DOI https://doi.org/10.15407/csc.2022.02.011
УДК 51.681.3

С.Л. КРИВИЙ, доктор фіз.-мат. наук, професор, кафедра інтелектуальних програмних систем, факультет комп'ютерних наук та кібернетики, Київський Національний університет імені Тараса Шевченка 03022, м. Київ, просп. Академіка Глушкова, 4Д, Україна, ORCID: https://orcid.org/0000-0003-4231-0691 sl.krivoi@gmail.com

Ю.О. НОРТМАН, студентка першого курсу магістратури факультету комп'ютерних наук та кібернетики, Київський Національний університет імені Тараса Шевченка 03022, м. Київ, проспект Академіка Глушкова, 4Д, ynortman@gmal.com

ПРОТОКОЛ ОБМІНУ ПОВІДОМЛЕННЯМИ НА ОСНОВІ ЛІНІЙНИХ ФУНКЦІЙ І ОПЕРАТОРІВ

Пропонується простий спосіб обміну повідомленнями на основі лінійних функцій і лінійних невироджених операторів у лінійних просторах скінченної розмірності та систем лінійних діофантових рівнянь над множиною натуральних чисел.

Ключові слова: *невироджений лінійний оператор, лінійні діофантові рівняння, обмін інформацією.*

Вступ

У статті розглядається простий протокол обміну повідомленнями між абонентами (Алісою та Бобом), який ґрунтується на властивостях лінійних функцій і лінійних невироджених операторів у векторних просторах скінченної розмірності та систем лінійних діофантових рівнянь над множиною натуральних чисел. Такого типу системи мають вигляд

$$S=\begin{cases} L_1(x)=a_{11}x_1+a_{12}x_2+\ldots a_{1n}x_n=b_1,\\ L_2(x)=a_{21}x_1+a_{22}x_2+\ldots a_{2n}x_n=b_2,\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ L_m(x)=a_{m1}x_1+a_{m2}x_2+\ldots a_{mn}x_n=b_m, \end{cases} \qquad (1)$$

де $m<n$, а коефіцієнти $a_{ij}\in Z$ є цілими числами і пошук розв'язків ведеться або у множині натуральних чисел, або в деякій скінченній області. Усі відомі алгоритми розв'язання систем лінійних рівнянь у множині натуральних чисел належать до класу складності $\#NP$ [1]. Ці алгоритми знаходять базис множини всіх розв'язків такої системи.

Протокол обміну повідомленнями

Пропонований протокол обміну повідомленнями ґрунтується на властивостях операторів у лінійному просторі та складності алгоритмів розв'язання систем лінійних неоднорідних діофантових рівнянь у множині натуральних чисел. Оператор, який діє у векторному просторі й має невироджену матрицю, реалізує бієктивне відображення векторного простору на цей самий простір. Використовуючи деяку послідовність таких відображень зі зсувами на деякі задані вектори, дістаємо можливість за результуючим значенням оператора знайти його початкове значення. У протоколі є

лише три кроки, які описують взаємодію між абонентами.

Аліса і Боб звертаються до Джона з проханням згенерувати для них спільний ключ c — ціле додатне число або вектор таких чисел заданої розмірності. Після того, як такий ключ Аліса та Боб отримали, виконується поданий далі протокол.

К р о к 1. 1) Аліса будує систему лінійних виразів Ax, де A — матриця розмірності $p \times n$, $B_0, B_1, \ldots, B_s$ — невироджені матриці розмірності $p \times p$ і вектори $b_0, b_1, \ldots, b_s, b$, де b, b_i — приватні вектори зсуву з цілими координатами, $i = 1, 2, \ldots, s$;

2) будує систему виразів

$$D(x) = B_s(B_{s-1}(\ldots B_1(B_0(A(x)) + b_0) + b_1)\ldots) + b_{s-1}) + b_s) + b$$

або

$$D(x) = B_s(B_{s-1(B_0}(\ldots B_1(B_0(A(x) + b_0) + b_1)\ldots) + b_{s-1}) + b_s) + b;$$

3) висилає Бобу вирази Ax і $D(x)$ відкритим каналом.

К р о к 2. 1) Бобу потрібно передати Алісі повідомлення v. Для цього він розв'язує систему лінійних рівнянь $Ax = v$ і знаходить розв'язок x розмірності $1 \times n$.

2) Обчислює значення $D(x) = d$ і висилає Алісі значення $d' = d + c$ відкритим каналом.

К р о к 3. 1) Аліса за значенням d' обчислює значення

$$B_0^{-1}(B_1^{-1}\ldots(B_{s-1}^{-1}(B_s^{-1}(d' - c) - b) - b_s) - b_{s-1}))\ldots - b_1) - b_0 = A(x)$$

за допомогою обернених матриць до матриць B_i.

Приклад 1. Нехай Джон згенерував для Аліси і Боба спільний ключ $c = (1, 2)$.

К р о к 1. 1) Аліса будує вираз

$$Ax = \begin{pmatrix} -2x_1 + x_2 - x_3 + 3x_4 \\ -3x_1 + 2x_2 - x_3 + x_4 \end{pmatrix}$$ і одну невироджену квадратну матрицю розмірності 2×2:

$$B = \begin{pmatrix} 2 & 1 \\ -1 & -1 \end{pmatrix}$$ та вектори $b_0 = (1,1), b = (1,1)$;

2) перетворює Ax до виразу $D(x) = B(A(x) + b_0) + b$:

$$D(x) = B(A(x) + b_0) + b = \begin{pmatrix} -7x_1 + 4x_2 - 3x_3 + 7x_4 + 4 \\ 5x_1 - 3x_2 + 2x_3 - 4x_4 - 4 \end{pmatrix};$$

3) висилає Бобу вирази Ax і $D(x)$ відкритим каналом зв'язку.

К р о к 2. 1) Бобу потрібно вислати Алісі вектор $v = (5, -2)$. Для цього він розв'язує систему лінійних рівнянь $Ax = v$ і знаходить розв'язок $x = (1,0,2,3)$;

2) обчислює значення Ax і $D(x)$: $v = Ax = (5,-2)^t$ і $d = D(x) = (12,-4)^t$. Пара $v = (5, -2)$ є приватною інформацією, а пара $d = d' + c = (12, -4) + (1,2) = (13, -2)$ — відкритою;

3) висилає Алісі $d' = (13, -2)$ відкритим каналом.

К р о к 3. 1) Аліса обчислює обернену матрицю: $B^{-1} = \begin{pmatrix} 1 & 1 \\ -1 & -2 \end{pmatrix}$. За значенням d' і векторами зсуву та оберненою матрицею B^{-1} обчислює таку послідовність значень:

$$d' - c = (13,-2) - (1,2) = (12,-4);$$
$$(12,-4) - b = (12,-4) - (1,1) = (11,-5);$$
$$B^{-1}(11,-5)^t = (6,-1);$$
$$(6,-1) - b_0 = (6,-1) - (1,1) = (5,-2) = v.$$

Отже, обмін повідомленням відбувся.

Твердження 1. Протокол коректно виконує обмін.

Доведення випливає безпосередньо з властивостей лінійних невироджених операторів у векторних просторах скінченної розмірності. Справді,

$$B_0^{-1}(B_1^{-1}(\ldots(B_{s-1}(B_s^{-1}(d' - c) - b) - b_s) - b_{s-1}))\ldots$$
$$\ldots - b_1) - b_0 = B_0^{-1}(B_1^{-1}(\ldots(B_{s-1}(B_s^{-1}(B_s(B_{s-1}(\ldots$$
$$\ldots B_1(B_0(A(x) + b_0) + b_1)\ldots) + b_{s-1}) + b_s) + b) -$$
$$-b) - b_s) - b_{s-1}))\ldots - b_1) - b_0 = A(x).$$

на підставі того, що

$$d' - c = D(x) = B_s(B_{s-1}(\dots B_1(B_0(A(x)+$$
$$+b_0)+b_1)\dots)+b_{s-1})+b_s)+b$$

дістаємо

$$B_0^{-1}(B_1^{-1}(\dots(B_{s-1}(B_s^{-1}(d' - c) - b) - b_s) - b_{s-1}))\dots$$
$$\dots - b_1) - b_0 = B_0^{-1}(B_1^{-1}(\dots(B_{s-1}(B_s^{-1}(B_s(B_{s-1}(\dots$$
$$\dots B_1(B_0(A(x)+b_0)+b_1)\dots)+b_{s-1})+b_s)+b) - b) - b_s) -$$
$$-b_{s-1}))\dots - b_1) - b_0 = A(x).$$

Надійність протоколу

Надійність наведеного протоколу цілком ґрунтується на таємності ключа *c* та складності розв'язання системи лінійних діофантових рівнянь у множині натуральних чисел, тобто на криптографічній функції f_c: $Dx+c=d'$. Оскільки явно при обміні повідомленнями саме повідомлення не фігурує у цьому процесі, то очевидними діями для його знаходження є розв'язання системи рівнянь вигляду $Dx+c=d'$. Але в результаті такого розв'язання зловмисник отримує вектор-розв'язок *y*, який не має жодного стосунку до реального повідомлення. Оскільки значення вектора *x* і *c* зловмиснику невідоме, то знайти значення *Ax* він не може. Наприклад, якщо зловмиснику вдалося розв'язати систему рівнянь із наведеного прикладу

$$B(A(x)+b_0)+b = \begin{pmatrix} -7x_1+4x_2-3x_3+7x_4=9 \\ 5x_1-3x_2+2x_3-4x_4=-1 \end{pmatrix} \quad (2)$$

і знайти серед інших розв'язок (0,1,17,8), то цей вектор при підстановці в *Ax* дає значення (8, -7).

Як зазначалося, складність розв'язання системи лінійних діофантових рівнянь (а точніше, побудови гільбертового базису множини всіх розв'язків системи) в множині натуральних чисел у загальному випадку належить до класу складності ***#NP***. Якщо зловмисник має у своєму розпорядженні комп'ютер і достатній об'єм пам'яті, то зрештою він зможе розв'язати відповідну систему. Але потім йому потрібно підібрати значення ключа *c*, за яким можна знайти явний текст, що виглядає доволі проблематично на підставі нескінченності способів вибору вектора *c*. Отже, в цьому протоколі використовується криптографічна функція із секретом $f_c=D(x)+c$, де секретом виступає вектор *c*.

Складність обчислень у наведеному протоколі не є високою і єдине, що потребує затрат часу — це обчислення послідовності добутків матриць і обернених до них матриць, на підставі яких будується вираз *D*. Але із наведеного опису протоколу та обчислень випливає, що можна обійтися двома-трьома матрицями, оскільки послідовність добутків невироджених матриць буде матрицею і її можна знайти за матрицями *A* і *D*. Тому досить мати дві-три матриці і для них знаходити обернені матриці. Проблему обчислення оберненої матриці легко розв'язати, якщо вибирати матриці спеціальним чином. Добре відомо [3], що для матриці вигляду

$$M = \begin{pmatrix} I_n & A & 0 \\ 0 & I_n & B \\ 0 & 0 & I_n \end{pmatrix}$$ обернена матриця має вигляд

$$M^{-1} = \begin{pmatrix} I_n & -A & AB \\ 0 & I_n & -B \\ 0 & 0 & I_n \end{pmatrix}$$, а для матриці

$$M_1 = \begin{pmatrix} I_n & 0 & 0 \\ A & I_n & 0 \\ 0 & 0B & I_n \end{pmatrix}$$ обернена матриця має вигляд

$$M_1^{-1} = \begin{pmatrix} I_n & 0 & 0 \\ -A & I_n & 0 \\ BA & -B & I_n \end{pmatrix}.$$

Наприклад, для матриці

$$M = \begin{pmatrix} 1 & 0 & 5 & 6 & 0 & 0 \\ 0 & 1 & 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 & 1 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$ обернена матриця

набуває вигляду:

$$M^{-1}=\begin{pmatrix}1&0&-5&-6&16&-2\\0&1&-3&-4&10&-2\\0&0&1&0&-2&-2\\0&0&0&1&-1&2\\0&0&0&0&1&0\\0&0&0&0&0&1\end{pmatrix}.$$

Це дає можливість щоразу обирати різні матриці в процесі побудови D.

Обмін наперед заданими повідомленнями

Розгляньмо питання про те, як можна обмінятися заданим повідомленням v, оскільки в реальній ситуації v не завжди може бути розв'язком системи $Ax=v$. Для того щоб обмінятися повідомленням v описаним протоколом, потрібна сумісність системи $Ax = v$. Це потребує додаткових двох пересилань між абонентами з метою вибору матриці A для забезпечення сумісності системи $Ax = v$. Якщо таке узгодження відбулося, то далі все йде за описаним протоколом.

Можливим є інший спосіб такого узгодження без додаткових пересилань. Таку можливість дає вибір області, над якою розв'язується система $Ax=v$.

Якщо Аліса і Боб вибрали скінченне поле лишків за модулем великого простого числа p і систему $Ax = v$, серед рівнянь якої немає лінійно залежних, то така система буде завжди сумісною над цим полем. Крім того, в цьому разі *TSS*-алгоритм [2] будує базис множини всіх розв'язків системи.

Отже, таким способом можна передавати повідомлення довільної довжини.

Приклад 2. Нехай потрібно передати вектор значень v за допомогою матриць A, D і вектора a, де $a=(-2, -3,4, -5, -2,2)$, $v=(16, 18, 21,12, 21, 16)$,

$$A=\begin{pmatrix}1&1&1&1&1&1&1&0&0&0&0&0\\1&1&1&1&1&1&1&1&0&0&0&0\\1&1&1&1&1&1&0&0&1&0&0&0\\1&1&1&1&1&1&0&0&0&1&0&0\\1&1&1&1&1&1&0&0&0&0&1&0\\1&1&1&1&1&1&0&0&0&0&0&1\end{pmatrix}.$$

У даному випадку очевидно, що рівняння системи є лінійно незалежними (за модулем 23) і тому система має розв'язки в полі Z_{23}. Розв'язуємо систему $Ax+a=v \pmod{23}$ *TSS*-алгоритмом, модифікованим під область поля Z_{23} [2]:

$$Ax+a-v=$$
$$=\begin{pmatrix}1&1&1&1&1&1&1&0&0&0&0&0&5\\1&1&1&1&1&1&0&1&0&0&0&0&2\\1&1&1&1&1&1&0&0&1&0&0&0&6\\1&1&1&1&1&1&0&0&0&1&0&0&6\\1&1&1&1&1&1&0&0&0&0&1&0&0\\1&1&1&1&1&1&0&0&0&0&0&1&9\end{pmatrix}=$$
$$=0\,(\mathrm{mod}\,23).$$

Розв'язком цієї системи (розв'язків є декілька і для роботи обирається один із них) є вектор $d=(0,0,0,0,0,0,18,21,17,17,0,14)$.

Побудуймо систему D шляхом застосування до системи $Ax+a-v$ матриці M:

$$M(Ax+a-v)+b\,(\mathrm{mod}\,23)=$$
$$=\begin{pmatrix}12&12&12&12&12&12&10&5&6&0&0&2\\8&8&8&8&8&8&0&1&4&0&0&21\\5&5&5&5&5&5&0&0&0&2&2&1\\0&0&0&0&0&0&0&0&1&1&2&11\\1&1&1&1&1&1&0&0&0&1&0&0\\1&1&1&1&1&1&0&0&0&0&1&9\end{pmatrix},$$

де b — деякий вектор зсуву. Діставши в такий спосіб вираз D, підставимо знайдений розв'язок у цю систему й обчислимо результати $D(x)+c$, де c — ключ. Подальша робота виконується за протоколом.

У такому варіанті протоколу секретом є ключ c, без знання якого неможливо знайти правильний розв'язок системи рівнянь і застосувати лінійні перетворення матриць. А коли зловмиснику невідомий і модуль поля p, то ця обставина покращує надійність протоколу.

Зауважмо, що складність розв'язання систем лінійних рівнянь у полі лишків за модулем простого числа належить до поліноміального класу складності [2].

Спрощений варіант протоколу

Описаний протокол та обчислення в ньому можна спростити, якщо розглядати не систему виразів Ax, а один вираз $l(x)$ з коефіцієнтами із множини цілих чисел Z.

Як і в попередньому протоколі Джон генерує для Аліси та Боба ключ — ціле число c, а потім вони виконують кроки такого протоколу.

К р о к 1. 1) Аліса будує лінійний вираз

$$l(x) = a_1x_1 + a_2x_2 + \ldots + a_nx_n, \quad (3)$$

де $a_i \in Z, x_i \in N$;

2) перетворює $l(x)$ до вигляду

$$L(x) = c_k(c_{k-1}(\ldots(c_1(l(x)) + c^1)\ldots) + c^{k-1}) + c^k$$

або

$$L(x) = c_k(c_{k-1}(\ldots(c_1(l(x) + c^1)\ldots) + c^{k-1}) + c^k, \quad (4)$$

де $c_i \in Z$, $c_{i,}c^i \neq 0$, $c^i \in Z$ — довільні;

3) висилає Бобу вирази

$$l(x) = a_1x_1 + a_2x_2 + \ldots + a_n,$$

$$L(x) = c_k(c_{k-1}(\ldots(c_1(l(x)) + c^1)\ldots + c^{k-1}) + c^k \quad (5)$$

або

$$l(x) = a_1x_1 + a_2x_2 + \ldots + a_n,$$

$$L(x + a) = c_k(c_{k-1}(\ldots c_1(l(x) + c^1)\ldots + c^{k-1}) + c^k) \quad (6)$$

відкритим каналом.

К р о к 2. 1) Боб обирає вектор x і обчислює значення $l(x) = v$, яке є приватним;

2) підставляє вектор x у вираз (5) або (6) і обчислює значення $d = L(x)$;

3) висилає значення $d' = d + c$ відкритим каналом Алісі.

К р о к 3. 1) За значенням d' Аліса обчислює значення v, оскільки їй відомі всі необхідні дані.

Приклад 3. Нехай, наприклад, Джон згенерував для Аліси і Боба ключ — число 3.

К р о к 1. Аліса будує

$$l(x)=2x_1+4x_2+6x_3\ 5x_4 \quad (7)$$

і до виразу (7) застосовує такі перетворення (параметри $c_2 = 3$, $c_1=5$, $c^1 = 2$, $c^2= 4$):

$$L(x) = -3(5(l(x) - 2) + 4 = \\ = -30x_1 - 60x_2 - 90x_3 + 75x_4 + 34. \quad (8)$$

Висилає вирази $l(x)$ і $L(x)$ Бобу, тобто $2x_1+4x_2+6x_3\text{-}5x_4$ і $-30x_1 - 60x_2 - 90x_3 + 75x_4 + 34$.

К р о к 2. Бобу потрібно вислати Алісі значення $v=2$. Для цього Боб розв'язує рівняння $l(x)=2$, знаходить розв'язок $x=(11,5,0,8)$, підставляє його у вираз $L(x)$ і дістає значення

$$L(x) = \text{-}330\ \text{-}300+600+34=4.$$

Висилає Алісі значення $d'=4+3=7$ (відкритим каналом).

К р о к 3. За отриманим значенням $d'=7$ Аліса знаходить приватне значення $l(x)$:

$$7\text{-}c=7\text{-}3=4,\ 4\text{-}c^2=4–4=0,\ 0/\text{-}15=0,$$

отже, $l(x)\text{-}2=0$ і $l(x) =2$.

Для обміну заданими повідомленнями таким протоколом Алісі необхідно побудувати вираз $l(x)$ так, щоб серед коефіцієнтів були два коефіцієнти з різними знаками та взаємно прості між собою. Ця умова є достатньою для сумісності рівняння $l(x)=v$ у множині натуральних чисел. Зазначмо, що обраними векторами та матрицями в розглянутих протоколах можна користуватися багаторазово, не змінюючи часто значення c, і змінювати щоразу при сеансі обміну лише значення векторів зсуву. А якщо в протоколах використовувати ці параметри одноразово, то вони стають доволі надійними і практичними способами обміну повідомленнями між абонентами.

Залишається визначитися з Джоном, який генерує ключі. Якщо між Алісою і Бобом є закритий канал зв'язку, то цей канал відіграє роль Джона і за його допомогою відбувається обмін ключами. Якщо ж такого каналу нема, то обмін ключами можна виконати за допомогою дискретного логарифма в полі лишків за модулем великого простого числа. Такі протоколи обміну ключами є добре відомими (наприклад,

протоколи Діффі-Хеллмана, Ель-Гамаля, Шаміра) [4, 5]. Справді, якщо в полі F_p використати дискретний логарифм, то обмін ключем можна виконати за протоколом Ель-Гамаля. У такий спосіб можна обійтися без Джона в процесі обміну ключами.

Приклад 4. Нехай Аліса з Бобом працюють у полі лишків за модулем p=11. Аліса обирає твірний мультиплікативної групи поля g=2 і приватний елемент d=7 та елемент m=3, яким вона хоче обмінятися з Бобом. При цьому відкритим ключем Боба є елемент d_B =5, а приватним — c_B =4. Аліса обчислює два елементи (r,e), де $r = g^d$, $e = m \cdot 5^d = m2^{28} = m2^8$ за модулем p = 11. Дістає елементи $r = g^7 = 2^7 = 7$ і $e = m \cdot 5^d = m \cdot 2^{4 \cdot 7} = 3 \cdot 9 = 5$ і висилає їх Бобу. Боб обчислює елемент $m' = e \cdot r^{p-1-4} = m \cdot 2^8 \cdot 2^{7 \cdot 6} = m \cdot 2^{10} = m = 3$.

Таким чином, обмін відбувся й далі робота йде за описаними протоколами.

Практична частина

Для реалізації протоколу було обрано мову програмування *Java* 11, оскільки вона є платформо-незалежною, а також зручною для використання. Вихідний код програми компілюється і конвертується у байткод, який може бути запущений на будь-якому пристрої, де встановлено віртуальну *Java*-машину (*JVM*) без необхідності перекомпіляції та внесення змін у початковий код. Одинадцята версія є версією з тривалою підтримкою (*LTS*), яка підтримуватиметься щонайменше до вересня 2026 р.

Оскільки в проєкті використовуються зовнішні бібліотеки, такі як фреймворк логування *Log4J* та клієнт для брокера повідом-лень *RabbitMQ*, було вирішено використовувати систему збірки *Maven* версії 3.8.1. *Maven* це фреймворк для автоматизації зборки проєкту, компіляції, створення дистрибутиву програми, генерації документації тощо. Інформація для зборки міститься у файлі *pom.xml*, який розташовано в корені проєкту.

Важливу роль у надійності протоколу відіграє вибір числа p, яке має бути великим простим числом. Тому в реалізації використовується тип *BigInteger* з пакету *java.math*, який підтримує цілочисельні значення в діапазоні від $-2^{Integer.MAX\_VALUE}$ до $2^{Integer.MAX\_VALUE}$. Значення *Integer.MAX_VALUE* є константою і дорівнює 2147483647. Крім того, цей клас надає готовий зручний функціонал операцій для роботи над числами в полі лишків F_p, такий як знаходження мультиплікативного оберненого до числа a, що використовуватиметься при реалізації алгоритму.

Реалізація модифікованого TSS-алгоритму в полі лишків F_p

Основою описаного алгоритму є розв'язання системи лінійних неоднорідних діофантових рівнянь (СЛНДР) у полі F_p. Для знаходження базису множини всіх розв'язків СЛНДР було реалізовано модифікований *TSS*-алгоритм, опис якого можна знайти у [2].

Реалізація протоколу обміну повідомленнями

Як уже було зазначено, для роботи протоколу необхідною є наявність системи лінійних виразів, представленої прямокутною матрицею A, та невироджених квадратних матриць B_i.

Побудова невиродженої квадратної матриці здійснюється у такий спосіб:

1. Будується верхньотрикутна (нижньотрикутна) матриця, жоден із діагональних елементів якої не дорівнює нулю

2. Над рядками матриці виконуються елементарні перетворення — перестановка двох рядків місцями, множення рядка на константу, відмінну від нуля, додавання до одного рядка іншого, помноженого на константу, відмінну від нуля.

Очевидно, що побудована в такий спосіб матриця буде невиродженою. Детермінант верхньотрикутної матриці визначається як добуток елементів головної діагоналі, а отже, він є відмінним від нуля і ранг такої матриці дорівнює розмірності самої матриці. Елементарні перетворення не змінюють рангу, тому ранг отри-

маної вихідної матриці дорівнює її розмірності, а отже, така матриця є невиродженою [8].

Для того, щоб побудувати систему розмірності $q \times n$, де $q \le n$, можна побудувати невироджену квадратну матрицю розмірності $q \times q$ у зазначений спосіб, а потім до отриманої матриці дописати (праворуч чи ліворуч від неї) матрицю, що складається з одиниць розмірності $q \times n - q$. Система рівнянь, що відповідає цій матриці, буде лінійно незалежною. У такий спосіб можна отримати матрицю A.

Оскільки для матриць B_i у процесі виконання алгоритму потрібно обраховувати обернені матриці, то для пришвидшення роботи алгоритму було вирішено будувати їх у вигляді $B_i = \begin{pmatrix} I_m & A & 0 \\ 0 & I_m & B \\ 0 & 0 & I_m \end{pmatrix}$, де I_m — одинична квадратна матриця розмірності $m \times m$, $m = n/3$, матриці A і B — невироджені квадратні матриці розмірності $m \times m$, які можна побудувати за описаним алгоритмом.

Наступним кроком є співставлення із символами вхідного тексту чисел із поля F_p. Для простоти було вирішено пронумерувати кожен символ абетки додатним числом із поля F_p, починаючи з нуля. Відображення між символами абетки та відповідним числом з поля F_p зберігається у двонаправленій мапі, представленій об'єктом класу *com.google.common.collect.HashBiMap*. Двонаправлена мапа — це мапа, яка гарантує унікальність і ключів, і значень, що в ній зберігаються. Така властивість дозволяє ввести поняття оберненої мапи, яка складається точно з тих самих пар елементів, де ключами виступають значення початкової мапи, а значеннями постають ключі. Вхідний алфавіт, а також нумерацію його символів наведено у Додатку А.

Для покращення стійкості алгоритму при кодуванні символів алфавіту можна застосовувати поняття ізоморфізму полів, однак цей спосіб буде розглянутий у наступних версіях представленого алгоритму.

Для передачі повідомлень використовується брокер повідомлень *RabbitMQ*, який працює за протоколом *AMQP* (*Advanced Message Queuing Protocol*) [9]. Брокер повідомлень — це програмне забезпечення, що дає змогу передавати повідомлення між додатками від відправника (постачальника) до одержувача (споживача). Є кілька широко використовуваних брокерів, таких як *RabbitMQ, Apache Kafka, Redis, Amazon SQS* тощо. У цьому проєкті було вирішено використати *RabbitMQ*, оскільки його встановлення та налаштування є доволі легким і зрозумілим. Крім того, *RabbitMQ* гарантує, що повідомлення будуть доправлені в тому ж порядку, в якому їх було відправлено, що є критично важливим для даної реалізації протоколу обміну повідомленнями [10].

```
rabbitmq.host=localhost
rabbitmq.port=5672
rabbitmq.username=username
rabbitmq.password=password
```

Рис. Файл властивостей

Постачальник передає повідомлення компоненті брокера, яка називається точкою обміну, до якої прив'язана одна або декілька черг. Обмінник ставить повідомлення у відповідну чергу, з якої його може отримати споживач. Отже, обмінник відіграє роль відправної точки брокера, тоді як черга є його точкою виходу. Для того, щоб підключитися до сервера *RabbitMQ*, потрібно вказати параметри з'єднання: назву хоста (або його *ip*-адреса), номер порту, ім'я користувача, пароль тощо. У даній реалізації ці параметри задаються у файлі властивостей *application.properties* як зображено на рисунку.

Під час запуску програми з боку відправника відкривається нове з'єднання, вказується назва точки обміну та прив'язаної до нього черги, в яку одержувачем буде надіслано матриці A та D. Після того, як їх буде отримано, відправник згідно з кроком 2 описаного протоколу, обчислить значення вектора d', яке так само надішле у чергу через обмінник. Тоді одержувач прочитає з черги вектор d' і виконає дії, описані у кроці 3.

Оскільки довжина повідомлення не завжди може співпадати з розмірністю матриць A і D, то повідомлення відправляється блоками довжини m, де m — кількість рядків матриці A. Якщо розмір блоку є меншим за число m, то він доповнюється порожніми символами до необхідної довжини. Для позначення кінця повідомлення відправник надсилає вектор d' довжини нуль і завершує роботу програми.

Налаштування та запуск програми

Для роботи програми необхідна наявність запущеного сервера брокера повідомлень *RabbitMQ*, параметри підключення до якого мають вказуватися у файлі властивостей *application.properties*, як було зазначено раніше. Зрозуміло, що для коректної роботи алгоритму відправник та отримувач мають бути підключені до того самого серверу *RabbitMQ*.

Після вказання всіх необхідних параметрів програму потрібно зібрати за допомогою команди *./mvn clean install*. В результаті цього з'явиться директорія *target*, яка й міститиме вихідний файл *message-exchange-protocol.jar*. Для того, щоб запустити програму, необхідно виконати команду *java — jar message-exchange-protocol.jar*<режим><*p*><*m*><*n*>, де аргумент режим отримує два значення: *sender* (відправник) або *receiver* (одержувач), p — це велике просте число, m *і* n — розмірність матриці A (кількість рівнянь у системі та кількість змінних).

Наприклад, для відправника команда може виглядати так: *java — jar message-exchange-protocol sender* 70054097 10 2. Вихідний код програми доступний за посиланням *https://github.com/JuliaNortman/MessageExchangeProtocol*.

Результати

У статті пропонується простий протокол обміну повідомленнями на основі систем лінійних діофантових рівнянь із цілими коефіцієнтами, розв'язання яких здійснюється або у скінченному полі лишків за модулем простого числа, або у множині натуральних чисел, а також спрощений протокол такого обміну. Наводяться приклади, які ілюструють роботу протоколів та обговорюються властивості їх надійності. Протоколи обміну інформацією на основі діофантових рівнянь поліноміального типу розглядалися також у роботі [6], а лінійного типу — в роботі [7].

Додаток А

Символ	Номер
a	0
b	1
c	2
d	3
e	4
f	5
g	6
h	7
i	8
j	9
k	10

Символ	Номер
3	29
4	30
5	31
6	32
7	33
8	34
9	35
A	36
B	37
C	38
D	39

Символ	Номер
W	58
X	59
Y	60
Z	61
.	62
,	63
?	64
/	65
!	66
@	67
#	68

Додаток А (*продовження*)

Символ	Номер
l	11
m	12
n	13
o	14
p	15
q	16
r	17
s	18
t	19
u	20
v	21
w	22
x	23
y	24
z	25
0	26
1	27
2	28

Символ	Номер
E	40
F	41
G	42
H	43
I	44
J	45
K	46
L	47
M	48
N	49
O	50
P	51
Q	52
R	53
S	54
T	55
U	56
V	57

Символ	Номер
$	69
;	70
{	71
}	72
[	73
]	74
%	75
^	76
:	77
&	78
*	79
(	80
)	81
_	82
-	83
+	84
=	85

ЛІТЕРАТУРА

1. *Hermann M., Juban L., Kolaitis P. G.* On the Complexity of Counting the Hilbert Basis of Linear Diophantine System. Springer Verlag. LNCS. 1999. № 1705. P. 13–32.
2. *Кривий С.Л.* Лінійні діофантові обмеження та їх застосування. К.: Інтерсервіс. 2021. 260 с.
3. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы построения и анализа (2-е изд.). Издательский дом «Вильямс». 2005. 290 с.
4. *Рябко Б.Я., Фионов А.Н.* Криптографические методы защиты информации. М: Горячая линия – Телеком. 2005. 229 с.
5. *Diffie W., Hellman M.E.* New direction in cryptography. IEEE Transaction on Information Theory. 1976, v. 22. P. 644–654.
6. *Berczes A., Lajos H., Hirete-Kohno N., Kovacs T.* A key exchange protocol based on Diophantine equations and *S*-integers. In JSIAM Letters. Dec. 2014. PP. 85–88.
7. *Kameswari P.A., Sriniasarao S.S., Belay A.* An Application of Linear Diophantine equations to Cryptography. Advanced in Mathematics: Scientific Journal. 2021, v. 10. N 6. PP. 2799–2806.
8. *Курош А.* Курс высшей алгебры. 2008. 432 с.
9. AMQP Advanced Message Queuing Protocol Specification [Електронний ресурс]. 0-9-1. 2008. URL: https://www.rabbitmq.com/resources/specs/amqp0-9-1.pdf.
10. *Videla A., Williams, J.* RabbitMQ in Action. Shelter Island: Manning Publications Co., 2012. 288 p.

Надійшла 01.02.2022

REFERENCES

1. *Hermann, M., Juban, L., Kolaitis, P.G.*, 1999. “On the Complexity of Counting the Hilbert Basis of Linear Diophantine System”. Springer Verlag. NCS. No. 1705, pp. 13–32.

2. *Kryvyi, S.L.*, 2021. Liniini diofantovi obmezhennia ta yikh zastosuvannia. K.: Interservis. 60 p. (In Ukrainian).
3. *Cormen, T., Leiserson, Ch., Rivest, R., Stein, C.*, 2005. Introduction to Algorithms (3d edition). 1290 p. (In Russian).
4. *Ryabko, B.Ya., Fionov, A.N.*, 2005. Kriptograficheskie metodyi zaschity iinformatsii. M: Goryachaya liniya – Telekom. 229 p. (In Russian).
5. *Diffie, W., Hellman, M.E.,* 1976. "New direction in cryptography". IEEE Transaction on Information Theory, v. 22, pp. 644–654.
6. *Berczes, A., Lajos, H., Hirete-Kohno, N., Kovacs, T.*, 2014. "A key exchange protocol based on Diophantine equations and S-integers". In JSIAM Letters. pp. 85–88.
7. *Kameswari, P.A., Sriniasarao, S.S., Belay, A.*, 2021. "An Application of Linear Diophantine equations to Crypto-graphy". Advanced in Mathematics: Scientific Journal, v. 10. N 6, pp. 2799–2806.
8. *Kurosh, A.*, 2008. Kurs vyisshey algebryi. 432 p. (In Russian).
9. AMQP Advanced Message Queuing Protocol Protocol Specification. 0-9-1. 2008. https://www.rabbitmq.com/resources/specs/amqp0-9-1.pdf.
10. *Videla A., Williams, J.,* 2012. RabbitMQ in Action. Shelter Island: Manning Publications Co., 288 p.

Received 01.02.2022

S.L. Kryvyi, Doctor habilit. of Physical and Mathematical Sciences, Department of Intelligent Software Systems, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, ave. Academician Glushkova, 4D, Kyiv, 03022, Ukraine,
sl.kryvoi@gmail.com
Yu.O. Nortman, M.S. in Software Engineering, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv,
ave. Academician Glushkova, 4D, Kyiv, 03022, Ukraine,
ynortman@gmail.com

A PROTOCOL FOR EXCHANGE INFORMATION ON THE BASE OF LINEAR FUNCTIONS AND OPERATORS

Introduction. Safety of human activity is required for almost every enterprise institution, organization, bank, etc. Therefore, it is extremely important to have a possibility to transform the information in such a way that it becomes inaccessible to the malicious user.

The article considers a simple messaging protocol between subscribers (Alice and Bob), based on the properties of linear functions and linear non-degenerate operators in vector spaces of finite dimension and systems of linear Diophantine equations over the set of natural numbers.

Purpose. The purpose of this article is to describe a protocol, based on linear function and operators properties, that allows to transmit the data in fast and secure way between two subscribers — sender and receiver.

Methods. The simple message exchange protocol is based on the properties of operators in linear space and the complexity of algorithms for solving systems of linear homogeneous Diophantine equations in the set of natural numbers.

Results. A simple message exchange method based on linear functions and linear non-degenerate operators in linear spaces of finite dimension and systems of linear Diophantine equa-tions over the set of natural numbers is proposed.

An application has been developed to securely transfer a message from a sender to a recipient.

The solution of these equations is carried out either in a finite field of surpluses modulo a prime number, or in the set of natural numbers and a simplified protocol for such an exchange.

Examples are given that illustrate the operation of protocols and the considered properties of their reliability.

Conclusion. The complexity of developed algorithm belongs to polynomial class O(n3) and complexity of decryption belongs to Exptime (#NP).

Keywords: *linear nonsingular operator, system of linear Diophantine equations, exchange information.*