УДК 004.056.5
DOI DOI https://doi.org/10.32782/2663-5941/2026.2.1/12

Засадна Х.О.
https://orcid.org/0009-0002-4143-7615
Приватний вищий навчальний заклад
«Європейський університет»

Коцун В.І.
https://orcid.org/0000-0003-2363-8157
Приватний вищий навчальний заклад
«Європейський університет»

ГЕНЕРУВАННЯ КЛЮЧІВ В КРИПТОСИСТЕМІ RSA ТА ЇХ ЗАСТОСУВАННЯ

У статті розглядається алгоритм генерування ключів у несиметричній криптосистемі RSA та їх використання. Під час викладання дисциплін «Інформаційна безпека» та «Криптографічний захист інформації» для студентів різних напрямів підготовки ми стикнулися з проблемою доступного пояснення алгоритму генерування ключів. Викладений у друкованій літературі та в мережі Інтернет алгоритм не можна зрозуміти без попереднього пояснення деяких основ шкільного курсу математики. У статті наведено математичні основи несиметричних криптосистем, необхідні для розуміння алгоритму генерування ключів у криптосистемі RSA. Особливу увагу приділено способам розв'язування рівняння Діофанта.

Для автоматизації генерування ключів запропоновано шаблон у табличному редакторі Microsoft Excel.

Багато сучасних електронних сервісів вимагають розуміння протоколів використання двох ключів. У найближчому майбутньому їх матиме і буде змушений використовувати кожен громадянин України для ідентифікації та захисту особистої електронної інформації.

Сьогодні сфер застосування є двоключових криптосистем є досить багато. А з появою криптоіндустрії стало ясно, що криптографічні методи захисту інформації давно і активно інтегруються в наше повсякденне життя.

Вважаємо, що відносно молода наука під назвою «криптологія» повинна сьогодні вивчатися у кожному зво: не обов'язково як окрема дисципліна, можливо як частина дисциплін, що вивчають безпеку інформації та її захист. Школи вже цікавляться криптологією, про це свідчать численні публікації та відеоуроки в мережі Інтернет. Симетричні крипосистеми, які використовувалися до 1976 року є дуже цікавими, вони пояснюють постановки основних задач криптології і містять багато цікавих алгоритмів шифрування та дешифрування інформації, які розвивають логіку та креативність і які можна успішно програмувати.

Ключові слова: *RSA, GCD, алгоритм Евкліда, рівняння Діофанта, ланцюгові дроби, використання несиметричних криптосистем, ЕЦП.*

Постановка проблеми. Пояснити алгоритми роботи несиметричних криптосистем студентам, які не мають глибоких знань зі спеціальних дисциплін також можна. Для цього цілком достатньо елементарних знань зі шкільного курсу алгебри, вміння працювати з Калькулятором будь-якого гаджета, Калькулятором Windows та мати елементарні навики роботи з Microsoft Excel.

Після того, як студент згенерує власні ключі, підготує сертифікат відкритого ключа, зашифрує та дешифрує повідомлення, накладе ЕЦП на невеликий електронний документ і перевірить його, можна буде вже детальніше розповідати йому про послуги Центрів сертифікації ключів, електронне голосування, електронний нотаріат, криптогаманець та криптобіржу, вбудований криптографічний захист інформації у програмному забезпеченні, криптографічний захист інформації у фінансових та державних установах, в Інтернет-сервісах і багато інших застосувань несиметричних криптосистем у його повсякденному житті та майбутній професійній діяльності. За нинішніх

обставин криптографічно захищені реєстри та послуги є практично незламними.

Аналіз останніх досліджень і публікацій. У роботах Вербіцького О.В., Ємця В., Мельника А., Поповича Р. [1, 2] та онлайн-енциклопедії Вікіпедія [3] викладено алгоритм генерування ключів у криптосистемі RSA, цей же алгоритм опублікований в багатьох інших джерелах. Однак без попереднього пояснення деяких математичних термінів та алгоритмів він буде незрозумілим. При розробці несиметричних криптосистем було детально проаналізовано всі недоліки попередніх симетричних (одноключових) криптосистем, враховано пропозиції криптоаналітиків і запропоновано використовувати два ключі. Але найбільше вражає те, що автори криптосистеми RSA використали при її розробці математичні задачі, які не мають однозначних розв'язків або не можуть бути розв'язані обчислювальним шляхом на той час і в найближчому майбутньому.

Постановка завдання. Метою даної публікації є пояснення операцій та рівнянь в алгоритмі генерування ключів в криптосистемі RSA та зв'язок між існуючими способами. Запропоновано реалізацію алгоритму для конкретних даних в середовищі MS Excel. Описані найчастіше використовувані застосування ключів.

Виклад основного матеріалу. Перш ніж описати ключові алгоритми криптосистеми RSA, треба пригадати деякі визначення зі шкільного курсу математики – математичні основи несиметричної криптосистеми RSA.

1. Прості та складені числа.

Прості числа – це натуральні числа, які мають тільки два дільники: одиницю і себе (наприклад, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29…). Складені числа мають більше двох дільників і можуть бути розкладені на добуток простих чисел (наприклад, 4, 6, 8, 9, 10, 12…). Число 1 ні просте, ні складене, бо має лише один дільник. В RSA використовуються прості числа.

2. Функція mod.

Нехай $n \in N$ – довільне натуральне число, $x \in Z$ – довільне ціле число. Через $x \bmod n$ позначають залишок від ділення націло числа x на число n. Залишок від ділення завжди менший за дільник n і одного з ним знаку (тобто він невід'ємний).

$8 \bmod 10 = 8$ ($8=0\times10+2$), $12 \bmod 10 = 2$ ($12=1\times10+2$),

$32 \bmod 10 = 2$ ($32=3\times10+2$), $52 \bmod 10 = 2$ ($52=5\times10+2$),

$-8 \bmod 10 = 2$ ($-8=-1\times10+2$), $-18 \bmod 10 = 2$ ($-18=-2\times10+2$),

$-58 \bmod 10 = 2$ ($-58=-6\times10+2$), $-78 \bmod 10 = 2$ ($-78=-8\times10+2$).

Тобто при $x>0$ від діленого віднімають дільник до тих пір, поки остача не стане меншою за нього, а при $x<0$ до діленого додають дільник до тих пір, поки остача не стане меншою за нього.

Якщо $x \bmod n = y \bmod n$, то кажуть, що x і y конгруентні (порівняльні) за модулем n і записують це так: $x \equiv y \pmod n$. Іншими словами, конгруентні числа при діленні на одне і те ж натуральне число мають однакові залишки. Так, $12 \equiv 32 \pmod{10}$, $32 \equiv 52 \pmod{10}$, $-8 \equiv -58 \pmod{10}$, $52 \equiv -78 \pmod{10}$.

Множина конгруентних чисел за модулем n є нескінченною, а рівняння $x \bmod 10 = 2$, як видно із вище наведених прикладів, має безліч розв'язків.

Функція mod є на Калькуляторах, вбудована в MS Excel та у бібліотеки мов програмування.

3. Найбільший спільний дільник.

Дільником даного натурального числа називається натуральне число, на яке дане число ділиться без остачі. Наприклад, 15 ділиться без остачі на 1, 3, 5 і 15, а число 5 є одним з дільників числа 15. У рівнянні $15 = 2 \times 6 + 3$ число 15 – ділене, 2 – частка, 6 – дільник, 3 – остача. Іншими словами, 15 при діленні на 6 дає частку 2 і остачу 3.

У загальному випадку для будь-якого натурального (а також цілого числа) a і натурального b однозначно визначені цілі числа q і r такі, що $a = q \times b + r$, $0 \le r < b$, q – частка, b – дільник, r – остача, $r = a \bmod b$ [1].

Спільним дільником кількох натуральних чисел називається число, на яке дані числа діляться без остачі. Один із способів знаходження спільних дільників – це розклад даних натуральних чисел на множники.

Найбільшим спільним дільником (НСД, GCD – Greatest Common Divisor) кількох натуральних чисел називається найбільше натуральне число, на яке кожне з даних натуральних чисел ділиться без остачі.

Два або кілька натуральних чисел, GCD яких дорівнює 1, називаються *взаємнопростими числами* і використовуються в криптології.

4. Знаходження GCD(a, b).

Якщо числа розкладаються на прості множники, то пошук їх GCD виконується саме цим шляхом. Наприклад, чисел GCD(360,336)=24, бо

$360 = 2 \times 2 \times 2 \times 3 \times 3 \times 5$; $336 = 2\times2\times2\times2\times3\times7$.

Спільними дільниками чисел 360 і 336 є множина чисел {2, 3, 4, 6, 8, 12, 24}, а 24 – найбільший дільник. Крім того, перетин множин спіль-

них дільників заданих чисел рівний {2, 2, 2, 3}, а GCD(360, 336) = 2 × 2 × 2 × 3 = 24.

Якщо ж числа важко розкласти на прості множники, для пошуку GCD використовують *алгоритм* Евкліда. У ньому остання ненульова остача і є GCD.

Даний алгоритм опирається на такі співвідношення.

GCD(*a, b*) = GCD(*b, a*);

GCD(*a, b*) = GCD(*b, a mod b*);

GCD(*a,* 0) = *a.*

Маючи ці три співвідношення алгоритму Евкліда, важко зрозуміти, як їх використовувати. Існує дуже просте пояснення, яке ми продемонструємо на конкретному прикладі.

Нехай треба знайти GCD чисел 2497 і 4200. Оскільки дані числа важко розкласти на прості множники, застосуємо алгоритм Евкліда, який використовує ділення двох цілих чисел. Ділити менше число на більше на першому кроці алгоритму – це зайва математична операція, бо $\frac{2497}{4200} = 0 \times 4200 + 2497.$ Маємо пояснення першого твердження алгоритму Еквкліда GCD(*a, b*)=GCD(*b, a*). Тому ділене – 4200, а дільник – 2497. Знаходимо частку і остачу: 4200 = 1×2497+1703. Дільник 2497 ділимо на остачу 1703 і знову знаходимо частку та остачу. Ділення продовжуємо до отримання остачі 0. Цю послідовність дій описує друге твердження алгоритму Евкліда GCD(*a, b*) = GCD(*b, a mod b*).

4200 = 1×2497+1703; 115 = 1×104+11;

2497 =1×1703+794; 104 = 9×11+5;

1703 = 2×794+115; 11 = 2×5+1;

794 = 6×115+104; 5 = 5×1+0.

В математиці на 0 ділити не можна, тому останнє рівняння в алгоритмі Евкліда пояснює, що ділення закінчене і GCD(*a,* 0) = *a*, тобто GCD(1*,* 0) = 1. Найбільший спільний дільник – це остання ненульова остача.

Отримуємо GCD(4200,2497) = GCD(2497,1703) = GCD(1703,794) = GCD(794,115) = GCD(115,104) = GCD(104,11) = GCD(11,5) = GCD(1,0)= 1.

Числа 4200 і 2497 мають лише один спільний дільник – одиницю, і є взаємнопрості. І такі взаємнопрості числа використовуються в RSA.

5. Наслідок з алгоритму Евкліда – рівняння Діофанта.

Алгоритм Евкліда можна використати для розв'язування рівняння Діофанта. Для наведеного вище прикладу воно має вигляд 2497×*x* 4200×*y* = 1. У рівнянні Діофанта дві невідомі – *x* та *y*, але воно має єдиний розв'язок, бо використовує алгоритм Евкліда пошуку GCD(2497*,*4200).

Є три способи знаходження розв'язку рівняння Діофанта. За рекурентними формулами Ейлера, з використанням ланцюгових дробів та арифметичний. Перший спосіб найлегший. Наведемо рекурентні співвідношення для послідовностей P_n, Q_n.

$$P_{-2} = 0,\ P_{-1} = 1;\ ,\ P_n = q_n P_{n-1} + P_{n-2},\ n \geq 0\ ,$$

$$Q_{-2} = 1,\ Q_{-1} = 0;\ \ Q_n = q_n Q_{n-1} + Q_{n-2},\ n \geq 0\ ,$$

де q_i –це частки від ділення в алгоритмі Евкліда. Невідомі *x* та *y* обчислюються за формулами

$$x = (-1)^k P_{k-1};\ \ y = (-1)^{k-1} Q_{k-1};\ P_k = 4200, Q_k = 2497.$$

Тут *k* –номер останнього дільника в алгоритмі Евкліда, при цьому нумерація дільників починається з нуля.

Таблиця 1

Частки від ділення в алгоритмі Евкліда

q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	k
1	1	2	6	1	9	2	5	7

Підставимо розв'язки в рівняння для перевірки: 2497×(–767)+4200×456=1.

Розглянемо другий спосіб пошуку невідомих у рівнянні Діофанта – з використанням ланцюгових дробів. Утворимо ланцюговий дріб раціонального числа $\frac{4200}{2497}$:

$$\frac{4200}{2497} = 1 + \frac{1703}{2497} = 1 + \frac{1}{\frac{2497}{1703}} = 1 + \frac{1}{1 + \frac{794}{1703}} = 1 + \frac{1}{1 + \frac{1}{\frac{1703}{794}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{115}{794}}} =$$

$$= 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{\frac{794}{115}}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{104}{115}}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{1}{\frac{115}{104}}}}}$$

$$= 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{1}{1 + \frac{11}{104}}}}} =$$

$$= 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{1}{1 + \frac{1}{\frac{104}{11}}}}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{1}{1 + \frac{1}{9 + \frac{5}{11}}}}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{1}{1 + \frac{1}{9 + \frac{1}{\frac{11}{5}}}}}}} =$$

$$= 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{6 + \frac{1}{1 + \frac{1}{9 + \frac{1}{2 + \frac{1}{5}}}}}}}$$

Маємо ланцюговий дріб $\frac{4200}{2497}$ = [1;1,2,6,1,9,2,5]. Зауважимо, що такий же ланцюговий дріб утворюють частки в алгоритмі Евкліда.

Для кожного ланцюгового дробу можна обчислити збіжні дроби – частини ланцюгових дробів

Таблиця 2

Пошук розв'язків рівняння Діофанта з використанням рекурентних співвідношень

$P_{-2}=0,\ P_{-1}=1$	$Q_{-2}=1,\ Q_{-1}=0$
$P_0=q_0\times P_{-1}+P_{-2}=1\times1+0=1$	$Q_0=q_0\times Q_{-1}+Q_{-2}=1\times0+1=1$
$P_1=q_1\times P_0+P_{-1}=1\times1+1=2$	$Q_1=q_1\times Q_0+Q_{-1}=1\times1+0=1$
$P_2=q_2\times P_1+P_0=2\times2+1=5$	$Q_2=q_2\times Q_1+Q_0=2\times1+1=3$
$P_3=q_3\times P_2+P_1=6\times5+2=32$	$Q_3=q_3\times Q_2+Q_1=6\times3+1=19$
$P_4=q_4\times P_3+P_2=1\times32+5=37$	$Q_4=q_4\times Q_3+Q_2=1\times19+3=22$
$P_5=q_5\times P_4+P_3=9\times37+32=365$	$Q_5=q_5\times Q_4+Q_3=9\times22+19=217$
$P_6=q_6\times P_5+P_4=2\times365+37=767$	$Q_6=q_6\times Q_5+Q_4=2\times217+22=456$
$P_7=q_7\times P_6+P_5=5\times767+365=\mathbf{4200}$	$Q_7=q_7\times Q_6+Q_5=5\times456+217=2497$
$x=(-1)^7\times P_6=-767$	$y=(-1)^6\times Q_6=456$

Таблиця 3

Значення збіжних дробів

Перший збіжний дріб (перший елемент)	[1] = 1
Другий збіжний дріб	$[1;1]=1+\frac{1}{1}=2$
Третій збіжний дріб	$[1;1,2]=1+\frac{1}{1+\frac{1}{2}}=\frac{5}{3}$
Четвертий збіжний дріб	$[1;1,2,6]=1+\frac{1}{1+\frac{1}{2+\frac{1}{6}}}=\frac{32}{19}$
П'ятий збіжний дріб	$[1;1,2,6,1]=1+\frac{1}{1+\frac{1}{2+\frac{1}{6+1}}}=\frac{37}{22}$
Шостий збіжний дріб	$[1;1,2,6,1,9]=1+\frac{1}{1+\frac{1}{2+\frac{1}{6+\frac{1}{1+\frac{1}{9}}}}}=\frac{365}{217}$
Сьомий збіжний дріб	$[1;1,2,6,1,9,2]=1+\frac{1}{1+\frac{1}{2+\frac{1}{6+\frac{1}{1+\frac{1}{9+\frac{1}{2}}}}}}=\frac{767}{456}$
Восьмий збіжний дріб. Останній збіжний дріб завжди дорівнює початковому дробу	$[1;1,2,6,1,9,2,5]=[1;1,2,6,1,9,2,5]=1+\frac{1}{1+\frac{1}{2+\frac{1}{6+\frac{1}{1+\frac{1}{9+\frac{1}{2+\frac{1}{5}}}}}}}=\frac{4200}{2497}$

на кожному кроці. Обчислимо збіжні дроби для отриманого вище ланцюгового дробу.

Перші шість збіжних дробів можна було і не обчислювати, бо розв'язки рівняння Діофанта знаходяться в передостанньому дробі. Розглянемо два останні збіжні дроби: $\frac{4200}{2497}$ і $\frac{767}{456}$. Вони утворюють рівняння Діофанта 4200×456 – 2497×767=1, звідки x=–767, y=456.

У роботі [5] описаний зв'язок між алгоритмом Евкліда та рекурентними співвідношеннями Ейлера. І «посередником» є саме ланцюгові дроби. Бо чисельник (знаменник) кожного збіжного ланцюгового дробу використовує значення чисельників (знаменників) двох попередніх збіжних дробів. Пояснимо це твердження на розглянутому прикладі. Частки від ділення в алгоритмі Евкліда $q_i, i = 0,1,\ldots 7$ наведені в таблиці 1.

Можна зробити такий висновок: використання збіжних дробів для знаходження розв'язків рівняння Діофанта більш трудоємке, ніж використання рекурентних співвідношень. Рекурентні співвідношення для послідовностей P_n, Q_n – це узагальнені формули для обчислення збіжних дробів, при цьому рекурентні співвідношення легко запрограмувати. Для того, щоб зручніше було їх використовувати і програмувати, заведені два додаткові початкові значення для кожної послідовності з від'ємними індексами: $P_{-2} = 0,\ P_{-1} = 1$, $Q_{-2} = 1,\ Q_{-1} = 0$.

Арифметичний спосіб утворення рівняння Діофанта використовує елементарні математичні операції – заміну, відкриття дужок, зведення подібних. Знайдемо остачі з рівнянь алгоритму Евкліда і виконаємо зворотні перетворення, починаючи з останнього рівняння.

1=11–2×5, 115=1703–2×794,
5=104–9×11, 794=2497–1×1703,
11=115–1×104, 1703=4200–1×2497.
104=794–6×115,

1=11–2×5=11–2×(104–9×11)=11–
2×104+18×11=19×11–2×104=1=
19×(115–1×104)–2×104=19×115—19×104–
2×104=19×115–21×104=1=
19×115–21×(794–6×115)=19×115–
21×794+126×115=145×115–21×794=1=
=145×(1703–2×794)–21×794=145×1703–
311×794=1=
=145×1703–311×(2497–1×1703)=145×1703–
311×2497+311×1703=
=456×1703–311×2497=1=456×(4200–1×2497)–
311×2497=456×4200–767×2497=1.

Арифметичний спосіб є «автономним», він не пов'язаний ні з рекурентними співвідношеннями, ні зі збіжними дробами. Не має значення, який метод використовувати для розв'язування рівняння Діофанта.

Загальний вигляд рівняння Діофанта $a \times x + b \times y = c$, де $c = \mathrm{GCD}(a,b)$.

Таблиця 4

Зв'язок між збіжними дробами та рекурентними співвідношеннями

1	$[q_0] = \frac{q_0}{1} = \frac{1}{1} = \frac{P_0}{Q_0} = \frac{q_0 P_{-1} + P_{-2}}{q_0 Q_{-1} + Q_{-2}}, P_{-1} = 1, P_{-2} = 0, Q_{-1} = 0, Q_{-2} = 1$
2	$[q_0; q_1] = q_0 + \frac{1}{q_1} = \frac{q_1 q_0 + 1}{q_1} = \frac{1+1}{1} = \frac{2}{1} = \frac{P_1}{Q_1} = \frac{q_1 P_0 + P_{-1}}{q_1 Q_0 + Q_{-1}}$
3	$[q_0; q_1, q_2] = q_0 + \frac{1}{q_1 + \frac{1}{q_2}} = \frac{q_2(q_1 q_0 + 1) + q_0}{q_2 q_1 + 1} = \frac{2(1+1)+1}{2 \times 1 + 1} = \frac{5}{3} = \frac{P_2}{Q_2} = \frac{q_2 P_1 + P_0}{q_2 Q_1 + Q_0}$
4	$[q_0; q_1, q_2, q_3] = \frac{q_3[q_2(q_1 q_0 + 1) + q_0] + (q_1 q_0 + 1)}{q_3(q_2 q_1 + 1) + q_1} = \frac{6 \times 5 + 2}{6 \times 3 + 1} = \frac{32}{19} = \frac{P_3}{Q_3} = \frac{q_3 P_2 + P_1}{q_3 Q_2 + Q_1}$
5	$[q_0; q_1, q_2, q_3, q_4] = \frac{q_4\{q_3[q_2(q_1 q_0 + 1) + q_0] + (q_1 q_0 + 1)\} + q_2(q_1 q_0 + 1) + q_0}{q_4\{q_3(q_2 q_1 + 1) + q_1\} + (q_2 q_1 + 1)} = \frac{1 \times 32 + 5}{1 \times 19 + 3} = \frac{37}{22} = \frac{P_4}{Q_4}$
6	$[q_0; q_1, q_2, q_3, q_4, q_5] = \frac{9 \times 37 + 32}{9 \times 22 + 19} = \frac{365}{217} = \frac{P_5}{Q_5} = \frac{q_5 P_4 + P_3}{q_5 Q_4 + Q_3}$
7	$[q_0; q_1, q_2, q_3, q_4, q_5, q_6] = \frac{2 \times 365 + 37}{2 \times 217 + 22} = \frac{767}{456} = \frac{P_6}{Q_6} = \frac{q_6 P_5 + P_4}{q_6 Q_5 + Q_4}$
8	$[q_0; q_1, q_2, q_3, q_4, q_5, q_6, q_7] = \frac{5 \times 767 + 365}{5 \times 456 + 217} = \frac{4200}{2497} = \frac{P_7}{Q_7} = \frac{q_7 P_6 + P_5}{q_7 Q_6 + Q_5}$

6. Генерування ключів у криптосистемі RSA.

Вважаємо, що тепер знайомство з алгоритмом генерування ключів у криптосистемі RSA буде зрозумілим, наведемо його [1, 2, 3].

1. Випадково вибираємо два великих простих числа p і q.

2. Обчислюємо їх добуток $n=p\cdot q$.

3. Обчислюємо число $\phi=(p-1)\cdot(q-1)$.

4. Вибираємо третє випадкове число e таке, що $e<\phi$ і взаємно просте з ϕ, тобто GCD $(e,\phi)=1$.

5. Пара чисел $(e,\ n)$ є утвореним *публічним* (*відкритим*) ключем.

6. Записуємо діофантове рівняння $e\times x+\varphi\times y=1$, у ньому e та φ відомі.

7. Знаходимо таємний ключ – число x, що задовольняє умову:

$x\times e\equiv 1 \pmod{\phi}$.

Пояснимо останнє рівняння. Таємний ключ – це множник при e, якщо він додатній ($d = x$ при $x > 0$). Якщо $x < 0$, то $d = x \bmod \phi$ (для знаходження таємного ключа шукаємо остачу від ділення цілого числа x на натуральне число ϕ).

8. Пара чисел $(d,\ n)$ є *таємним* (*закритим*) ключем криптосистеми RSA.

Очевидно, що наведений вище алгоритм можна запрограмувати.

Зауважимо, що в літературі трапляються також рівняння $d\times e\equiv 1 \pmod{\phi}$ (правильне лише при d=x>0), $d\equiv e^{-1}(mod\varphi)$ (не слід трактувати e^{-1} як $\frac{1}{e}$).

7. Програмна реалізація алгоритму Евкліда та розв'язування рівняння Діофанта.

Для виконання обчислень при генеруванні ключів можна використати звичайний або програмований калькулятор, якщо вхідні дані – числа p, q та e є невеликими. Автоматизувати обчислення можна в MS Excel, а найкраще написати програму на будь-якій мові програмування. Нижче на рис.1–4 наведені скріни аркушів програми MS Excel з даними та формулами для розглянутих вище прикладів.

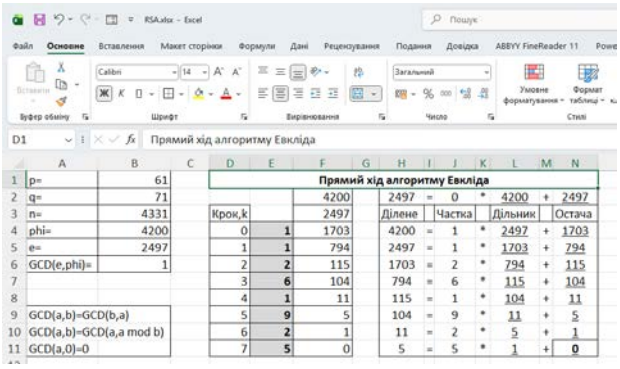

Рис. 1. Результати пошуку GCD (e, φ) за алгоритмом Евкліда

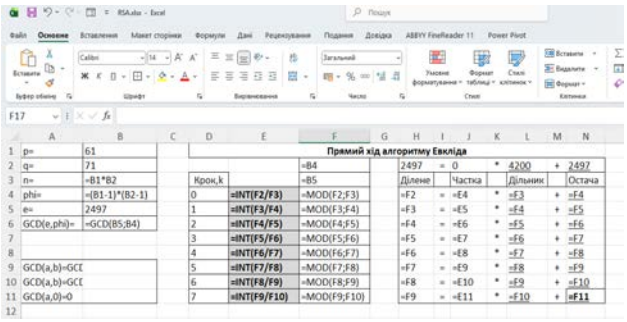

Рис. 2. Формули пошуку GCD (e, φ) за алгоритмом Евкліда

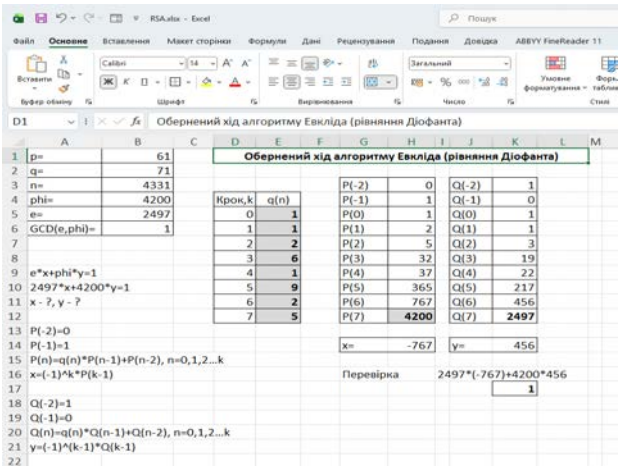

Рис. 3. Розв'язки рівняння Діофанта за рекурентними формулами

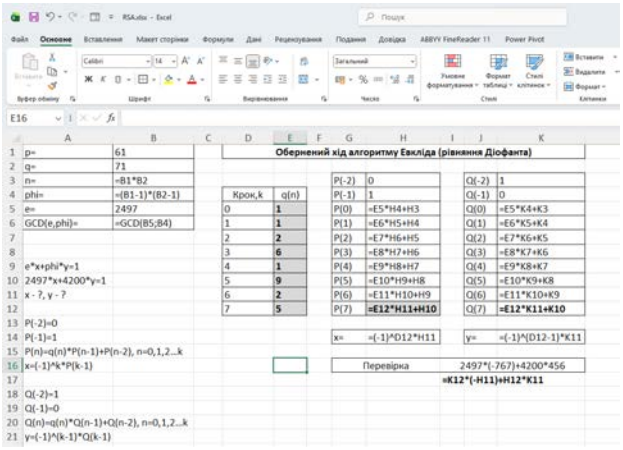

Рис. 4. Формули для пошуку розв'язків рівняння Діофанта

8. Застосування ключів криптосистеми RSA.

Пара ключів – відкритий і таємний – генеруються для юридичних та фізичних осіб. Фізичні особи можуть згенерувати їх для особистого користування, але юридичної сили такі ключі не матимуть. Ключі також генерують ЦСК (Центри Сертифікації Ключів), АЦСК (Акредитовані центри сертифікації ключів) та КНЕП (Кваліфіковані

Надавачі Електронних довірчих Послуг). Для відкритого ключа готують документ під назвою «Сертифікат відкритого ключа». Відкритий ключ зберігається у таблиці відкритих ключів. Таємний ключ копіюють на зовнішній носій інформації. Це може бути флеш-диск, флеш-токен або смарт-картка. Файли з таємним ключем, не мають єдиного фіксованого розширення, воно залежать від конкретного програмного забезпечення для генерації ключів, переважно зустрічаються .dat, .pfx, .p12, .pk8, .zs2, .jks. Доступ до файла захищається особистим паролем власника ключа. На ключовому флеш-диску можна також зберігати сертифікат.

Смарт-картка – це пластикова картка з чіпом, яка доступ до якої захищений пін-кодом власника, вона зберігає КЕП (Кваліфікований електронний підпис**).** КЕП використовують не лише для підписання документів, але й для подання звітності, доступу до держпослуг, в якості електронної печатки для юридичних осіб.

Флеш-токен – це апаратний носій для підписування електронних документів.

Найпоширеніші на сьогодні застосування несиметричних криптосистем – шифрування інформації та підписування електронних повідомлень. Відповідні алгоритми у криптосистемі RSA використовують операції піднесення до степеня та знаходження остачі від ділення. При цьому використовуються елементи колишніх симетричних криптосистем – заміна символів та блокове шифрування. Алгоритми шифрування та дешифрування, які використовуються в несиметричних криптосистемах опубліковані в мережі Інтернет і не становлять таємниці. Зауважимо лише, що зашифрований документ з використанням алгоритмів криптосистеми RSA – це набір десяткових цифр.

Шифрування собистої інформації. Якщо треба захистити певну інформацію, використовують для шифрування особисті ключі: для шифрування – відкритий, для дешифрування – таємний. Очевидно, що таємний ключ (ключ дешифрування) має бути в безпеці, оскільки від нього залежить доступ до зашифрованої інформації. Таким чином можна захищати в електронному вигляді особисті, фінансові, медичні дані від доступу неуповноважених осіб.

Обмін повідомленнями по відкритому каналу зв'язку. Відправник шифрує повідомлення відкритим ключем отримувача і надсилає його каналом зв'язку. При цьому відправник не може прочитати зашифроване повідомлення, це може зробити лише отримувач за допомогою свого таємного ключа.

Підписування електронних документів без приховування їх змісту. Передбачає шифрування документа таємним ключем підписувача і надсилання отримувачу двох документів: оригінального та підписаного (зашифрованого) документа. Кажуть, що на електронний документ накладають ЕЦП (Електронний Цифровий підпис). Перевірка підпису полягає у виконанні двох кроків: дешифруванні ЕЦП відкритим ключем підписувача та порівнянні його з оригіналом документа. Якщо обидва документи співпадають, то підписувач ідентифікований і підпис дійсний. ЕЦП має юридичну силу власноручного підпису і найвищий рівень безпеки. Використовуються лише ключі підписувача електронних документів.

Підписування електронних документів з приховуванням їх змісту. Електронний документ шифрується відкритим ключем отримувача і одночасно підписується таємним ключем відправника. Це фактично шифрування одного і того ж документа різними ключами. Після отримання документа його дешифрують таємним ключем отримувача і перевіряють ЕЦП відкритим ключем відправника. Після перевірки ЕЦП і дешифрування обидва документи повинні бути ідентичними. Тільки тоді вони не спотворені, не сфальсифіковані і ЕЦП чинний.

У такий спосіб захищаються державні електронні послуги, електронний документообіг, електронна звітність, дистанційні банківські операції, інші важливі торгові і фінансові трансакції. Згаданий вище КЕП – це електронний ключ для роботи з цифровими сервісами, який призначений для накладання електронного підпису, що відповідає стандартам безпеки України та ЄС (eIDAS), Завдяки використанню криптографічних алгоритмів він має юридичну силу власноручного підпису. КЕП можна отримати в застосунку «Дія».

Сьогодні в Україні працює більше 100 державних сервісів та Інтернет-послуги банків. державних та бізнес-сервісів онлайн (Дія, податкова, Пенсійний фонд, тощо). електронного документообігу для громадян і бізнесу

Поділ таємниці. Схеми поділу таємниці використовуються для спільного управління технологіями та процесами. В такому управлінні можуть приймати участь n об'єктів або суб'єктів. Загальна таємниця ділиться на n частин, які називають частками. При об'єднанні $k \leq n$ часток таємниці використовується метод попереднього розподілу ключів, що дозволяє одноразово встановити ключ при згоді не менше ніж k об'єктів та суб'єктів. При цьому можлива ситуація, коли жодна група з n–1

особи не можуть отримати доступ до захищеної інформації. Поділ таємниці – це послідовність дій (процедура, протокол), яка визначає черговість дій для встановлення ключів чи паролів, які забезпечують доступ до таємниці.

Онлайн голосування. Має на меті забезпечити таємність голосування, неможливість його фальсифікації і правильність підрахунку результатів. Проведення голосування у такому форматі передбачає використання ЕЦП (відповідно наявності у голосуючого чинного Сертифіката відкритого ключа та носія з таємним ключем) та проведення зборів в режимі відеозасідання. При цьому канал зв'язку має бути надійно захищеним і передбачати дистанційну ідентифікацію фізичної особи голосуючого. Усі дії користувачів (приймають участь в голосуванні) та адміністраторів (обслуговують програмне забезпечення) неможливо видалити або змінити, лише зафіксувати. Будь-які правки в результатах голосування або документах не можуть залишатися непоміченими. Результати голосувань автоматично вносяться до протоколу і фіксуються з перевіркою ЕЦП або КЕП. Протоколи можуть відразу публікуватися на сайті установи [4].

Електронний нотаріат. Сьогодні можна підтвердити факт існування юридичного документа без розголошення його змісту. До тексту документа застосовують необоротну вкорочуючу функцію (її називають хеш-функцією). Вона перетворює вхідні дані будь-якого розміру в дані фіксованого розміру, які називають хеш документа, хеш-код, хеш-відбиток, хеш-образ, хеш-значення. Прикладом такої функції є, наприклад, розглянута вище функція *mod*. Хеш документа зберігається у нотаріуса в електронному вигляді. Знаючи хеш-образ документа, неможливо відновити сам документ. Для нотаріального підтвердження володіння електронним документом нотаріус накладає ЕЦП на хеш-документ і публікує хеш-документ, його ЕЦП та свій відкритий ключ (або вказує електронну адресу, за якою ці дані знаходяться). Після перевірки ЕЦП дешифрований хеш-документ порівнюють з його оригіналом. Якщо обидва документи ідентичні, це дозволяє підтвердити цифровий підпис та наявність оригіналу документа у нотаріуса. При цьому зміст самого документа відомий лише нотаріусу, оскільки такий протокол передбачає використання лише хеш-образу юридичного документа, який не містить жодної інформації про його оригінал.

Хеш-функції використовуються також при накладанні і перевірці цифрових підписів, для зберігання паролів (зберігають не сам пароль, а його хеш), при перевірці цілісності файлів (якщо хеш змінився, то файл було змінено) та у технології блокчейн.

Ринок криптовалют та технологія блокчейн. Криптовалюти та інші цифрові активи були розроблені з використанням криптографічних методів, що робить транзакції безпечними та надійними. Криптовалюта – це цифрові гроші. Операції з криптовалютою використовують технологію блокчейн, передбачають реєстрацію на криптовалютній біржі та створення криптовалютного гаманця. При його створенні генерується пара ключів – відкритий і таємний. Адреса гаманця генерується за допомогою відкритого ключа, її можна оприлюднювати. Закритий ключ використовується для накладання цифрових підписів та перевірки трансакцій з криптовалютою. Відкритий ключ використовується відправником для шифрування інформації, тоді як закритий ключ використовується одержувачем для її розшифрування. Ця система перевірки цифрового підпису гарантує, що лише особа, яка має приватний ключ, пов'язаний із відповідним гаманцем криптовалюти, може переміщувати кошти. Для створення унікальних ідентифікаторів транзакцій та забезпечення цілісності даних використовуються хеш-функції. Після перевірки транзакції шляхом підтвердження хешу, що міститься в цифровому підписі, цю транзакцію додають до книги блокчейну. Криптовалютні транзакції шифруються та розшифровуються з використанням ключів.

Висновки. У цій публікації ми просто і доступно пояснили алгоритм генерування ключів у криптосистемі RSA. Запропонували найпростіший спосіб генерування та його реалізацію в MS Excel для невеликих чисел. Описали використання ключів та пояснили необхідність зберігання у безпеці таємного ключа та призначення сертифіката відкритого ключа. Знаючи, як працює алгоритм генерування ключів та алгоритми шифрування і накладання ЕЦП, можна пояснювати термінологію Законів України «Про електронні документи та електронний документообіг», «Про електронні довірчі послуги», знайомитись з іншими застосуваннями криптосистеми, шляхами її зламування та стійкістю, зрозуміти необхідність розробки наступного покоління криптосистем.

При роботі над цією статтею ми використовували AI ChatGPT. І зауважили, що програма робила помилки при розв'язуванні рівняння Діофанта, неправильно обчислювала деякі збіжні дроби, але виправляла помилки після зауважень. ChatGPT пропонує також програми генерування ключів на різних мовах програмування.

Список літератури:

1. Вербіцький О.В. Вступ до криптології. – Львів: Вид-во науково-технічної літератури, 1998. – 247 с.
2. В.Ємець. Сучасна криптографія. Основні поняття / В.Ємець, А.Мельник, Р.Попович. – Львів: БаК, 2003. – 144 с.
3. Вікіпедія – вільна енциклопедія. URL: https://uk.wikipedia.org/wiki/RSA.
4. 201923_golosuvannya-onlayn-z-bud-yako-tochki-svtu-yak-pratsyu-nova-sistema-vdeozasdan-dlya-organv-vladi. URL: https://biz.ligazakon.net/news/
5. Засадна Х.О. Алгоритми генерування ключів криптосистеми RSA/ Засадна Х.О. // Соціально-економічні проблеми сучасного періоду України. Фінансовий ринок України: стабілізація та євроінтеграція (зьірник наукових праць) / НАН України. Інститут регіональних досліджень. – Львів, – 2010. – Вип.1(81). – С.458-467.

Zasadna Kh. O., Kotsun V. I. KEY GENERATION IN THE RSA CRYPTOSYSTEM RSA AND THEIR APPLICATION

The article discusses the algorithm for generating keys in the asymmetric RSA cryptosystem and their use. While teaching the disciplines «Information Security» and «Cryptographic Information Protection» to students of various fields of study, we encountered the problem of an accessible explanation of the key generation algorithm. The algorithm presented in printed literature and on the Internet cannot be understood without a prior explanation of some fundamentals of the school mathematics curriculum. The article presents the mathematical foundations of asymmetric cryptosystems, necessary for understanding the key generation algorithm in the RSA cryptosystem. Particular attention is paid to the methods of solving the Diophantine equation.

To automate key generation, a template is proposed in the Microsoft Excel spreadsheet editor.

Many modern electronic services require an understanding of the protocols for using two keys. In the near future, every citizen of Ukraine will have them and will be forced to use them to identify and protect personal electronic information.

Today, there are quite many areas of application for two-key cryptosystems. With the emergence of the crypto industry, it has become clear that cryptographic methods of information protection have long been actively integrated into our everyday lives.

We believe that the relatively young science known as cryptology should now be studied at every higher education institution: not necessarily as a separate discipline, but possibly as part of courses that deal with information security and its protection. Schools are already showing interest in cryptology, as evidenced by numerous publications and video lessons available on the Internet. Symmetric cryptosystems, which were used until 1976, are particularly interesting: they explain the formulation of the fundamental problems of cryptology and contain many intriguing algorithms for information encryption and decryption that develop logical thinking and creativity and can be successfully programmed.

Keywords: *RSA, GCD, Euclidean algorithm, Diophantine equation, continued fractions, application of asymmetric cryptosystems.*

Дата першого надходження статті до видання: 09.03.2026
Дата прийняття статті до друку після рецензування: 03.04.2026
Дата публікації (оприлюднення) статті 11.05.2026