com/standard/GMT0002-2012.html.
30. Whitfield Diffie and George Ledin. SMS4 Encryption Algorithm for Wireless Networks.
31. Hai Cheng. Improvements of SM4 Algorithm and Application inEthernet Encryption System Based on FPGA / Hai Cheng, Shuxia Zhai, Lianzhong Fang, Qun Ding and Chunguang Huang. // Journal of Information Hiding and Multimedia Signal Processing (Ubiquitous International). – Volume 5, Number 3, July 2014.
32. SM4 https://ru.wikipedia.org/wiki/SM4. https://en.wikipedia.org/wiki/SM4_(cipher).
33. FIPS PUB 186-4: Digital Signature Standard (DSS), July 2013: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
34. Michael Braun. A Note on Signature Standards, 2007 / Michael Braun and Anton Kargl. // http://eprint.iacr.org/2007/357
35. ISO/IEC 15946-2. Information Technology — Security Techniques – Cryptographic Techniques Based on Elliptic Curves – Part 2: Digital —Signatures, 2002.
36. Rivest R. How to leak a secret / Ronald L. Rivest, Adi Shamir, Yael Tauman // Proceedings of Asiacrypt 2001 – Springer-Verlag, 2001. – V 2248 of LNCS, pp. 552-565.
37. Jing Xu. A Ring Signature Scheme Using Bilinear Pairings / Jing Xu, Zhenfeng Zhang, Dengguo Feng R. // Conference Paper in Lecture Notes in Computer Science. – 2004: https://www.researchgate.net/publication/221239543.
38. Козина Г.Л. Протокол слепой цифровой подписи на основе стандарта ECGDSA / Козина Г.Л., Никулищев Г.И., Молдовян Н.А. // Вопросы защиты информации. – 2014. – №1.– С.40-45.
39. Нікуліщев Г.І. Анонімність як критерій оцінки захищеності протоколів сліпого електронного цифрового підпису / Нікуліщев Г.І., Козина Г.Л. // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2012. – №2.–С.59-65.
40. Hess E. The Digital Signature Scheme ECGDSA / Hess E., Schafheutle M., Serf P. // http://www.teletrust.de/fileadmin/files/oid/ecgdsa_final.pdf.

41. ISO/IEC 14888-3:2006. Information Technology — Security Techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms, 2006.
42. KCDSA Task Force Team. The Korean Certificate-Based Digital Signature Algorithm, August 1988. http://grouper.ieee.org/groups/1363/P1363a/PSSigs.html.
43. Chaum D. Blind signatures for untraceable payments / D. Chaum // Advances in Cryptology, Crypto '82. – Springer-Verlag. – 1983. – P. 199-203.
44. Кочубинский А.И. Алгоритмы вычисления слепой цифровой подписи на основе стандарта ДСТУ 4145-2002 и ГОСТ Р 34.10-2001 / Кочубинский А.И., Фаль А. М. // Кибернетика и системный анализ. – 2012. – №4. –С. 95-101.
45. Boneh D. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps / D. Boneh, C. Gentry, B. Lynn and H. Shacham // Advances in Cryptology EUROCRYPT 2003, May 4-8. – 2003. – P. 416-432.
46. Макаров А.О. Схема пост-квантовой агрегированной подписи на основе теории алгебраического кодирования. Вопросы кибербезопасности. 2019. №2 (30). С. 69-76. doi:10.21681/2311-3456-2019-2-69-76.
47. Yunlei Zhao. Aggregation of Gamma-Signatures and Applications to Bitcoin / Cryptology ePrint Archive, 2018. – https://eprint.iacr.org/2018/414/20180510:203542.
48. Mihir Bellare. Digital Multi Signature Schemes / Mihir Bellare, Gregory Neven // Springer-Verlag Berlin Heidelberg, 2007. – pp. 145–162.
49. Кочубинский А.И. Слепые мультиподписи на основе стандартов ДСТУ 4145-2002 и ГОСТ Р 34.10-2001 / А.И. Кочубинский, Н.А. Молдовян, А. М. Фаль // Reports of the National Academy of Sciences of Ukraine, 2012. – №3. –С. 38-44.
50. Козіна Г.Л. Колективне підписання різних документів нерівноправними учасниками протоколу / Г.Л.Козіна, Л.М.Карпуков, Д.М.Піза, М.А. Молдов'ян //Захист інформації: науково-технічний журнал. – К:ДУІКТ, 2009. – № 3. – С. 74-80.
51. Бондаренко М.Ф. Инфраструктура открытых ключей как основа обеспечения информационной безопасности

национальных, ведомственных и коммерческих систем информационных технологий / М.Ф.Бондаренко, И.Д.Горбенко, С.П.Черных и др. // Радиотехника. – 2002. – №126. – С.5-17.

52. Ростовцев А.Г. Подпись "вслепую" на эллиптической кривой для электронных денег / А.Г. Ростовцев // Проблемы информационной безопасности. Компьютерные системы. – 2000. - № 1. – С. 40-45.
53. Chaum D. Blind signatures for untraceable payments / D. Chaum // Advances in Cryptology, Crypto '82. – Springer-Verlag. – 1983. – P. 199-203.
54. Савчук М.Н. Математические основания асимметрической криптографии // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2003. – №6. – С.140-148.
55. Мытник К.Я. Смарт-карты и информационная безопасность: под редакцией д.т.н. профессора В.Ф. Шаньгина / Мытник К.Я., Панасенко С.П. – М.: ДМК Пресс, 2019. – 516с.

Навчальне видання

КОЗІНА Галина Леонідівна

КРИПТОГРАФІЯ ВІД ІСТОРІЇ ДО СУЧАСНИХ СТАНДАРТІВ

Навчальний посібник

Художник: *Еліна Фенцик*
Комп'ютерний набір: *Козіна Г.Л.*
Комп'ютерна верстка: *Дяченко О.О.*

Підписано до друку 09.07.2020. Формат 60×84/16. Ум. друк. арк. 11,16.
Тираж 100 прим. Зам. № 690.

Національний університет «Запорізька політехніка»
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64
Тел.: (061) 769–82–96, 220–12–14

Свідоцтво суб'єкта видавничої справи ДК № 6952 від 22.10.2019.