Основи теорії чисел і криптографії

Подільність і модульна арифметика

Матеріал, який ми вивчатимемо в цій темі, ґрунтується на понятті подільності. Ділення цілого числа на додатне ціле дає в результаті частку й остачу. Вивчення остач веде до модульної арифметики, яка відіграє важливу роль у математиці й значно використовується в комп'ютерних науках. Ми розглянемо важливі застосування модульної арифметики, такі як генерування псевдовипадкових чисел, призначення комп'ютерної пам'яті для локалізації файлів, побудову контрольних розрядів та шифрування повідомлень.

Ділення

Коли ціле число ділять на інше ненульове ціле, то частка може бути, а може й не бути цілою. Наприклад, 12/4 = 3 – ціле, а 13/4 = 3.25 – ні. Це проводить до такого означення.

Нехай a та b – цілі числа та $a \neq 0$. Говорять, що a *ділить* b, якщо існує таке ціле c, що $b = ac$. Еквівалентне формулювання: a *ділить* b, якщо b/a – ціле. Коли a ділить b, говорять, що a – *фактор* або *дільник* b, і що b *кратне* a. Запис $a \mid b$ означає, що a ділить b. Якщо a **не** ділить b, то використовують запис $a \nmid b$.

Зауваження. Можна записати $a \mid b$ з використанням квантифікації як $\exists c\,(ac = b)$, де предметна область – множина цілих чисел.

Приклад. Нехай n та d – додатні цілі. Скільки додатних цілих не більших n діляться на d? Усі додатні цілі, які діляться на d, можна записати формі dk, де k – додатне ціле. Отже, кількість додатних цілих, які діляться на d і не більших n дорівнює кількості цілих чисел k, для яких $0 < dk \le n$, тобто $0 < k \le n/d$. Отже, є $\lfloor n/d \rfloor$ додатних цілих не більших n, які діляться на d.

У теоремі 1 сформульовано головні властивості подільності цілих чисел.

Теорема 1. Нехай a, b, c – цілі числа, причому $a \neq 0$. Тоді:

($а$) якщо $a \mid b$ і $a \mid c$, то $a \mid (b + c)$;

($б$) якщо $a \mid b$ то $a \mid bc$ для всіх цілих c;

($в$) якщо $a \mid b$ і $b \mid c$, то $a \mid c$.

Наслідок. Якщо a, b, c – цілі числа, де $a \neq 0$, такі, що $a \mid b$ і $a \mid c$, то $a \mid (mb + nc)$ для будь-яких цілих m і n.

Коли ціле число ділять на додатне ціле, то виникають частка й остача.

Теорема 2. Нехай a – ціле число, d – додатне ціле. Тоді існують **єдині** цілі q і r, $0 \le r < d$, такі, що $a = dq + r$.

У рівності, поданій у теоремі 2, d називають *дільником*, a – *діленим*, q – *часткою*, r – *остачею*. Наступний запис використовують для частки й остачі:

$$q = a \textbf{ div } d, \quad r = a \textbf{ mod } d.$$

Зауваження. Зазначимо, що a **div** d та a **mod** d за фіксованого d є функціями на множині цілих чисел. Більше того, коли a – ціле та d – додатне ціле, більше за 1, то

$$a \textbf{ div } d = \lfloor a/d \rfloor \text{ і } a \textbf{ mod } d = a - d\lfloor a/d \rfloor.$$

Приклад. Знайдемо частку й остачу від ділення 101 на 11. Маємо $101 = 11 \cdot 9 + 2$. Отже, частка від ділення 101 на 11 становить $9 = 101$ **div** 11, а остача становить $2 = 101$ **mod** 11.

Приклад. Знайдемо частку й остачу від ділення –11 на 3. Маємо $-11 = 3 \cdot (-4) + 1$. Отже, частка від ділення –11 на 3 становить $-4 = -11$ **div** 3, а остача становить $1 = -11$ **mod** 3.

Зазначимо, що остача не може бути від'ємною, навіть через рівність $-11 = 3 \cdot (-3) - 2$. Справді, $r = -2$ не задовольняє умову $0 \le r < 3$.

Зазначимо також, що ціле число a подільне на ціле число d тоді й тільки тоді, коли при діленні a на d одержимо нульову остачу.

Множину всіх можливих остач при діленні на m позначають як Z_m – це множина всіх цілих невід'ємних чисел, менших ніж m, тобто $Z_m = \{0, 1, 2, \ldots, m-1\}$.

Зауваження. Мови програмування мають один, можливо, два оператори для модульної арифметики. Такий оператор позначають як mod (BASIC, Maple, Mathematica, EXCEL, SQL), % (C, C++, Java, Python), rem (Ada, Lisp) тощо. Потрібно бути уважним під час використання цих операторів, бо для $a < 0$ деякі з них повертають $a - m\lceil a/m \rceil$ замість правильної відповіді a **mod** $m = a - m\lfloor a/m \rfloor$ (див. попереднє зауваження). Також, на відміну від a **mod** m, деякі з цих операторів визначені для $m < 0$, і навіть для $m = 0$.

Модульна арифметика

Почнемо з простого прикладу модульної арифметики. Якщо відрахувати 14 годин від 15 години біжучого дня, то одержимо 5 годину наступного дня: (15 + 14) **mod** 24 = 5. Тут 29 поділено на 24 і записано остачу.

Оскільки часто цікавими є тільки остачі, то використовують спеціальний запис для них. Ми завжди можемо використовувати запис a **mod** m для подання остачі від ділення цілого числа a на додатне ціле m. Зараз ми введемо інший, але співвіднесений запис, який указує, що два цілих числа мають одну й ту саму остачу від ділення їх на додатне ціле m.

Нехай a і b – цілі числа, а m – додатне ціле. Тоді говорять, що a *конгруентне до* b *за модулем* m, якщо m ділить $a - b$. Це записують як $a \equiv b \pmod{m}$. Якщо a та b не конгруентні, то пишуть $a \not\equiv b \pmod{m}$.

Хоча обидва записи $a \equiv b \pmod{m}$ і a **mod** $m = b$ містять «mod», вони репрезентують фундаментально різні концепції. Перший репрезентує відношення на множині цілих чисел, тоді як другий репрезентує функцію. Проте відношення $a \equiv b \pmod{m}$ і функція **mod** m тісно пов'язані – це складає зміст теореми 3.

Теорема 3. Нехай a, b – цілі числа, m – додатне ціле. Для того, щоб $a \equiv b \pmod{m}$, необхідно й достатньо, щоб a **mod** $m = b$ **mod** m.

Приклад.
$21 \equiv 9 \pmod 6$, тому що 6 ділить $21 - 9 = 12$, а $21 \not\equiv 11 \pmod 6$, бо 6 не ділить $21 - 11 = 10$.

Концепцію конгруентності наприкінці вісімнадцятого століття досліджував видатний німецький математик Карл Фрідріх Гаус (1777–1855). Поняття конгруентності відіграє важливу роль у розвитку теорії чисел. У теоремі 4 сформульовано важливий для роботи з конгруенціями результат.

Теорема 4. Нехай m – додатне ціле число. Цілі числа a та b конгруентні за модулем m тоді й тільки тоді, коли існує ціле k таке, що $a = b + km$.

Множину всіх цілих чисел, конгруентних до цілого числа a за модулем m, називають *класом конгруентності* a за модулем m і позначають як $[a]_m$. Відношення конгруентності на множині цілих чисел є відношенням еквівалентності; клас конгруентності $[a]_m$ являє собою клас еквівалентності. Отже, відношення конгруентності здійснює розбиття множини цілих чисел на класи конгруентності. Можна показати, що таки класів (різних) є точно m.

Теорема 5. Нехай m – додатне ціле число. Нехай $a \equiv b \pmod m$ і $c \equiv d \pmod m$. Тоді $a + c \equiv b + d \pmod m$ і $ac \equiv bd \pmod m$.

Наслідок. Нехай m – додатне ціле число і a та b – цілі числа. Тоді

$$(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m$$

і

$$ab \bmod m = (a \bmod m \cdot b \bmod m) \bmod m.$$

Арифметика за модулем m

Нагадаємо, що множину всіх можливих остач при діленні на m позначають як $\mathbf{Z}_m$ – це множина всіх цілих невід'ємних чисел, менших ніж m, тобто $\mathbf{Z}_m = \{0, 1, 2, \ldots, m-1\}$. На цій множині можна означити арифметичні операції додавання $+_m$ і множення $\cdot_m$.

$$a +_m b = (a + b) \bmod m,$$

$$a \cdot_m b = (a \cdot b) \bmod m.$$

Зауваження. У правій частині двох останніх рівнянь – звичайне додавання і, відповідно, множення цілих чисел.

Операції $+_m$ і $\cdot_m$ називають додаванням і множенням за модулем m і, коли ці операції використовують, то говорять про *арифметику за модулем m.*

Приклад. Обчислимо $7 +_{11} 9$ та $7 \cdot_{11} 9$.

Маємо: $7 +_{11} 9 = (7 + 9) \bmod 11 = 16 \bmod 11 = 5$; $7 \cdot_{11} 9 = (7 \cdot 9) \bmod 11 = 63 \bmod 11 = 8$.

Операції $+_m$ і $\cdot_m$ задовольняють багато властивостей звичайного додавання і множення цілих чисел.

Операції $+_m$ і $\cdot_m$ на множині $\mathbf{Z}_m$ задовольняють умові *замкненості*: якщо a і b належать $\mathbf{Z}_m$, то і $a +_m b$ та $a \cdot_m b$ належать $\mathbf{Z}_m$.

Множина $\mathbf{Z}_m$ разом з операцією $+_m$ утворює *абелеву групу*, бо задовольняються такі властивості.

Асоціативність. Якщо a, b і c належать $\mathbf{Z}_m$, то

$$(a +_m b) +_m c = a +_m (b +_m c).$$

Нейтральний елемент. Елемент $0 \in \mathbf{Z}_m$ є нейтральним елементом по додаванню: якщо a належить $\mathbf{Z}_m$, то $a +_m 0 = 0 +_m a = a$.

Обернений елемент. Якщо $a \neq 0$ належить $\mathbf{Z}_m$, то $m - a$ є оберненим до a за модулем m, а 0 є оберненим до самого себе. Отже, $a +_m (m - a) = 0$ і $0 +_m 0 = 0$.

Комутативність. Якщо a і b належать $\mathbf{Z}_m$, то $a +_m b = b +_m a$.

Множина $\mathbf{Z}_m$ разом з двома операціями $+_m$ і $\cdot_m$ утворює *комутативне кільце з одиницею*, бо задовольняються такі властивості.

1. Стосовно операції $+_m$ множина $\mathbf{Z}_m$ утворює абелеву групу.

2. Операції множення і додавання пов'язані *дистрибутивними законами*. Якщо a, b і c належать $\mathbf{Z}_m$, то $a \cdot_m (b +_m c) = (a \cdot_m b) + (a \cdot_m c)$ і $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

3. Для операції множення виконуються такі властивості.

Комутативність: $a \cdot_m b = b \cdot_m a$.

Асоціативність: $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$

Нейтральний елемент (існування одиниці). Елемент $1 \in \mathbf{Z}_m$ є нейтральним елементом по множенню: $a \cdot_m 1 = 1 \cdot_m a = a$.

Зауваження. Коли працюють з множиною $\mathbf{Z}_m$, часто використовують нотації + і $\cdot$ замість $+_m$ і $\cdot_m$, тобто індекс m не пишуть.

Модульне піднесення до степеня

У сучасній криптографії важливим є можливість ефективно обчислити $b^n \bmod m$, де b, n та m – великі цілі числа. Непрактично спочатку обчислювати b^n, а потім знаходити остачу від ділення результату на m, бо b^n – велике число. Замість цього ми розглянемо алгоритм, який використовує двійкове подання показника степеня n.

Перед тим, як подати алгоритм, ми проілюструємо головну ідею. Ми пояснимо, як використати двійковий подання n, а саме $n = (a_{k-1} \ldots a_1 a_0)_2$, для обчислення b^n. Спочатку зазначимо, що

$$b^n = b^{a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2 + a_0} = b^{a_{k-1} \cdot 2^{k-1}} \cdots b^{a_1 \cdot 2} \cdot b^{a_0}.$$

Це показує, що для обчислення b^n нам потрібно лише обчислити значення b, b^2, $\left(b^2\right)^2 = b^4$, $\left(b^4\right)^2 = b^8$, $\left(b^8\right)^2 = b^{16}$, …, b^{2^k}. Ми перемножуємо лише ті з отриманих термів b^{2^j}, для яких $a_j = 1$. Це дасть b^n.

Наприклад, для обчислення 3^{13} спочатку зазначимо, що $13 = (1101)_2$, отже, $3^{13} = 3^8 3^4 3^1$. Після послідовних піднесень до квадрату, одержимо $3^2 = 9$, $3^4 = 9^2 = 81$ і $3^8 = (81)^2 = 6561$. Отже, $3^{13} = 3^8 \cdot 3^4 \cdot 3^1 = 6561 \cdot 81 \cdot 3 = 1\,594\,323$.

У разі знаходження $b^n \bmod m$ для ефективності обчислень після кожного множення потрібно здійснювати редукцію результату за модулем m. А саме, алгоритм послідовно знаходить $b \bmod m$, $b^2 \bmod m$, $b^4 \bmod m$, $b^8 \bmod m$, $b^{16} \bmod m$, …, $b^{2^{k-1}} \bmod m$ і перемножує лише ті терми $b^{2^j} \bmod m$, для котрих $a_j = 1$, знаходячи після кожного множення залишок від ділення результату на m. Псевдокод для цього алгоритму подано нижче.

Алгоритм 4.1. Модульне піднесення до степеня.

procedure *modexp*(b: integer, $n = (a_{k-1}a_{k-2}\ldots a_1 a_0)_2$, m: positive integer)
$x := 1$
$power := b \bmod m$
for $i := 0$ to $k-1$
    if $a_i = 1$ then $x := (x \cdot power) \bmod m$
    $power := (power \cdot power) \bmod m$
return x (x *equals* $b^n \bmod m$)

Описаний алгоритм був відомий ще до нашої ери в Індії. Його іноді називають *бінарним методом*.

Роботу алгоритму 1 проілюстровано наступним прикладом.

Приклад. Використаємо алгоритм 1 для знаходження $3^{644} \bmod 645$.

Знаходимо $(644)_{10} = (1010000100)_2$.

Послідовно обчислюємо.

$i = 0$. Оскільки $a_0 = 0$, то $x = 1$ і $power = 3^2 \bmod 645 = 9 \bmod 645 = 9$.

$i = 1$. Оскільки $a_1 = 0$, то $x = 1$ і $power = 9^2 \bmod 645 = 81 \bmod 645 = 81$.

$i = 2$. Оскільки $a_2 = 1$, то $x = (1 \cdot 81) \bmod 645 = 81$ і $power = 81^2 \bmod 645 = 111$.

$i = 3$. Оскільки $a_3 = 0$, то $x = 81$ і $power = 111^2 \bmod 645 = 12321 \bmod 645 = 66$.

$i = 4$. Оскільки $a_4 = 0$, то $x = 81$ і $power = 66^2 \bmod 645 = 4356 \bmod 645 = 486$.

$i = 5$. Оскільки $a_5 = 0$, то $x = 81$ і $power = 486^2 \bmod 645 = 236196 \bmod 645 = 126$.

$i = 6$. Оскільки $a_6 = 0$, то $x = 81$ і $power = 126^2 \bmod 645 = 15876 \bmod 645 = 396$.

$i = 7$. Оскільки $a_7 = 1$, то $x = (81 \cdot 396) \bmod 645 = 471$ і $power = 396^2 \bmod 645 = 81$.

$i = 8$. Оскільки $a_8 = 0$, то $x = 471$ і $power = 81^2 \bmod 645 = 6561 \bmod 645 = 111$.

$i = 9$. Оскільки $a_9 = 1$, то знаходимо $x = (471 \cdot 111) \bmod 645 = 36$.

Отже, за алгоритмом 1 ми одержали результат $3^{644} \bmod 645 = 36$.

Прості числа

Просте число — це додатне ціле число, більше від одиниці, яке має рівно два різних натуральних дільники (лише 1 і саме число). Решту чисел, окрім одиниці, називають *складеними*. Таким чином, всі натуральні числа, більші від одиниці, розбивають на прості й складені. Теорія чисел вивчає властивості простих чисел.

Ось усі прості числа, що не більші 100: 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.

Теорема 6 (основна теорема арифметики). Кожне натуральне число, яке більше одиниці, можна представити як добуток простих чисел, причому, в єдиний спосіб з точністю до порядку множників.

Таким чином, прості числа – це елементарні «будівельні блоки» натуральних чисел.

Представлення натурального числа у вигляді добутку простих називають *розкладом на прості* або *факторизацією* числа. Нині невідомі поліноміальні алгоритми факторизації чисел, хоча й не доведено, що таких алгоритмів не існує (тут мова йде про поліноміальну залежність часу роботи алгоритму від логарифма розміру числа, тобто від кількості його цифр). На припущенні про високу обчислювальну складність задачі факторизації ґрунтується криптосистема RSA.

Приклад. Факторизацію чисел 100, 641, 999 та 1024 подано нижче:

$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2;$

$641 = 641;$

$999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37;$

$1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}.$

Пробне ділення.

Часто важливо показати, що задане ціле число – просте. Наприклад, у криптології великі прості числа використовують для шифрування повідомлень. Один метод перевірки числа на простоту на такій теоремі.

Теорема 7. Якщо n – складене число, то n має простий дільник, який не більший ніж $\sqrt{n}$.

Наслідок. Якщо ціле число m не ділиться на жодне просте число, яке не більше ніж $\sqrt{m}$, то число m – просте.

На цій теоремі засновано метод перевірки цілих чисел на простоту «в лоб», за допомогою алгоритму, відомого як **пробне ділення**.

Приклад. Покажемо, що число 107 – просте. Прості, які не більші ніж $\sqrt{107}$, такі: 2, 3, 5, 7. Тому що жодне з цих чисел не ділить 107, то число 107 – просте.

Цю теорему можна використати й для факторизації складених чисел. Алгоритм тут такий. Починаємо з простого числа 2. Якщо n число складене, то за теоремою 7 простий множник p, який не перевищує $\sqrt{n}$, буде знайдено. Коли простий множник p знайдено, продовжуємо факторизацію n/p. Зазначимо, що n/p не має простих множників, менших p. На наступному кроці, якщо n/p не має простих множників більших або рівних p і менших $\sqrt{n/p}$, то число n/p просте. Інакше, якщо воно має простий множник q, і ми продовжимо факторизацію $n/(pq)$. Процедура завершиться, коли на якомусь кроці одержимо просте число.

Приклад. Знайдемо факторизацію числа 7007. Жодне з простих чисел 2, 3 та 5 не ділить 7007. Проте, 7 ділить 7007, причому $7007/7=1001$. Тепер пробуємо ділити 1001 на прості, починаючи з 7. Маємо одразу $1001/7=143$. Продовжуємо кроки алгоритму, використовуючи пробне ділення 143 на послідовні прості числа, починаючи з 7. Одержимо $143/11=13$. Оскільки 13 – число просте, то алгоритм зупиняється. Результат: $7007=7^2\cdot 11\cdot 13$.

Алгоритм Евкліда.

Нехай a та b – цілі числа, які одночасно не дорівнюють нулю. Найбільше ціле d таке, що $d \mid a$ і $d \mid b$ називають *найбільшим спільним дільником* a та b і позначають як $\gcd(a,b)$.

Для знаходження $\gcd(a,b)$ можна використати факторизацію. Нехай

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n},\ b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n},$$

тут $p_1, p_2, \ldots, p_n$ – прості множники (деякі степені можуть дорівнювати нулю). Тоді

$$\gcd(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)}.$$

Приклад. Оскільки $120=2^3\cdot 3\cdot 5$, $500=2^2\cdot 5^3$, то

$$\gcd(120,500)=2^{\min(3,2)}\cdot 3^{\min(1,0)}\cdot 5^{\min(1,3)}=2^2\cdot 3^0\cdot 5^1=20.$$

Числа a та b називають *взаємно простими*, якщо $\gcd(a,b)=1$.

Числа $a_1, a_2, \ldots, a_n$ називають *попарно взаємно простими*, якщо $\gcd(a_i, a_j) = 1$ для всіх $1 \le i < j \le n$. Наприклад, числа 10, 17 і 21 – попарно взаємно прості, а числа 10, 19 і 24 – ні.

Функцією Ейлера називають функцію φ, визначену на множині додатних цілих чисел, значення якої дорівнює кількості додатних цілих чисел, не більших n, які є взаємно простими з n.

Можна довести, що число n є простим тоді й тільки тоді, коли $\varphi(n) = n - 1$.

Обчислювати найбільший спільний дільник двох цілих чисел виходячи з їхньої факторизації нераціонально. Ефективніше це робити за допомогою алгоритму Евкліда. Цей алгоритм зручно пояснити на прикладі. Знайдемо $\gcd(91, 287)$. Спочатку поділимо більше з цих двох чисел на менше:

$$287 = 91 \cdot 3 + 14 \text{ (перше ділення).}$$

Кожен дільник 91 і 287 є також дільником $287 - 91 \cdot 3 = 14$. Отже, кожний дільник 91 і 14 також є дільником $287 = 91 \cdot 3 + 14$. Отже, найбільший спільний дільник 91 і 287 той самий, що й найбільший спільний дільник 91 і 14. Із цих міркувань випливає, що задачу знаходження $\gcd(91, 287)$ можна спростити до задачі знаходження $\gcd(91, 14)$.

Далі, поділимо 91 на 14:

$$91 = 14 \cdot 6 + 7 \text{ (друге ділення).}$$

Аналогічні міркування приводять до висновку, що $\gcd(91, 14) = \gcd(14, 7)$.

Поділивши 14 на 7, одержимо

$$14 = 7 \cdot 2 \text{ (третє ділення).}$$

Із того, що 7 ділить 14 випливає, що $\gcd(14,7)=7$. Оскільки $\gcd(91,287) = \gcd(91,14) = \gcd(14,7) = 7$, то задачу знаходження $\gcd(91,287)$ розв'язано, бо 7 є останньою ненульовою остачею.

Нижче подано алгоритм Евкліда у вигляді псевдокоду.

Алгоритм 4.2. Алгоритм Евкліда.

```
procedure gcd(a,b : positive integers)
x := a
y := b
while y ≠ 0
      begin
       r := x mod y
       x := y
       y := r
       end
return x {gcd(a,b) is x}
```

В алгоритмі початкові значення для змінних x та y – це a та b відповідно, причому $a \geq b$. На кожній ітерації алгоритму x замінюється на y, а y замінюється на x **mod** y, що являє собою остачу від ділення x на y. Ітерації продовжуються допоки виконується умова $y \neq 0$. Алгоритм зупиняється коли $y = 0$, і значення x у цій точці, яке є останньою ненульовою остачею в цій

процедурі, є найбільшим спільним дільником a та b. Кількість операцій ділення в цьому алгоритмі складає $O(\log b)$.

Найбільші спільні дільники як лінійні комбінації.

Важливою властивістю найбільшого спільного дільника двох додатних цілих чисел a та b є те, що його можна подати як лінійну комбінацію

$$sa + tb,$$

де s та t – цілі. Наприклад, $\gcd(6,14) = 2$, і $2 = (-2)\cdot 6 + 1\cdot 14$. Цей факт констатується в наступній теоремі.

Теорема 8 (теорема Безу). Якщо a та b – додатні цілі числа, то існують такі цілі числа s і t, що $\gcd(a, b) = sa + tb$.

Опишемо на прикладі метод, який дає змогу знайти фактичне подання $\gcd(a,b)$ як лінійної комбінації $sa + tb$. Цей метод ґрунтується на зворотному проходженні кроків алгоритму Евкліда, отже, спочатку потрібно виконати сам алгоритм.

Приклад. Виразити $\gcd(252,198) = 18$ як лінійну комбінацію 252 і 198 з цілими коефіцієнтами.

Спочатку покажемо, що $\gcd(252,198) = 18$. Застосуємо алгоритм Евкліда.

$252 = 1\cdot 198 + 54$ (перше ділення),
$198 = 3\cdot 54 + 36$ (друге ділення),
$54 = 1\cdot 36 + 18$ (третє ділення),
$36 = 2\cdot 18$ (четверте ділення).

Остання ненульова остача 18, отже, $\gcd(252,198) = 18$. Використовуючи передостаннє (третє ділення) ми можемо записати 18 як лінійну комбінацію 54 та 36. Маємо

$$18 = 54 - 1 \cdot 36.$$

Друге ділення показує, що

$$36 = 198 - 3 \cdot 54.$$

Підставимо цей вираз для 36 у попередню рівність, тоді подамо 18 як лінійну комбінацію 54 і 198:

$$18 = 54 - 1 \cdot 36 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198$$

Перше ділення показує, що $54 = 252 - 1 \cdot 198$. Підставляючи цей вираз у попередню рівність дістанемо вираз для $\gcd(252,198) = 18$ у вигляді лінійної комбінації 252 і 198. Остаточно матимемо:

$$18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198.$$

Отже, ми дістали вираз для $\gcd(252,198) = 18$ у вигляді лінійної комбінації $s \cdot 252 + t \cdot 198$, де $s = 4$, $t = -5$.

Наведений спосіб визначення коефіцієнтів s і t є наочним, але не оптимальним, бо вимагає збереження в пам'яті проміжних обчислень алгоритму Евкліда. Розглянемо кращий спосіб, так званий *розширений алгоритм Евкліда*. Цей алгоритм дає змогу отримати вираз для $\gcd(a,b)$ у вигляді лінійної комбінації a та b за один прохід, без використання зворотних кроків.

Лінійні конгруенції.

Конгруенцію $ax \equiv b \pmod m$, де m – додатне ціле число, a та b – цілі, x – змінна, називають *лінійною конгруенцією*. Такі конгруенції виникають у теорії чисел та її застосуваннях.

Як розв'язати лінійну конгруенцію $ax \equiv b \pmod m$, тобто як знайти всі цілі числа x, які задовольняють цю конгруенцію? Метод, що ми його тут розглянемо, використовує ціле число a^{-1} таке, що $a^{-1}a \equiv 1 \pmod m$, якщо таке число існує. Таке ціле a^{-1} називають *оберненим* до a за модулем m. Теорема 9 гарантує, що обернене до a за модулем m існує, якщо a та m взаємно прості.

Теорема 9. Якщо a та m взаємно прості цілі числа і $m > 1$, то обернене до a за модулем m існує. Більше того, воно єдине обернене до a за модулем m. (Це означає, що існує єдине додатне ціле $a^{-1} < m$, обернене до a за модулем m, а кожне інше число, обернене до a за модулем m, буде конгруентним до a^{-1} за модулем m.)

Доведення. Доведемо лише існування. За теоремою 8 (Безу) із $\gcd(a, m) = 1$ випливає існування таких цілих s і t, що $sa + tm = 1$. Звідси випливає, що $sa + tm \equiv 1 \pmod m$. Оскільки $tm \equiv 0 \pmod m$, то $sa \equiv 1 \pmod m$.

Для практичного знаходження a^{-1} можна скористатись розширеним алгоритмом Евкліда.

Приклад. Знайдемо обернене до 3 за модулем 7. Оскільки $\gcd(3, 7) = 1$, то теоремою 9 таке обернене існує. У цьому простому прикладі звичайний алгоритм Евкліда одразу приводить до результату:

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

із останньої рівності маємо

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

Із цього випливає, що коефіцієнти Безу для 3 і 7 становлять –2 і 1 відповідно. Отже, –2 є оберненим до 3 за модулем 7. Зазначимо, що кожне ціле, конгруентне до –2 за модулем 7, є також

оберненим до 3: це числа –9, 5, 12 тощо. Єдиним цілим, оберненим до 3 за модулем 7, про яке йдеться в теоремі 3.9, є 5.

Приклад. Знайдемо обернене до 17 за модулем 3432. За розширеним алгоритмом Евкліда знайдемо $\gcd(3432, 17)$ як лінійну комбінацію 17 і 3432. Під час використання алгоритму Евкліда одержимо такі часки й остачі: $q_1 = 201$, $r_2 = 15$, $q_2 = 1$, $r_3 = 2$, $q_3 = 7$, $r_4 = 1$, $q_4 = 2$, тобто $\gcd(3432, 17) = 1$ (остання ненульова остача), і $17^{-1} \bmod 3432$ існує. Для його знаходження скористаємося рекурентними рівностями розширеного алгоритму Евкліда:

$$\begin{aligned} s_2 &= s_0 - q_1 s_1 = 1 - 201 \cdot 0 = 1 & t_2 &= t_0 - q_1 t_1 = 0 - 201 \cdot 1 = -201 \\ s_3 &= s_1 - q_2 s_2 = 0 - 1 \cdot 1 = -1 & t_3 &= t_1 - q_2 t_2 = 1 - 1 \cdot (-201) = 202 \\ s_4 &= s_2 - q_3 s_3 = 1 - 7 \cdot (-1) = 8 & t_4 &= t_2 - q_3 t_3 = -201 - 7 \cdot 202 = -1615 \end{aligned}$$

Отже, $\gcd(3432, 17) = 1 = 8 \cdot 3432 + (-1615) \cdot 17$. Коефіцієнт Безу при 17 є шуканою відповіддю, він дорівнює (–1615), що є тим самим, що й 1817 за модулем 3432. (В обчисленнях використовують найменше додатне значення оберненого до a за модулем m; таке $a^{-1} < m$, за теоремою 9, єдине. Запам'ятаємо таку, як у цьому прикладі, можливу ситуацію для подальших застосувань.)

Як тільки ми отримали a^{-1}, обернене до a, ми можемо розв'язати конгруенцію $ax \equiv b \pmod m$, помноживши обидві її частини на a^{-1}. Наступний приклад ілюструє ці дії.

Приклад. Знайдемо розв'язки конгруенції $3x \equiv 4 \pmod 7$. Із одного з попередніх прикладів випливає, що 5 є оберненим до 3 за модулем 7. Помноживши обидві частини конгруенції на 5 дістанемо

$$5 \cdot 3x \equiv 5 \cdot 4 \pmod 7.$$

Оскільки $15 \equiv 1 \pmod 7$ і $20 \equiv 6 \pmod 7$, то $x \equiv 6 \pmod 7$. Це такі числа: 6, 13, 20, … і -1, -8, -15, …

Китайська теорема про остачі.

Системи лінійних конгруенцій досить широко використовують. Пропонована теорема, яка супроводжується конструктивним доведенням, дає змогу ефективно розв'язувати такі системи.

Теорема 10 (китайська теорема про остачі). Нехай $m_1, m_2, \ldots m_n$ – попарно взаємно прості додатні цілі числа, більші від 1, і $a_1, a_2, \ldots a_n$ – довільні цілі. Тоді система

$$x \equiv a_1 \pmod{m_1},$$
$$x \equiv a_2 \pmod{m_2},$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$x \equiv a_n \pmod{m_n}$$

має єдиний розв'язок за модулем $m = m_1 m_2 \cdots m_n$. (Це слід розуміти так, що існує розв'язок x, $0 \le x < m$, а всі інші розв'язки конгруентні до цього розв'язку за модулем m.)

Доведення. Тут ми доведемо лише існування розв'язку, причому доведення конструктивне: побудуємо алгоритм конструювання цього розв'язку.

Нехай $\mu_k = m/m_k$ для $k = 1, 2, \ldots, n$. Отже, μ_k – добуток усіх модулів за виключенням m_k. Оскільки m_i та m_k не мають спільних множників більших від 1, для $i \neq k$, то $\gcd(m_k, \mu_k) = 1$. Отже, за теоремою 3.9 існує ціле y_k, яке є оберненим до μ_k за модулем m_k тобто $y_k = \mu_k^{-1}$, і, отже

$$\mu_k y_k \equiv 1 \pmod{m_k}.$$

Сумісний розв'язок побудуємо як суму

$$x = a_1\mu_1 y_1 + a_2\mu_2 y_2 + \cdots + a_n\mu_n y_n.$$

Тепер покажемо, що x справді є таким розв'язком. Передусім зазначимо, що $\mu_j \equiv 0 \pmod{m_k}$ коли $j \neq k$, бо тоді μ_j містить m_k як співмножник. Тому всі доданки окрім k-го конгруентні до 0 за модулем m_k. У той же час $\mu_k y_k \equiv 1 \pmod{m_k}$, бо $y_k = \mu_k^{-1}$ за модулем m_k. Остаточно

$$x \equiv a_k\mu_k y_k \equiv a_k \pmod{m_k}$$

для $k = 1, 2, \ldots, n$. Отже, доведено, що x – сумісний розв'язок n конгруенцій.

Доведення існування розв'язку в теоремі 3.10 дає загальний метод розв'язування систем лінійних конгруенцій із попарно взаємно простими модулями.

Приклад. Розв'яжемо систему лінійних конгруенцій $x \equiv 1 \pmod 5$, $x \equiv 2 \pmod 6$ і $x \equiv 3 \pmod 7$. Передусім обчислимо $m = 5 \cdot 6 \cdot 7 = 210$, $\mu_1 = m/5 = 42$, $\mu_2 = m/6 = 35$, $\mu_3 = m/7 = 30$. Далі знаходимо обернене до μ_1 за модулем 5: $y_1 = 3$, обернене до μ_2 за модулем 6: $y_2 = 5$ і обернене до μ_3 за модулем 7: $y_3 = 4$. Розв'язок системи

$$x \equiv a_1\mu_1 y_1 + a_2\mu_2 y_2 + a_3\mu_3 y_3 = 1 \cdot 42 \cdot 3 + 2 \cdot 35 \cdot 5 + 3 \cdot 30 \cdot 4 = 836 \equiv 206 \pmod{210}.$$

Мала теорема Ферма

Мала теорема Ферма надзвичайно корисна для обчислення остач за модулем p від великих степенів цілих чисел,

Теорема 3.11 (мала теорема Ферма). Якщо p – просте число, a – ціле, неподільне на p, то

$$a^{p-1} \equiv 1 \pmod{p}.$$

Більше того, для будь-якого цілого a ми маємо

$$a^{p} \equiv a \pmod{p}.$$

Приклад. Знайдемо 7^{222} **mod** 11. Ми можемо використати малу теорему Ферма для швидшого обчислення ніж за алгоритмом 4.1 модульного піднесення до степеня. За малою теоремою Ферма $7^{10} \equiv 1 \pmod{11}$, отже, $\left(7^{10}\right)^{k} \equiv 1 \pmod{11}$ для кожного додатного цілого k. Для того, щоб скористатись виграшем від останньої конгруенції, поділимо показник 222 на 10, тоді $222 = 22 \cdot 10 + 2$. Тоді

$$7^{222} = 7^{22 \cdot 10 + 2} = \left(7^{10}\right)^{22} 7^{2} \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}.$$

Отже, 7^{222} **mod** $11 = 5$.

Цей приклад показує, як можна використати малу теорему Ферма для обчислення a^{n} **mod** p, якщо p – просте число й $p \nmid a$. Поділимо n на $(p-1)$, тоді дістанемо частку q і остачу r, звідки $n = q \cdot (p-1) + r$, де $0 \le r < p-1$. Звідси

$$a^{n} = a^{q(p-1)+r} = \left(a^{p-1}\right)^{q} a^{r} \equiv 1^{q} a^{r} \equiv a^{r} \ (\mathbf{mod}\ p).$$

Первісні корені й дискретні логарифми

Нехай p – просте число. *Первісним коренем за модулем* p називають ціле r із множини $\mathbf{Z}_p = \{0, 1, 2, \ldots, p-1\}$ таке, що кожний ненульовий елемент $\mathbf{Z}_p$ являє собою степінь r.

Приклад. Якщо ми обчислимо степені $2 \in \mathbf{Z}_{11}$ за модулем 11, то дістанемо: $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 5$, $2^5 = 10$, $2^6 = 9$, $2^7 = 7$, $2^8 = 3$, $2^9 = 6$, $2^{10} = 1$. Легко побачити, що кожний ненульовий елемент множини $\mathbf{Z}_{11}$ є степенем 2. Отже, 2 є первісним коренем 11. Якщо ж ми будемо обчислювати степені 3 за модулем 11, то матимемо $3^1 = 3$, $3^2 = 9$, $3^3 = 5$, $3^4 = 4$, $3^5 = 1$; при подальших піднесеннях до степеня ця послідовність буде повторюватись. Тому що не всі елементи $\mathbf{Z}_{11}$ є степенями 3, то доходимо висновку, що 3 не є первісним коренем 11.

Важливий результат теорії чисел полягає в тому, що для кожного простого p існує первісний корінь за модулем p.

Нехай p – просте число, r – первісний корінь за модулем p і a – ціле число в межах від 1 до $p-1$ включно, тобто a – ненульовий елемент $\mathbf{Z}_p$. Відомо, що існує єдиний показник e такий, що $r^e = a$ в $\mathbf{Z}_p$, тобто $r^e \bmod p = a$.

Нехай p – просте число, r – первісний корінь за модулем p і a – ціле число в межах від 1 до $p-1$ включно. Якщо $r^e \bmod p = a$ та $0 \le e \le p-1$, то e називають *дискретним логарифмом числа a за модулем p при основі r* і пишуть $\log_r a = e$ (тут просте число p мається на увазі).

Приклад. Щойно ми обчислили степені 2 за модулем $p=11$, зокрема, $2^8=3$ і $2^4=5$ в $\mathbf{Z}_{11}$. Отже, дискретні логарифми 3 і 5 за модулем 11 при основі 2 є, відповідно, 8 та 4. (Це степені 2, які дорівнюють, відповідно, 3 і 5 в $\mathbf{Z}_{11}$.) Ми пишемо $\mathbf{log}_2 3=8$ і $\mathbf{log}_2 5=4$ (тут модуль 11 мається на увазі і явно не записується).

Задача обчислення дискретного логарифму як вхід має просте число p, первісний корінь r за модулем p і додатне ціле $a\in\mathbf{Z}_p$. Її вихід – дискретний логарифм числа a за модулем p при основі r. Для розв'язування цієї задачі невідомий жодний поліноміальний алгоритм. Тому що ця задача складна для розв'язування, вона відіграє важливу роль у криптографії.

Українська абетка

А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И
0	1	2	3	4	5	6	7	8	9	10
І	Ї	Й	К	Л	М	Н	О	П	Р	С
11	12	13	14	15	16	17	18	19	20	21
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
22	23	24	25	26	27	28	29	30	31	32

Латинська абетка

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

Класична криптографія

Шифри зсуву і афінні шифри

Одним із найбільш ранніх відомих користувачів криптографії був давньоримський імператор Юлій Цезар. Він зашифровував свої повідомлення у спосіб, коли кожна буква заміщується деякою іншою, а саме тою, що знаходиться в алфавіті через три позиції. Стосовно української абетки це означає, що А міняється на Г, Б на Ґ, В на Д, Г на Е і т.д. Останні ж три букви абетки Ь, Ю та Я заміщуються буквами, що знаходяться через три позиції *циклічно*, тобто переходять у А, Б та В, відповідно. Щоб описати цей *шифр Цезаря* математично, спочатку замінимо кожну букву українського алфавіту елементом множини $\mathbf{Z}_{33}$, тобто цілим числом від 0 до 32: кожна буква замінюється своїм порядковим номером, причому нумерація починається з 0. Наприклад, А міняється на 0, Ї – на 12, Я – на 32. Метод шифрування Цезаря можна подати функцією f, яка визначена на множині $\mathbf{Z}_{33}$ і набуває значення із цієї ж множини:

$$f(p) = (p+3) \bmod 33.$$

Під час шифрування повідомлення буква, яка представлена p міняється на букву, представлену $(p+3) \bmod 33$.

Приклад. Зашифруємо шифром Цезаря повідомлення «Я ПРИЇДУ ЗАВТРА». Спочатку замінимо кожну букву її номером, тоді одержимо:

32 19 20 10 12 5 23 9 0 2 22 20 0.

Тепер замінимо кожний із цих номерів p на $(p+3) \bmod 33$, це дасть:

2 22 23 13 15 8 26 12 3 5 25 23 3.

Повертаючись тепер від цифр назад до букв, одержимо зашифроване повідомлення:

«В ТУЙЛЖЦ ЇГДХУГ».

Для одержання оригінального повідомлення із секретного, зашифрованого шифром Цезаря, використовують функцію f^{-1}, обернену до f. Ця функція відображає ціле число p із множини Z_{33} у $f^{-1}(p)=(p-3)\,\mathbf{mod}\,33$. Процес знаходження оригінального повідомлення із зашифрованого називають *дешифруванням*.

Зауваження. У разі шифрування повідомлень, написаних англійською мовою, очевидно використовують функцію $f(p)=(p+3)\,\mathbf{mod}\,26$, а для розшифрування – функцію $f^{-1}(p)=(p-3)\,\mathbf{mod}\,26$. Область визначення та область значень обох функцій – множина $\mathbf{Z}_{26}$.

Шифр Цезаря можна узагальнити в різний спосіб. Наприклад, замість зсуву числового еквівалента кожної букви на 3, можна зсувати числовий еквівалент кожної букви на k, отже

$$f(p)=(p+k)\,\mathbf{mod}\,33.$$

Такий шифр називають *шифром зсуву*. Зазначимо, що для розшифрування тут має бути використана функція $f^{-1}(p)=(p-k)\,\mathbf{mod}\,33$. Тут ціле число k називають *ключем*.

Подальше узагальнення шифру зсуву, яке трохи посилює його стійкість до розкриття, є використання функції $f(p)=(ap+b)\,\mathbf{mod}\,33$, де a та b цілі, які вибирають так, щоб функція f була **бієкцією**. (Для того, щоб функція $f(p)=(ap+b)\,\mathbf{mod}\,33$ була бієкцією, необхідно й достатньо, щоб $\gcd(a,33)=1$). Таке відображення називають *афінним перетворенням*, а відповідний шифр – *афінним шифром*.

Приклад. Якою буквою буде замінена буква Ю, якщо для шифрування використати функцію $f(p)=(7p+3)\ \mathbf{mod}\ 33$? Оскільки 31 репрезентує букву Ю, то використовуючи задану шифрувальну функцію, дістанемо $f(31)=(7\cdot 31+3)\ \mathbf{mod}\ 33=22$. Оскільки 22 репрезентує букву Т, то в зашифрованому повідомленні Ю заміниться на Т.

Тепер покажемо, як розшифрувати повідомлення, яке зашифроване афінним шифром. Припустімо, що $c=(ap+b)\ \mathbf{mod}\ 33$, причому $\gcd(a,33)=1$. Для дешифрування нам потрібно показати, як виразити p через c. Щоб це зробити, розглянемо шифрувальну конгруенцію $c\equiv ap+b\ (\mathrm{mod}\ 33)$ і розв'яжемо її відносно p. Щоб це зробити, спочатку віднімемо b від обох частин конгруенції, тоді матимемо $c-b\equiv ap\ (\mathrm{mod}33)$. Тому що $\gcd(a,33)=1$, існує обернене a^{-1} до a за модулем 33. Помножимо обидві частини останньої конгруенції на a^{-1}, тоді одержимо $a^{-1}(c-b)\equiv a^{-1}ap\ (\mathrm{mod}33)$. Оскільки $a^{-1}a\equiv 1(\mathrm{mod}\ 33)$, то $p\equiv a^{-1}(c-b)\ \mathrm{mod}\ 33$. Це визначає p, бо p належить $\mathbf{Z}_{33}$.

Блокові шифри

Шифри зсуву та афінні шифри міняють кожну букву алфавіту на іншу букву того ж алфавіту. Тому їх називають *моноалфавітними* шифрами. Розкриття таких шифрів успішно здійснюється аналізом частот появи букв у зашифрованому тексті. Зашифрований текст можна зробити більш стійким до розкриття, якщо при шифруванні заміняти **блоки** букв на інші **блоки** букв. Такі шифри називають *блоковими шифрами*.

Розглянемо простий тип блокового шифру, який називають *шифром перестановки*. Як ключ використовуватимемо перестановку σ множини $\{1, 2, \ldots, m\}$ для деякого додатного цілого m – це **ін'єктивна** функція із $\{1, 2, \ldots, m\}$ у цю ж множину. Для шифрування повідомлення ми спочатку розбиваємо його на блоки, кожний з яких містить по m букв. (Пробіли між буквами та знаки пунктуації ігноруються.) Якщо кількість букв у повідомленні не ділиться на m, то останній блок доповнюємо в кінці випадковими буквами.) Блок $p_1 p_2 \ldots p_m$ шифруємо як $c_1 c_2 \ldots c_m = p_{\sigma(1)} p_{\sigma(2)} \ldots p_{\sigma(m)}$. Для розшифрування блоку $c_1 c_2 \ldots c_m$ криптотексту ми переставляємо його букви використовуючи перестановку σ^{-1}, обернену до перестановки σ.

Приклад. Застосуємо шифр перестановки з такою перестановкою σ множини $\{1, 2, 3, 4\}$: $\sigma(1) = 2$, $\sigma(2) = 4$, $\sigma(3) = 1$, $\sigma(4) = 3$. Зашифруємо повідомлення «Я ПРИЇДУ ЗАВТРА». Розбиваємо на блоки по чотири букви, останній блок доповнюємо випадковими буквами:

ЯПРИ ЇДУЗ АВТР АДЛГ.

Застосуємо для кожного блоку перестановку σ й отримаємо:

ПИЯР ДЗЇУ ВРАТ ДГАЛ.

Для дешифрування використаємо обернену перестановку: $\sigma^{-1}(1) = 3$, $\sigma^{-1}(2) = 1$, $\sigma^{-1}(3) = 4$, $\sigma^{-1}(4) = 2$.

Ще один тип блокового шифру – *шифр Віженера*. Відкритий текст і криптотекст записуються в одному й тому ж алфавіті. Для букв x та y цього алфавіту означимо їхню суму $x + y$ як результат додавання номерів цих букв за модулем 26 для англомовного повідомлення і за модулем 33 – для україномовного. Нагадаємо що нумерація букв алфавіту починається з нуля.

Шифр Віженера застосовують до повідомлення, записаного в рядок без пропусків і розділових знаків. Ключем є слово в тому ж алфавіті. Якщо ключ коротший за повідомлення, то його

записують багато разів поспіль, допоки не вийде рядок такої ж довжини. Рядок із розмноженим ключем записують під рядком із повідомленням, і букви, що опинилися одна над одною, додають. Як результат отримують рядок тої ж довжини, який і є криптотекстом.

Для дешифрування потрібно від значень букв коду відняти значення букв ключа і результат щоразу редукувати за модулем 26 чи 33 залежно від алфавіту повідомлення.

Приклад. Шифрування наказу БОРОНІТЬ КОРОЛІВНУ ВІД ВОРОГІВ з ключем КЛЮЧ відбувається так

	Б	О	Р	О	Н	І	Т	Ь	К	О	Р	О	Л	І	В	Н	У	В	І	Д	В	О	Р	О	Г	І	В
+	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю	Ч	К	Л	Ю
	Л	А	О	Ї	Ю	Ц	Р	Ф	Ш	А	О	Ї	Щ	Ц	А	І	Ґ	Н	З	Я	М	А	О	Ї	Н	Ц	А

Результатом шифрування є нижній рядок. Як можна побачити, при використанні шифру Віженера однаковим буквам у відкритому тексті можуть відповідати різні букви у криптотексті. Ця обставина, безперечно, ускладнює частотний криптоаналіз.

Історична довідка. У 20-х роках минулого століття були винайдені роторні шифрувальні пристрої, які вдосконалювались упродовж наступних десятиліть та інтенсивно використовувались під час ІІ світової війни. Прикладом може служити відомий німецький шифр *Enigma*. Роторні системи реалізовували багаторівневу композицію шрифтів Віженера, що давало шифр з дуже великим періодом.

Криптосистеми з відкритим ключем.

Усі класичні шифри, зокрема зсуву, афінні – це криптосистеми з секретним ключем. Відмінна особливість таких криптосистем полягає в тому, що кожний, кому відомий шифрувальний ключ, швидко може знайти ключ дешифрувальний. Отже, знання як зашифровано повідомлення з використанням секретного ключа дає змогу розшифрувати повідомлення, яке було зашифроване за допомогою цього ключа. Тому класичні криптосистеми називають *симетричними*.

Нині широко застосовують криптосистеми з відкритим ключем. Такі системи називають *асиметричними* – для шифрування й розшифрування вони використовують різні ключі. Ключ, що його використовують для шифрування, є відкритим (не секретним) і може бути повідомлений усім бажаючим надіслати секретне повідомлення. Ключ для розшифрування – секретний і зберігається таємно одержувачем шифрованих повідомлень. Навіть знання всього зашифрованого повідомлення й відкритого ключа не дає змоги дешифрувати повідомлення (без знання секретного ключа).

Система шифрування RSA

У 1976 р. дослідники з Массачусетського технологічного інституту Рональд Райвест (Ronald Rivest), Аді Шамір (Adi Shamir) та Леонард Адлеман (Leonard Adleman) запропонували систему шифрування з відкритим ключем, нині відому як **система RSA**, за першими буквами прізвищ її винахідників.

Коротко опишемо цю систему шифрування.

•**1.** Одержувач повідомлень здійснює генерування відкритого ключа (пара чисел n та e) і секретного ключа (число d). Для цього:

- вибирають два простих числа p і q;
- обчислюють першу частину відкритого ключа $n = pq$;
- визначають другу частину відкритого ключа – вибирають невелике непарне число e, яке має бути взаємно простим з числом $(p-1)(q-1)$;
- обчислюють секретний ключ d, який є цілим числом, оберненим до e за модулем $(p-1)(q-1)$: $d = e^{-1}$ **mod**$((p–1)(q–1))$, тобто $de \equiv 1$ (mod$(p–1)(q–1)$); таке обернене існує, бо gcd $(e, (p-1)(q-1)) = 1$.

•**2.** Повідомлення перетворюють у цифрову форму, тобто записують у вигляді послідовності цілих чисел. Щоб це зробити, ми спочатку замінюємо кожну букву повідомлення на двоцифрове число, використовуючи ту саму заміну, що й для шифру зсуву, але з однією відмінністю. А саме, ми включимо початковий нуль для букв від А до З (в українській абетці), отже А заміниться на 00, Б – на 01, …, З – на 09. Після цього ми об’єднуємо ці двоцифрові числа в цифровий рядок. Нарешті, ми розбиваємо цей рядок на рівного розміру **блоки** з $2N$ цифр, де $2N$ – найбільше

додатне число таке, що 3232…32 (для українського алфавіту), чи 2525…25 (для англійського) із $2N$ цифр не перевищує n. У разі необхідності ми доповнюємо повідомлення фіктивним символом так, щоб останній блок мав такий самий розмір, що й інші. Отже, повідомлення подано як послідовність блоків M_1, M_2, …, M_k для якогось цілого k.

Шифрування полягає в трансформації кожного блоку M_i у зашифрований блок C_i. Це робиться з використанням функції

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

Для виконання шифрування ми використовуємо алгоритм швидкого піднесення до степеня в модульній арифметиці (алгоритм 1).

Ми одержуємо зашифроване повідомлення як послідовність блоків цілих чисел і відправляємо її бажаному одержувачу. Тому що криптосистема RSA перетворює блоки букв у блоки букв, вона є **блоковим шифром**.

•**3.** Одержувач розшифровує повідомлення за допомогою секретного ключа d. Це здійснюють для кожного зашифрованого блоку C_i за допомогою функції

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

Зазначимо, що для дешифрування використовують той самий алгоритм швидкого піднесення до степеня в модульній арифметиці (алгоритм 1), що й для шифрування.

Перед тим, як обґрунтувати коректність розглянутої системи шифрування, наведемо елементарний приклад. Для обчислень у модульній арифметиці скористаємося Modular Arithmetic Calculator (http://ptrow.com/perl/calculator.pl).

Приклад.

Генерування ключів.
1.Вибираємо $p = 53$, $q = 67$.
2. $n = pq = 53 \cdot 67 = 3551$.
3. $(p - 1)(q - 1) = 52 \cdot 66 = 3432$, $e = 17$.
4. $d = e^{-1} \bmod (52 \cdot 66) = 17^{-1} \bmod 3432 = 1817$; для обчислення можна скористатись розширеним алгоритмом Евкліда та теоремою Безу або Modular Arithmetic Calculator.

Шифрування повідомлення. Нехай потрібно передати вказівку КУПИ. Спочатку перетворюємо повідомлення в цифрову форму, замінюючи кожну букву її двоцифровим номером в алфавіті (нагадаємо, що нумерація починається з 0: буква А заміниться на 00); отримаємо 14231910. З нашим модулем $n = 3551$ цифрове повідомлення розбивається на блоки по чотири цифри, бо $3232 < 3551 < 323232$:

1423 1910 (тобто $M_1 = 1423,\ M_2 = 1910,\ k = 2$).

Ми шифруємо кожен блок M_i, використовуючи функцію $C = M^e \bmod n$. Обчислення за допомогою алгоритму швидкого піднесення до степеня в модульній арифметиці або за допомогою Modular Arithmetic Calculator дають $1423^{17} \bmod 3551 = 3153,\ 1910^{17} \bmod 3551 = 2335$. Зашифроване повідомлення 3153 2335 (тобто $C_1 = 3153,\ C_2 = 2335$).

Розшифрування повідомлення. Кожний блок шифру C_i розшифровуємо, використовуючи функцію $M = C^d \bmod n$. Обчислення за допомогою алгоритму швидкого піднесення до степеня в модульній арифметиці або за допомогою Modular Arithmetic Calculator дають $3153^{1817} \bmod 3551 = 1423,\ 2335^{1817} \bmod 3551 = 1910$. Ми одержали вихідне повідомлення у цифровій формі:1423 1910. Повертаючись до букв українського алфавіту, одержуємо вихідне повідомлення КУПИ.

Обґрунтування коректності системи RSA.

Нагадаємо, що $de \equiv 1 \pmod{(p-1)(q-1)}$, тому існує ціле j таке, що $de = 1 + j(p-1)(q-1)$. Звідси випливає, що

$$C^d \equiv \left(M^e\right)^d \equiv M^{de} \equiv M^{1+j(p-1)(q-1)} \pmod n.$$

Зазначимо, що $M^{1+j(p-1)(q-1)} = M \cdot \left(M^{p-1}\right)^{j(q-1)} = M \cdot \left(M^{q-1}\right)^{j(p-1)}$.

Подальше обґрунтування зробимо в додатковому припущенні, що

$$\gcd(M, p) = \gcd(M, q) = 1,$$

яке виконується за виключенням рідких випадків.

Тоді за малою теоремою Ферма можемо записати

$$C^d \equiv M \cdot \left(M^{p-1}\right)^{j(q-1)} \equiv M \cdot 1 = M \pmod{p},$$

$$C^d \equiv M \cdot \left(M^{q-1}\right)^{j(p-1)} \equiv M \cdot 1 = M \pmod{q}.$$

Оскільки $\gcd(p,q)=1$, то з китайської теореми про остачі випливає, що $C^d \equiv M \pmod{pq}$.

Чому криптосистема RSA підходить для криптографії з відкритим ключем?

По-перше, можна швидко побудувати відкритий ключ, знайшовши два великих простих числа p і q, кожне з яких має більш ніж 200 цифр, і знайти ціле число e, взаємно просте з $(p-1)(q-1)$. Коли ми знаємо розклад n на множники, тобто, коли ми знаємо p і q, ми можемо швидко знайти d, яке є цілим, оберненим до e за модулем $(p-1)(q-1)$. [Це робиться за допомогою алгоритму Евкліда: знаходять коефіцієнти Безу s і t для e і $(p-1)(q-1)$, тоді s – обернене до e за модулем $(p-1)(q-1)$.]

Знання d дає змогу розшифрувати повідомлення, відправлені за допомогою нашого ключа. Однак, невідомий спосіб розшифрування повідомлень, який не заснований на пошуку факторизації n (тобто розкладу n на прості множники). Факторизація вважається важким завданням, на відміну від знаходження великих простих чисел p і q, **яке може бути зроблено швидко за допомогою імовірнісних методів**. Найбільш відомі (станом на 2018 р.) ефективні методи факторизації вимагають мільярди років для факторизації 400-значних чисел. Отже, якщо p і q 200-розрядний прості числа, то вважається, що повідомлення, зашифровані за допомогою $n = pq$, не можуть бути розшифровані в розумний період часу, за виключенням ситуації, коли прості числа p і q відомі.

Звичайно, коли числа p і q невеликі, як-от у нашому навчальному прикладі, задача факторизації n не є складною і такий шифр неважко розкрити.

Криптографічні протоколи

Досі ми вивчали, як криптографію можна використати, щоб засекретити повідомлення. Проте, є й інші важливі застосування криптографії. Одне з них – криптографічні протоколи, які дають змогу досягти певного рівня безпеки при обміні повідомленнями між сторонами або учасниками протоколу. Під протоколом ми будемо розуміти послідовність узгоджених приписів, згідно з якими відбувається обмін повідомленнями. Зокрема, ми покажемо, як можна використати криптографію, щоб дати змогу двом сторонам обмінюватись секретним ключем через незахищений канал зв'язку. Ми покажемо також, як криптографію можна використати для відправлення підписаних секретних повідомлень таким чином, щоб одержувач міг бути впевненим, що повідомлення прийшло від передбачуваного відправника.

Обмін ключем

Класична симетрична система захисту конфіденційності листування ґрунтується на наявності надійного каналу для обміну секретним ключем. Канал цей може бути набагато повільнішим, ніж канал для обміну повідомленнями, але безумовно він має бути захищеним від посягань суперника. У класиці такий канал реалізовували за допомогою кур'єра.

В асиметричних криптосистемах проблеми пересилання ключа не існує, бо секретний ключ є особистою власністю кожної сторони, а відкритий ключ перебуває у відкритому доступі. Зазначимо однак, що з появою асиметричних криптосистем симетричні системи не вийшли зі вжитку, бо вони є набагато швидкішими. Фактор швидкості шифрування або дешифрування стає визначальним при пересиланні великих обсягів інформації. Проте асиметричні криптосистеми відкривають нові можливості для обміну ключами при використанні криптосистем симетричних.

Наприклад, практичним є пересилання ключа тим же каналом зв'язку, що й звичайних повідомлень, але зашифрованого за допомогою асиметричної криптосистеми. І хоча швидкодія криптосистеми з відкритим ключем нижча, для цієї мети вона достатня, адже ключ буде посилатися значно рідше, ніж звичайні повідомлення.

Нижче наводиться **інше** елегантне розв'язання проблеми, а саме, *протокол експоненціального обміну ключем*. Двоє учасників протоколу – їх за усталеною традицією звуть Аліса і Боб – спілкуються через канал, що ймовірно прослуховується, і тому хочуть домовитися про спільний секретний ключ. Протокол, за яким вони діятимуть, містить такі кроки, де обчислення виконуються в Z_p.

(1) Аліса вибирає велике просте число p і первісний корінь r за модулем p, і **відкрито**, не роблячи з цього жодної таємниці, посилає p і r Бобові.

(2) Аліса вибирає **секретне** число k_1 у межах від 1 до $p-1$ включно, а Боб – **секретне** число k_2 у тих же межах.

(3) Аліса обчислює $r^{k_1} \bmod p$ і **відкрито** посилає це значення Бобові, а Боб обчислює $r^{k_2} \bmod p$ і теж **відкрито** посилає Алісі.

(4) Аліса обчислює число $\left(r^{k_2}\right)^{k_1} \bmod p$.

(5) Боб обчислює число $\left(r^{k_1}\right)^{k_2} \bmod p$.

Як результат – Аліса і Боб обчислюють одне і теж число

$\left(r^{k_2}\right)^{k_1} \bmod p = \left(r^{k_1}\right)^{k_2} \bmod p = r^{k_1 k_2} \bmod p$, яке і приймають у якості **секретного** ключа.

Бачимо, що p, r, $r^{k_1} \bmod p$, $r^{k_2} \bmod p$ – передбачається як відкрита інформація, а k_1, k_2 та спільний ключ $r^{k_1 k_2} \bmod p$ – як інформація секретна. Для видобування секретної інформації з відкритої суперникові потрібно розв'язати конкретну задачу обчислення дискретного логарифму.

Справді, суперникові потрібно знайти k_1 і k_2 із $r^{k_1} \bmod p$ і $r^{k_2} \bmod p$, відповідно. Жодний інший спосіб видобути цю секретну інформацію із відкритої невідомий. У свій час ми наголошували, що задача обчислення дискретного логарифму є практично нерозв'язною, коли числа p і r є достатньо великими. За досяжної нині потужності комп'ютерів ця система вважається незламною, коли p має більше 300 десяткових цифр, а k_1 і k_2 – більше 100 десяткових цифр кожне.

Цифровий підпис

Нині певні фінансові операції мають здійснюватись за короткий період часу, що унеможливлює використання традиційних засобів засвідчення платіжних документів на зразок великої гербової печатки та підпису головного бухгалтера. Але як тоді банкові вберегтися від злодія-інтелектуала, який добре знається і на фінансах, і на електроніці, і може від імені співробітника банку надіслати вимогу перевести гроші на власний підставний рахунок? Тут ми покажемо, як криптографію можна використати для того, щоб особа, яка отримала інформацію, була впевненою, що ця інформація отримана саме від відомої їй людини. Це питання вирішується за допомогою *протоколу цифрового підпису*. Ми розглянемо конкретну реалізацію такого протоколу на базі системи RSA.

Нехай (n, e) – відкритий ключ Аліси, а d – секретний. Аліса може шифрувати повідомлення x, використовуючи *шифрувальну* функцію $E_{(n,e)}(x) = x^e \bmod n$ і може розшифровувати шифроване повідомлення y, використовуючи *дешифрувальну* функцію $D_{(n,e)}(y) = y^d \bmod n$.

Зазначимо, що Аліса бажає надіслати повідомлення так, щоб кожний, хто його отримає, був упевнений, що це повідомлення саме від неї. Так само, як і під час RSA-шифрування, вона переводить букви повідомлення (незашифрованого) у цифрові еквіваленти і розділяє отриманий цифровий рядок на блоки $m_1, m_2, \ldots, m_k$ рівного розміру (розмір блоків визначають точно так, як і при RSA-шифруванні). Після цього вона застосовує свою **дешифрувальну** функцію $D_{(n,\,e)}$ до кожного блоку і дістає $D_{(n,\,e)}(m_i)$, де $i = 1, 2, \ldots, k$. Аліса посилає цей результат усім запланованим адресатам.

Коли будь-який адресат отримує її лист, він застосовує Алісину **шифрувальну** функцію $E_{(n,\,e)}$ до кожного отриманого блока цифр, – це доступно для будь-кого, бо Алісин відкритий ключ (n, e) – доступна інформація. Результат – блок повідомлення, яке пересилалось, бо $E_{(n,\,e)}\left(D_{(n,\,e)}(m_i)\right) = m_i$. Отже, Аліса має змогу надсилати свої листи багатьом адресатам, і, якщо вона діятиме за описаним протоколом, кожний адресат може бути впевненим, що лист прийшов саме від Аліси. Наступний приклад ілюструє цей протокол.

Приклад. Припустімо, що Алісин відкритий ключ системи RSA той самий, що й у попередньому прикладі, тобто $n = 53 \cdot 67 = 3551$ і $e = 17$. Її секретний ключ, як знайдено у попередньому прикладі, $d = 1817$. Нехай Алиса хоче передати повідомлення «ЗУСТРІЧ ВІДМІНЕНО». Що саме вона має послати?

Спершу Аліса перекладе повідомлення у блоки цифр й отримає таку послідовність блоків:

0923 2122 2011 2702 1105 1611 1706 1718.

Далі вона застосує **дешифрувальну** функцію $D_{(3551,\,17)}(y) = y^{1817} \bmod 3551$ до кожного блоку. Використовуючи швидке модульне піднесення до степеня (з використанням комп'ютера) вона знайде, що $0923^{1817} \bmod 3551 = 0445$, $2122^{1817} \bmod 3551 = 1928$, $2011^{1817} \bmod 3551 = 3284$,

$2702^{1817}\ \mathbf{mod}\ 3551 = 0953$, $1105^{1817}\ \mathbf{mod}\ 3551 = 3501$, $1611^{1817}\ \mathbf{mod}\ 3551 = 1465$,
$1706^{1817}\ \mathbf{mod}\ 3551 = 2188$, $1718^{1817}\ \mathbf{mod}\ 3551 = 3042$.

Отже, лист, розділений на блоки, який надішле Аліса, виглядає так:

0445 1928 5284 0953 3501 1465 2188 3042.

Коли її товариші отримають цей лист, вони застосують її (тобто Алісину) **шифрувальну** функцію (яка відкрита) $E_{(3551,17)}(x) = x^{17}\ \mathbf{mod}\ 3551$ до кожного з цих блоків. Коли вони зроблять це, то отримають блоки цифр оригінального листа, який легко можна перекласти українською мовою. Зокрема, $E_{(3551,17)}(0445) = 0445^{17}\ \mathbf{mod}\ 3551 = 0923$, що при перекладі дасть «ЗУ…» і т. д.

Зазначимо, що в протоколі цифрового підпису головну роль відіграють співвідношення

$$D_{(n,e)}\left(E_{(n,e)}(x)\right) = E_{(n,e)}\left(D_{(n,e)}(x)\right) = x.$$

Ці співвідношення зводяться до рівностей

$$\left(x^e\right)^d \bmod n = \left(x^d\right)^e \bmod n = x,$$

і виражають той факт, що шифрувальна функція $E_{(n,e)}$ і де шифрувальна функція $D_{(n,e)}$ є взаємно оберненими.