



П. Ю. Грицюк, Ю. І. Грицюк

Національний університет "Львівська політехніка", м. Львів, Україна

МЕТОДИ ГЕНЕРУВАННЯ ПОЛІНОМІВ ЛЮКА ТА ОСОБЛИВОСТІ ЇХ ВИКОРИСТАННЯ ДЛЯ ШИФРУВАННЯ ДАНИХ

Розроблено методи генерування послідовностей поліномів Люка n -го степеня як основи для шифрування поточкових і блокових даних, що дає можливість ефективно передавати каналами зв'язку відповідні повідомлення різної величини. З'ясовано, що за останнє десятиліття надруковано значну кількість публікацій, в кожній з яких обґрунтовано різні підходи до генерування поліномів Люка та доведено доцільність їх використання для шифрування даних. Проте більшість досліджень стосується окремих процедур захисту даних, що в теорії та практиці криптографії трапляються вкрай рідко. Встановлено основну складність проблеми генерування послідовностей поліномів Люка n -го степеня, які є основою шифрування поточкових і блокових даних, що дасть можливість здійснювати ефективний їх захист. Наведено відомі способи подання чисел Фібоначчі та Люка, а також поліномів Люка, які можна застосувати у традиційному методі шифрування даних. Встановлено, що в послідовностях чисел Фібоначчі та Люка, де відношення двох послідовних елементів наближається до золотого перерізу, їхні члени є наближеннями цілих степенів золотого перерізу. Запропоновано метод матричного подання поліномів Люка та їх обернених еквівалентів, які можна застосувати у традиційному методі шифрування даних. Для цього необхідно перемножити спеціальну матрицю n -го порядку з відповідними коефіцієнтами на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Люка $L_n(x)$ відповідного степеня. Зазвичай, спеціальна матриця є нижньою трикутною матрицею, всі елементи головної діагоналі якої одиниці. Розроблено метод генерування поліноміальних матриць Люка n -го степеня та їх обернених матриць, елементами яких є поліноми Люка, які можна застосувати у традиційному методі шифрування даних. Оскільки поліноміальні обернені матриці використовують для розшифрування даних, то процедура їхнього генерування має мати загальний вигляд. Розроблено ПЗ, яке дає змогу генерувати поліноміальні матриці Люка n -го степеня та m -го порядку, а також їхні обернені поліноміальні матриці аналогічного степеня та порядку. За результатами виконаного дослідження зроблено висновки та надано відповідні рекомендації щодо їх практичного використання як основи для шифрування поточкових і блокових даних.

Ключові слова: числа Люка; рекурентна послідовність поліномів; золотий переріз; шифрування поточкових і блокових даних; поліноміальна матриця Люка; обернена поліноміальна матриця.

Вступ / Introduction

Числа Фібоначчі та Люка [22] є фундаментальною математичною основою, яка має важливе значення для сучасних інформаційних технологій (ІТ). Їх універсальність відчутна в різних ІТ-галузях – від розроблення програмного забезпечення до криптографії та фінансових технологій. Розуміння послідовностей таких чисел допомагає програмістам поглибити свої знання з організації структур даних і алгоритмів їх оброблення, покращити навички програмування чи тестування, а також розробляти дещо ефективніші програми для різних областей знань. Тому, виходячи з їх широкого застосування, кожен програміст повинен бути ознайомлений з концепцією генерування послідовностей чисел Фібоначчі та Люка, щоб краще їх розуміти та використовувати у своїй повсякденній роботі.

Ще в 2005 році український вчений О. Стахов у своїй роботі [48] запропонував так звану теорію шиф-

рування даних і коригування помилок p -числами Фібоначчі, метод дешифрування яких використовував певні властивості цих чисел, їх узагальнення та наближення до золотого перерізу [5]. Автор стверджував, що запропонований метод (роз)шифрування даних має здатність виправлення помилок 93,33 %. Проте, його обнадійливі заяви в багатьох дослідників [5, 7, 10, 38, 42] викликали деякі підозри насамперед з точки зору теорії інформації. Вони вважали, що потрібно провести дещо глибший і точніший аналіз результатів дослідження О. Стахова як щодо надмірності кодів, так і щодо їхньої можливості виявлення та виправлення помилок у зашифрованих повідомленнях.

Не секрет, що теорія шифрування даних упродовж багатьох десятиліть має важливе значення для інформаційно-комунікаційних технологій [29], внаслідок чого розроблено багато різних підходів до виявлення та виправлення можливих помилок, що виникають під час пе-

Інформація про авторів:

Грицюк Павло Юрійович, магістр, аспірант, кафедра автоматизованих систем управління. Email: pavlo.y.hrytsiuk@lpnu.ua;

<https://orcid.org/0009-0003-5409-2043>

Грицюк Юрій Іванович, д-р техн. наук, професор, кафедра програмного забезпечення. Email: yurii.i.hrytsiuk@lpnu.ua;

<https://orcid.org/0000-0001-8183-3466>

Цитування за ДСТУ: Грицюк П. Ю., Грицюк Ю. І. Методи генерування поліномів Люка та особливості їх використання для шифрування даних. Науковий вісник НЛТУ України. 2024, т. 34, № 8. С. 160–176.

Citation APA: Grytsiuk, P. Yu., & Hrytsiuk, Yu. I. (2024). Methods for generating Lucas polynomials and features of their use for data encryption. *Scientific Bulletin of UNFU*, 34(8), 160–176. <https://doi.org/10.36930/40340818>

редачі зашифрованих повідомлень [19]. Водночас, застосування сучасних методів шифрування даних з виправленням помилок під час розроблення криптографічних систем [21, 24, 27], стійких до квантових атак, зацікавило багатьох науковців до творчих пошуків і сприяло появі результатів нових досліджень. Проте, основний недолік багатьох криптографічних методів шифрування даних з виправленням помилок у тому, що вони використовують ключі великої розмірності з недостатньою коригувальною здатністю [25, 26, 28]. Тому розроблення нових підходів до (роз)шифрування даних з виявленням і виправленням помилок є важливою дослідною роботою, а для криптографічних систем проблема пошуку компактних ключів досі є відкритою.

Упродовж останніх десятиліть різні дослідники у своїх роботах [6, 11, 13, 16, 37, 47] навели особливості генерування поліномів Фібоначчі та Люка і похідних від них. Автори вважали, що поліноми Фібоначчі та Люка є природним розширенням їхніх відповідних послідовностей чисел, тому їм притаманні багато їхніх безпосередніх властивостей. Також вони наводять похідні цих поліномів у формі відповідних згорток, що дало їм змогу сформувати сімейство відповідних послідовностей. Ними також доведено багато співвідношень для похідних поліномів Фібоначчі та Люка. Послідовність поліномів Фібоначчі та золотий переріз з'явилися в багатьох галузях науки, в т.ч. фізиці високих енергій, криптографії та шифруванні даних [5, 13].

Проте, виконана чимала кількість досліджень [4, 17, 18, 30, 45, 51, 53] переважно стосується різних підходів до генерування поліномів Фібоначчі та Люка, а також їх використання для (роз)шифрування даних. Водночас, як окрема процедура такий підхід в практиці захисту та передачі даних трапляється вкрай рідко. Тому виникає потреба проведення додаткових досліджень щодо ефективних методів генерування поліномів Фібоначчі та Люка, а також встановлення певних особливостей їх використання для (роз)шифрування даних тільки як локального методу.

Об'єкт дослідження – генерування поліномів Люка як основи для шифрування даних.

Предмет дослідження – алгоритми і методи генерування послідовностей поліномів Люка n -го степеня, які стануть основою шифрування поточкових і блокових даних, що дасть можливість здійснювати ефективний їх захист.

Мета роботи – розробити методи генерування послідовностей поліномів Люка n -го степеня як основи шифрування поточкових і блокових даних, що дасть змогу ефективно передавати каналами зв'язку відповідні повідомлення різної величини.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- проаналізувати останні дослідження і публікації та встановити основну складність проблеми генерування послідовностей поліномів Люка n -го степеня, які стануть основою шифрування поточкових і блокових даних, що дасть можливість здійснювати ефективний їх захист;
- навести традиційні способи подання чисел Фібоначчі та Люка, а також поліномів Люка n -го степеня, що мають багато корисних властивостей, використати тільки ті із них, які можна застосувати у традиційному методі шифрування даних;
- запропонувати метод матричного подання поліномів Люка та їх обернених еквівалентів, які можна застосувати у

традиційному методі шифрування даних, навести конкретний приклад їх застосування;

- розробити метод генерування поліноміальних матриць Люка n -го степеня та їх обернених матриць, елементами яких є поліноми Люка, які можна застосувати у традиційному методі шифрування даних, навести конкретний приклад їх застосування;
- зробити висновки за результатами виконаного дослідження та надати відповідні рекомендації щодо їх практичного використання.

Аналіз останніх досліджень та публікацій. Послідовності чисел Фібоначчі та Люка є основою математичного збагачення дослідників певними знаннями та здобули популярність через можливість широкого використання [48]. Водночас, різні послідовності поліномів Фібоначчі та Люка хоча й трапляються інколи в науковій літературі, позаяк вони тісно пов'язані між собою [4], однак вони значно менше досліджені. З розвитком комп'ютерної техніки поліноми Фібоначчі та Люка почали застосовувати в різних областях знань [13], в т.ч. й для шифрування даних, насамперед матричним методом [26].

У роботі [33] розглянуто особливості генерування чисел Фібоначчі та Люка з різними додатками, де зосереджено вичерпний аналіз найцікавіших чисельних математичних компонент. Автор вважає, що числа Фібоначчі та Люка протягом століття цікавлять математиків-аматорів і професійних математиків. Їхня ж робота є першою спробою зібрати наявну історію та провести авторитетний аналіз цих відомих цілочисельних компонент разом із множиною захопливих програмних рішень, повних прикладів і цікавих вправ, які пропонують численні можливості для дослідження та експериментування. Числа Фібоначчі та Люка з додатками надають математикам велику кількість довідкового матеріалу в одному зручному зібранні та представляють глибокий і цікавий ресурс для ентузіастів будь-якого рівня та з будь-яким набутим досвідом.

У роботі [3] проаналізовано деякі властивості гаусових чисел Фібоначчі та Люка k -порядку. Автори визначили та вивчили числа Фібоначчі та Люка k -порядку з граничними умовами. Вони ідентифікували та підтвердили генеруючі функції, формули Біне, формули підсумовування, матричне подання гаусових чисел Фібоначчі k -порядку та деякі істотні зв'язки між гаусовими числами Фібоначчі k -порядку та числами Люка k -порядку, пов'язані зі звичайними числами Фібоначчі k -порядку.

У роботі [31] наведено деякі результати дослідження щодо непарних і парних індексованих послідовностей чисел Фібоначчі та Люка, доведено єдиність суми елементів їхніх скінченних підмножин. Показано, що непарні або парні індексовані послідовності чисел Фібоначчі та Люка є унікальними суперзростаючими послідовностями. Використовуючи встановлені властивості унікальності, авторами на прикладі показано новий метод криптографії, призначений для шифрування даних.

У роботі [54] розглянуто деякі тотожності узагальнених послідовностей чисел Фібоначчі та Люка. Спочатку автори ввели поняття узагальненої послідовності чисел Люка, де навели перелік елементарних співвідношень щодо узагальнених послідовностей чисел Фібоначчі та Люка. Потім вони навели узагальнення формул Біне для узагальнених послідовностей відповідних чисел, а також різні можливості їх застосування. Також дослідники вивели багато тотожностей і співвідношень

конгруенції для узагальнених послідовностей зазначених чисел за допомогою операторного методу.

У роботі [20] розроблено методи генерування послідовностей поліномів Фібоначчі n -го степеня як основи шифрування поточкових і блокових даних, що дає можливість ефективно передавати каналами зв'язку відповідні повідомлення різної величини. Запропоновано метод матричного подання поліномів Фібоначчі та їх обернених еквівалентів, які можна застосовувати у традиційному методі шифрування даних. Для цього необхідно перемножити спеціальну матрицю n -го порядку з відповідними коефіцієнтами на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Фібоначчі $F_n(x)$ відповідного степеня. Зазвичай, спеціальна матриця є нижньою трикутною матрицею, всі елементи головної діагоналі якої одиниці. Розроблено метод генерування поліноміальних матриць Фібоначчі n -го степеня та їх обернених матриць, елементами яких є поліноми Фібоначчі, які можна застосовувати у традиційному методі шифрування даних. Розроблено ПЗ, яке дає змогу генерувати поліноміальні матриці Фібоначчі n -го степеня та m -го порядку, а також їхні обернені поліноміальні матриці аналогічного степеня та порядку. За результатами виконаного дослідження зроблено висновки та надано відповідні рекомендації щодо їх практичного використання як основи для шифрування поточкових і блокових даних.

У роботі [35] розглянуто особливості узагальнення послідовності чисел Люка та пов'язаних з ними тотожностей. Автори намагалися узагальнити послідовність чисел Люка, генеруючи певну кількість послідовностей, члени яких отримані шляхом додавання двох останніх згенерованих членів попередньої послідовності. Послідовність чисел Люка отримана як окремий випадок генерування тільки однієї такої послідовності. Автори навели деякі результати, які можна розглядати як узагальнену форму очікуваних результатів, які справедливі для послідовності чисел Люка. Дослідники отримали тотожність Кассіні для цих узагальнених послідовностей чисел Люка. Водночас, у роботі [2] розглянуто узагальнені числа гіпер-Люка та особливості їх застосування. Автори навели деякі комбінаторні властивості нового узагальнення чисел гіпер-Люка для розширення визначника Кассіні. Також вони вивчали норми деяких циркулянтних і s -циркулянтних матриць, побудованих на підставі узагальнених чисел гіпер-Люка.

Як зазначено в роботах [8, 41], так звані p -числа Фібоначчі також можна узагальнити, враховуючи деякі лінійні рекурентні їхні послідовності, такі як p -числа Люка [34, 42] або Пелля-Люка [38], які дещо узагальнюють p -числа Фібоначчі та методи (роз)шифрування даних з їхнім використанням. Наприклад, автор роботи [40] запропонував використовувати рекурентні послідовності $F_n^{(p)}(x, y)$, застосовуючи поліноми Фібоначчі [11]. Подальші узагальнення результатів вирішення цієї проблеми можна знайти в роботах [6, 7, 13, 36].

У роботі [17] наведено деякі зауваги щодо узагальнених чисел Фібоначчі-Люка та узагальнених кватерніонів Фібоначчі-Люка. Автори запропонували породжувальні функції для узагальнених чисел Фібоначчі-Люка та для їхніх узагальнених кватерніонів, а також для них тотожність Кассіні та інші їхні цікаві властивості, різні можливості застосування цих елементів.

У роботі [45] наведено результати побудови дуально-узагальнених комплексних кватерніонів Фібоначчі

та Люка. Автори розглядали властивості як дуально-узагальненого комплексного числа, так і кватерніона. Також було отримано загальні рекурентні співвідношення для відповідних послідовностей кватерніонів, формули Біне, тотожності Тагіури (або Вайди), Гонсбергера, д'Оканя, Кассіні та Каталана. Наведено набір матричних подань цих спеціальних кватерніонів і виражено добуток дуально-узагальнених комплексних кватерніонів Фібоначчі та Люка у вигляді їхніх різних матричних подань.

У роботі [1] наведено деякі нові формули поліномів Люка за допомогою різних підходів. Автори вважають, що ці нові формули пов'язані з добре відомими симетричними поліномами Люка, є основною темою їхнього дослідження. Для встановлення цих формул вони використовували різні підходи. Для отримання деяких фундаментальних властивостей таких формул було використано матричний підхід до подання поліномів Люка. Зокрема, деякі рекурентні співвідношення та детермінантні форми вони визначали відповідними матрицями Гессенберга. Виведено та досліджено спряжені поліноми Люка та генеруючі функції. Дослідники вирішили декілька проблем зв'язку між поліномами Люка та іншими відомими симетричними та несиметричними ортогональними поліномами, такими як поліноми Чебишева першого та другого типів і їх зміщені відповідники. Ними доведено, що в коефіцієнтах зв'язку задіяно декілька гіпергеометричних функцій аргументного типу. Окрім цього, вони побудували нові формули для старших похідних від поліномів Люка через їх початкові поліноми, а також формули для повторних інтегралів від поліномів Люка.

У роботі [44] наведено особливості реалізації модифікованої криптографії з відкритим ключем на підставі узагальнених матриць Люка. Автори пропонують узагальнену матрицю Люка (рекурсивну матрицю вищого порядку), отриману з узагальнених послідовностей поліномів Фібоначчі. Вони отримали їхні алгебраїчні властивості, такі як пряме обернене обчислення, так і рекурсивний їхній характер тощо. Потім вони запропонували модифіковану криптографію з відкритим ключем, використовуючи для цього узагальнені матриці Люка як ключовий елемент, який оптимізує складність конструкції простору ключів. Окрім цього, автори встановили ключову домовленість для шифрування-розшифрування даних з комбінацією термінів узагальнених послідовностей Люка під операцією множення та отримання залишку.

У роботі [51] розглянуто нове застосування теорії шифрування числами Фібоначчі та Люка. Автори вважають, що алгоритми (роз)шифрування мають велике значення для покращення інформаційної безпеки, яка в останні роки стала серйозною проблемою багатьох фірм і організацій. Вони навели два нових алгоритми (роз)шифрування з використанням Q -матриць і R -матриць Фібоначчі. Розроблені моделі засновані на використанні заблокованих матриць повідомлень і шифруванні кожної такої матриці різними ключами. Автори вважають, що нові алгоритми не тільки підвищують безпеку інформації, але й мають високу коригувальну здатність.

У роботі [38] наведено узагальнену теорію шифрування даних на підставі p -чисел Пелля-Люка $S_p A^n$ та відповідний метод їх (роз)шифрування. Вони встанови-

ли співвідношення між елементами матриці шифрування для $p = 1$ та $i=1$, які дають змогу виявляти та виправляти помилки в зашифрованих повідомленнях. Під час обговорення результатів свого дослідження автори стверджують, що коригувальна здатність їхнього методу становить 93,33 % для $p = 1$, $i=1$, а для $p = 2$, $i=2$ – 99,80 %. Загалом, коригувальна здатність методу зростає зі збільшенням його значення.

У роботі [46] проаналізовано поліноми Фібоначчі та детерміновані тотожності, пов'язані з ними. Встановлено, що поліноми Фібоначчі та Люка відомі тим, що володіють чудовими та дивовижними властивостями, а також відповідними детермінованими тотожностями, які було детально описано. З'ясовано, що записи детермінованих тотожностей задовольняють рекурентні співвідношення поліномів Фібоначчі та поліномів Люка.

У роботі [47] наведено узагальнені детерміновані тотожності з поліномами Люка. Встановлено, що детерміновані тотожності відіграють значну роль у різних областях математики. Існують різні точки зору на вивчення таких тотожностей. Багато проблем щодо тотожностей послідовності чисел Фібоначчі та послідовності чисел Люка з'являлися в різних випусках журналу *Fibonacci Quarterly* (Щоквартальник Фібоначчі). Автори надають узагальнені детерміновані тотожності, що містять поліноми Люка та інші поліноми в різних порядках. Також наведено записи цих тотожностей, що задовольняють рекурентне співвідношення поліномів Люка та інших поліномів, таких як поліноми Фібоначчі, поліноми Чебишева, поліноми Пелла, поліноми Пелла-Люка, поліноми Вієта-Люка.

Отже, за результатами проведеного аналізу останніх досліджень та публікацій стосовно наявних підходів до генерування поліномів Фібоначчі та Люка, а також особливостей їх використання для шифрування даних було встановлено, що навіть за останнє десятиліття виконано багато різноманітних досліджень, в кожному з яких тією чи іншою мірою обґрунтовано різні підходи до генерування таких поліномів і доведено доцільність їх використання для шифрування даних. Водночас, як окрема процедура захисту поточкових і блокових даних поліномами Люка у теорії та практиці криптографії трапляються вкрай рідко. Тому проведення додаткових досліджень щодо ефективних методів генерування поліномів Люка, а також встановлення певних особливостей їх використання для шифрування поточкових і блокових даних тільки як локального методу є актуальною проблемою, яку й спробуємо частково вирішити в цьому дослідженні.

Результати дослідження та їх обговорення / Research results and their discussion

Числа Люка або ряд Люка – цілочисельна послідовність, названа на честь математика Франсуа Едуара Анатолія Люка (1842–1891), який вивчав як їхню послідовність, так і тісно пов'язані числа Фібоначчі [33]. Числа Люка та числа Фібоначчі утворюють доповняльні випадки послідовностей чисел Люка. Послідовність чисел Люка має аналогічне рекурентне співвідношення як і послідовність чисел Фібоначчі, де кожен доданок є сумою двох попередніх доданків, але з різними початковими значеннями. Це приводить до послідовності, де відношення послідовних двох доданків наближаються до золотого перерізу [5] і, фактично, самі члени є наб-

лиженнями цілих степенів золотого перерізу. Числа Люка також мають різноманітні взаємозв'язки з числами Фібоначчі. Наприклад, додавання будь-яких двох чисел Фібоначчі, розділених двома членами в їхній послідовності чисел, приводить до числа Люка між ними.

1. Подання чисел Фібоначчі та Люка. Послідовність чисел Фібоначчі визначають таким рекурентним співвідношенням [48]:

$$F_{n+1}(x) = xF_n(x) + F_{n-1}(x), \quad (1)$$

де: $F_0(x) = 0$, $F_1(x) = 1$ для $n \geq 2$. Послідовність чисел Фібоначчі для $x = 1$ має вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10916, ... Послідовність чисел Фібоначчі для від'ємних індексів визначають як $F_{-n}(x) = (-1)^{n-1}F_n(x)$ для $n \geq 1$.

Для генерування послідовності чисел Люка використовують таке рекурентне співвідношення з різними початковими значеннями [22]:

$$L_{n+1}(x) = xL_n(x) + L_{n-1}(x), \quad (2)$$

де $L_0(x) = 2$, $L_1(x) = x$ для $n \geq 2$ і $L_n(x) = L_{n+1}(x) + L_{n-1}(x)$ для $n \in \mathbb{Z}$. Послідовність чисел Люка для $x = 1$ має такий вигляд: 2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, 9349, 15127, 24476, 39603, 64079, 103682, 167761, 271443, ... Послідовність чисел Люка для від'ємних індексів визначають як $L_{-n}(x) = (-1)^n L_n(x)$ для $n \geq 1$.

Як окремі випадки послідовностей Фібоначчі, поліноми Люка задовольняють деяким тотожностям [4]:

$$F_{m+n}(x) = F_{m+1}(x)F_n(x) + F_m(x)F_{n-1}(x);$$

$$L_{m+n}(x) = L_m(x)L_n(x) - (-1)^n L_{m-n}(x);$$

$$F_{n+1}(x)F_{n-1}(x) - F_n(x)^2 = (-1)^n;$$

$$F_{2n}(x) = F_n(x)L_n(x).$$

2. Подання поліномів Люка. Числа Фібоначчі та Люка є популярною темою для математичного збагачення відповідними знаннями багатьох дослідників і популяризації цих чисел серед інших науковців [48]. Водночас, різні послідовності поліномів під назвами поліномів Фібоначчі та Люка трапляються в науковій літературі хоча й давно, позаяк вони тісно пов'язані між собою, проте вони не так широко та ґрунтовно досліджені, як відповідні послідовності аналогічних іменованих чисел. Зазначені такі поліноми з'являються в різних областях застосування, в тому числі й для шифрування даних, насамперед матричним методом.

У математиці поліноми Фібоначчі – це поліноміальна послідовність, яку можна розглядати як узагальнення чисел Фібоначчі [13]. Поліноми, утворені аналогічно із чисел Люка, називають поліномами Люка [47]. Їхні властивості практично аналогічні як і для поліномів Фібоначчі, однак тільки з деякими відмінностями, про які буде вказано дещо нижче.

Метод генерування поліномів Люка полягає у використанні рекурентного співвідношення (2), згідно з яким спочатку потрібно змінну x помножити послідовно на елементи даного полінома, потім потрібно згрупувати схожі доданки з елементами попереднього полінома, внаслідок чого отримаємо елементи наступного полінома. Наприклад, за вхідних поліномів $L_6(x) = x^6 + 6x^4 + 8x^2 + 2$ і $L_7(x) = x^7 + 7x^5 + 14x^3 + 7x$ наступний поліном матиме такий вигляд:

$$L_8(x) = xL_7(x) + L_6(x) = x(x^7 + 7x^5 + 14x^3 + 7x) + (x^6 + 6x^4 + 8x^2 + 2) = x^8 + 7x^6 + 14x^4 + 7x^2 + x^6 + 6x^4 + 8x^2 + 2 = x^8 + 8x^6 + 20x^4 + 16x^2 + 2.$$

Перші декілька поліномів Люка мають такий вигляд:

$$\begin{aligned}
L_0(x) &= 2; \\
L_1(x) &= x; \\
L_2(x) &= x^2 + 2; \\
L_3(x) &= x^3 + 3x; \\
L_4(x) &= x^4 + 4x^2 + 2; \\
L_5(x) &= x^5 + 5x^3 + 5x; \\
L_6(x) &= x^6 + 6x^4 + 9x^2 + 2; \\
L_7(x) &= x^7 + 7x^5 + 14x^3 + 7x; \\
L_8(x) &= x^8 + 8x^6 + 20x^4 + 16x^2 + 2; \\
L_9(x) &= x^9 + 9x^7 + 27x^5 + 30x^3 + 9x; \\
L_{10}(x) &= x^{10} + 10x^8 + 35x^6 + 50x^4 + 25x^2 + 2.
\end{aligned}$$

$$L_{0-10}(x) = \begin{pmatrix} 2 \\ x \\ x^2 + 2 \\ x^3 + 3x \\ x^4 + 4x^2 + 2 \\ x^5 + 5x^3 + 5x \\ x^6 + 6x^4 + 9x^2 + 2 \\ x^7 + 7x^5 + 14x^3 + 7x \\ x^8 + 8x^6 + 20x^4 + 16x^2 + 2 \\ x^9 + 9x^7 + 27x^5 + 30x^3 + 9x \\ x^{10} + 10x^8 + 35x^6 + 50x^4 + 25x^2 + 2 \end{pmatrix}.$$

Отже, послідовності поліномів Люка [4, 52] є природним розширенням відповідних послідовностей чисел Люка, тому багато їхніх властивостей допускають пряме їх доведення [16, 37, 54].

3. Матричне подання поліномів Люка. Часто в математичних розрахунках, особливо це стосується криптографічних методів захисту даних, доводиться мати справу з матричним поданням поліномів Люка для прямого ходу та їх оберненим виглядом – для зворотного. Для цього необхідно перемножити спеціальну матрицю n -го порядку з відповідними коефіцієнтами на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Люка $L_n(x)$ відповідного степеня. Зазвичай, спеціальна матриця є нижньою трикутною матрицею, всі елементи головної діагоналі якої є одиничними [13, 22, 37].

Для генерування поліномів Люка та їх обернених еквівалентів потрібно використати спеціальну матрицю n -го порядку з відповідними коефіцієнтами та поліном n -го степеня. Метод генерування поліномів Люка полягає у виконанні матричної дії множення спеціальної матриці n -го порядку на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Люка від 1-го до n -го степеня. Метод генерування обернених еквівалентів поліномів Люка полягає у знаходженні матриці, оберненої до спеціальної матриці n -го порядку, та виконанні матричної дії множення оберненої матриці n -го порядку на поліном n -го степеня, внаслідок чого отримаємо набір обернених поліномів Люка від 1-го до n -го степеня.

Для генерування поліномів Люка та їх обернених еквівалентів потрібно використати спеціальну матрицю n -го порядку з відповідними коефіцієнтами та поліном n -го степеня. Метод генерування поліномів Люка полягає у виконанні матричної дії множення спеціальної матриці n -го порядку на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Люка від 0-го до n -го степеня. Метод генерування обернених еквівалентів поліномів Люка полягає у знаходженні матриці, оберненої до спеціальної матриці n -го порядку, та виконанні матричної дії множення оберненої матриці n -го порядку на поліном n -го степеня, внаслідок чого отримаємо набір обернених поліномів Люка від 0-го до n -го степеня.

Математично це матиме такий вигляд:

$$\bar{L}_n(x) = \bar{A}_{n \times n} \times \bar{X}_n, \quad (3)$$

де: $\bar{L}_n(x)$ – набір поліномів Люка від 0-го до n -го степеня; $\bar{A}_{n \times n}$ – спеціальна матриця n -го порядку; $\bar{X}_n$ – набір елементів полінома n -го степеня.

Нижче наведено набір поліномів Люка $\bar{L}_n(x)$ від 0-го до 6-го степеня [1, ст. 5], а саме:

$$\bar{L}_{0-6}(x) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 5 & 0 & 5 & 0 & 1 & 0 \\ 2 & 0 & 9 & 0 & 6 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 2 \\ x \\ x^2 + 2 \\ x^3 + 3x \\ x^4 + 4x^2 + 2 \\ x^5 + 5x^3 + 5x \\ x^6 + 6x^4 + 9x^2 + 2 \end{pmatrix}. \quad (7)$$

Водночас, для отримання обернених поліномів Люка [1, ст. 65] потрібно перемножити матрицю, обернену до матриці коефіцієнтів, на елементи полінома, внаслідок чого отримаємо набір обернених поліномів Люка $\bar{L}_n^{-1}(x)$ відповідного степеня, а саме:

$$\begin{aligned}
\bar{L}_{0-6}^{-1}(x) &= \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 5 & 0 & 5 & 0 & 1 & 0 \\ 2 & 0 & 9 & 0 & 6 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \\
&= \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 1 & 0 & 0 & 0 \\ 3 & 0 & -4 & 0 & 1 & 0 & 0 \\ 0 & 10 & 0 & -5 & 0 & 1 & 0 \\ -10 & 0 & 15 & 0 & -6 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 1/2 \\ x \\ x^2 - 1 \\ x^3 - 3x \\ x^4 - 4x^2 + 3 \\ x^5 - 5x^3 + 10x \\ x^6 - 6x^4 + 15x^2 - 10 \end{pmatrix}. \quad (8)
\end{aligned}$$

Математично це матиме такий вигляд:

$$\bar{L}_n^{-1}(x) = \bar{A}_{n \times n}^{-1} \times \bar{X}_n, \quad (6)$$

де: $\bar{L}_n^{-1}(x)$ – набір обернених поліномів Люка від 0-го до n -го степеня; $\bar{A}_{n \times n}$ – матриця, до спеціальної матриці n -го порядку; $\bar{X}_n$ – елементи полінома n -го степеня. Оскільки елементи спеціальної матриці є цілі числа, розміщені нижче головної діагоналі, а її визначник становить 1 (одиниця), то обернену матрицю можна визначити звичайними методами, а її елементи також будуть цілими числами, в т.ч. й від'ємними.

Нижче наведено набір квадратів поліномів Люка $L_n^2(x)$ та їх обернених еквівалентів $L_n^{-2}(x)$ від 0-го до 6-го степеня, а саме:

$$L_{0-6}^2(x) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 5 & 0 & 5 & 0 & 1 & 0 \\ 2 & 0 & 9 & 0 & 6 & 0 & 1 \end{pmatrix}^2 \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 4 \\ x \\ x^2 + 6 \\ x^3 + 6x \\ x^4 + 8x^2 + 14 \\ x^5 + 10x^3 + 25x \\ x^6 + 12x^4 + 42x^2 + 36 \end{pmatrix}. \quad (9)$$

$$\begin{aligned}
&= \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 1 & 0 & 0 & 0 \\ 14 & 0 & 8 & 0 & 1 & 0 & 0 \\ 0 & 25 & 0 & 10 & 0 & 1 & 0 \\ 36 & 0 & 42 & 0 & 12 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 4 \\ x \\ x^2 + 6 \\ x^3 + 6x \\ x^4 + 8x^2 + 14 \\ x^5 + 10x^3 + 25x \\ x^6 + 12x^4 + 42x^2 + 36 \end{pmatrix}.
\end{aligned}$$

$$L_{0-6}^{-2}(x) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 5 & 0 & 5 & 0 & 1 & 0 \\ 2 & 0 & 9 & 0 & 6 & 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \quad (10)$$

$$= \begin{pmatrix} 1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & -6 & 0 & 1 & 0 & 0 & 0 \\ 17 & 0 & -8 & 0 & 1 & 0 & 0 \\ 2 & 0 & 35 & 0 & -10 & 0 & 1 \\ -48 & 0 & 54 & 0 & -12 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 1/4 \\ x \\ 2x^2 - 3 \\ 2 \\ x^3 - 6x \\ 2x^4 - 16x^2 + 17 \\ 2 \\ x^5 - 10x^3 + 35x \\ x^6 - 12x^4 + 54x^2 - 48 \end{pmatrix}.$$

Нижче наведено набір поліномів Люка $L_n^5(x)$, піднесених до 5-го степеня, та їх обернених еквівалентів $L_n^{-5}(x)$ від 0-го до 6-го степеня, а саме:

$$L_{0-6}^5(x) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 5 & 0 & 5 & 0 & 1 & 0 \\ 2 & 0 & 9 & 0 & 6 & 0 & 1 \end{pmatrix}^5 \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} \quad (11)$$

$$= \begin{pmatrix} 32 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 62 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 1 & 0 & 0 & 0 \\ 270 & 0 & 20 & 0 & 1 & 0 & 0 \\ 0 & 175 & 0 & 25 & 0 & 1 & 0 \\ 1610 & 0 & 285 & 0 & 30 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 32 \\ x \\ x^2 + 62 \\ x^3 + 15x \\ x^4 + 20x^2 + 270 \\ x^5 + 25x^3 + 175x \\ x^6 + 30x^4 + 285x^2 + 1610 \end{pmatrix}.$$

$$L_{0-6}^{-5}(x) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 5 & 0 & 5 & 0 & 1 & 0 \\ 2 & 0 & 9 & 0 & 6 & 0 & 1 \end{pmatrix}^{-5} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} \quad (12)$$

$$= \begin{pmatrix} 1/32 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -31 & 0 & 1 & 0 & 0 & 0 & 0 \\ 16 & -15 & 0 & 1 & 0 & 0 & 0 \\ 485 & 0 & -20 & 0 & 1 & 0 & 0 \\ 16 & 0 & 200 & 0 & -25 & 0 & 1 \\ -815 & 0 & 315 & 0 & -30 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{pmatrix} = \begin{pmatrix} 1/32 \\ x \\ 16x^2 - 31 \\ 16 \\ x^3 - 15x \\ 16x^4 - 320x^2 + 485 \\ 16 \\ x^5 - 25x^3 + 200x \\ 2x^6 - 60x^4 + 630x^2 - 815 \end{pmatrix}.$$

Отже, з огляду на наведене вище, запропоновано метод матричного подання поліномів Люка та їх обернених еквівалентів, які можна застосувати у традиційному методі шифрування даних. Для цього необхідно перемножити спеціальну матрицю n -го порядку з відповідними коефіцієнтами на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Люка $L_n(x)$ відповідного степеня. Зазвичай, спеціальна матриця є нижньою трикутною матрицею, всі елементи головної діагоналі якої є одиницями.

Приклад. Для розуміння зазначеного вище наведемо конкретний приклад шифрування поточкових даних з використанням полінома Люка.

Спробуємо зашифрувати таке повідомлення: $P = \{\text{Терпи, козаче, отаманом будеш.}\}$. З врахуванням пропусків між словами, одновимірний масив символів вхідного повідомлення матиме такий вигляд: $P' = \{\text{Терпи, козаче, отаманом будеш.}\}$, яке міститиме $K = 30$ символів. Для розуміння процесу шифрування повідомлення $P' = \{p_k, k = \overline{1, K}\}$ запишемо його у вигляді двовимірного масиву символів P , в якому буде $n = 5$ стовпців і $m = 6$ рядків, а символи повідомлення в цей масив заноситимемо рядками за такою формулою:

$$P = \{P_i = \{p_{ij} = p'_{(i-1)n+j}, j = \overline{1, n}; i = \overline{1, m}\}\}. \quad (13)$$

Для шифрування даних застосуємо поліном Люка 3-го степеня та його обернений еквівалент такого вигляду:

$$L_3(x) = x^3 + 3x; L_3^{-1}(x) = x^3 - 3x. \quad (14)$$

Нижче наведено результати шифрування та розшифрування зазначеного вище повідомлення P .

Прямий хід. Спочатку символи повідомлення з масиву P перетворюємо у числові коди відповідно до таблиці ASCII, а саме:

$$A = \{A_i = \{a_{ij} = \text{Ord}(p_{ij}), j = \overline{1, n}; i = \overline{1, m}\}\}, \quad (15)$$

де: $\text{Ord}(x)$ – функція, аргументом якої є символ, повертає його код в межах від 0 до 255. Нагадаємо, ASCII (англ. *American Standard Code for Information Interchange*) – американський стандартний код для обміну інформацією. ASCII – 8-бітне (однобайтне) кодування, призначене для подання десяткових цифр, латинського та національних алфавітів, знаків пунктуації та керуючих символів, містить $2^8 = 256$ (від 0 до 255) комбінацій кодів.

Застосовуючи поліном Люка $L_3(x) = x^3 + 3x$, перетворюємо числові коди з масиву (15) у зашифровані повідомлення, а саме

$$L = \{L_i = \{l_{ij} = a_{ij}^3 + 3a_{ij}, j = \overline{1, n}; i = \overline{1, m}\}\}. \quad (16)$$

Далі, перемноживши отримані значення за модулем $m = 256$, отримуємо зашифровані числові коди

$$L' = \{L'_i = \{l'_{ij} = l_{ij} \otimes m, j = \overline{1, n}; i = \overline{1, m}\}\}, \quad (17)$$

які знову ж таки перетворюємо у символи зашифрованого повідомлення за формулою:

$$A' = \{A'_i = \{a'_{ij} = \text{Chr}(l'_{ij}), j = \overline{1, n}; i = \overline{1, m}\}\}, \quad (18)$$

де: $\text{Chr}(x)$ – функція, аргументом якої є число від 0 до 255, повертає символ із введеним кодом згідно з таблицею ASCII. І, на завершення, символи зашифрованого повідомлення з двовимірного масиву A' заносимо рядками у одновимірний масив P' за такою формулою:

$$P' = \{p'_{(i-1)n+j} = a'_{ij}, j = \overline{1, n}; i = \overline{1, m}\}. \quad (19)$$

На цьому шифрування вхідного повідомлення P' завершено.

Зворотний хід. Спочатку символи зашифрованого повідомлення з масиву P' перенесемо у двовимірний масив символів A' , в якому буде $n = 5$ стовпців і $m = 6$ рядків, а символи повідомлення в цей масив заносимо рядками за такою формулою:

$$A' = \{A'_i = \{a'_{ij} = p'_{(i-1)n+j}, j = \overline{1, n}; i = \overline{1, m}\}\}. \quad (20)$$

Символи зашифрованого повідомлення з двовимірного масиву A' перетворюємо у числові коди відповідно до таблиці ASCII, а саме:

$$L' = \{L'_i = \{l'_{ij} = \text{Ord}(a'_{ij}), j = \overline{1, n}; i = \overline{1, m}\}\}. \quad (21)$$

Застосовуючи обернений поліном Люка 3-го степеня $L_3^{-1}(x) = x^3 - 3x$, перетворюємо числові коди з масиву (21) у зашифровані повідомлення, а саме

$$L'' = \{L''_i = \{l''_{ij} = l'_{ij}^3 - 3l'_{ij}, j = \overline{1, n}; i = \overline{1, m}\}\}. \quad (22)$$

Далі, перемноживши отримані значення за модулем $m = 256$, отримуємо розшифровані числові коди

$$A = \{A_i = \{a_{ij} = l''_{ij} \otimes m, j = \overline{1, n}; i = \overline{1, m}\}\}, \quad (23)$$

які знову ж таки перетворюємо у символи розшифрованого повідомлення за формулою:

$$P = \{P_i = \{p_{ij} = \text{Chr}(a_{ij}), j = \overline{1, n}; i = \overline{1, m}\}\}. \quad (24)$$

І, на завершення, символи розшифрованого повідомлення з двовимірною масиву P заносимо рядками у одновимірний масив P за такою формулою:

$$P = \{p_{(i-1) \cdot n + j} = p_{ij}, j = \overline{1, n}; i = \overline{1, m}\}. \quad (25)$$

Вхідне повідомлення:

$P = \{\text{Терпи_козаче_отаманом_будеш.}\}$

1. Прямий хід: шифрування повідомлення

Кількість символів	$m = 30$	шт.
Кількість стовпців	5	шт.
Кількість рядків	6,0	шт.

P

	1	2	3	4	5
0	Т	е	р	п	и
1	,	—	к	о	з
2	а	ч	е	,	—
3	о	т	а	м	а
4	н	о	м	—	б
5	у	д	е	ш	.

L

	1	2	3	4	5
0	9261630	12009676	13824720	13652636	12487864
1	85316	857660	12813606	13481986	12327084
2	11240096	15069964	12009676	85316	857660
3	13481986	14173214	11240096	13144964	11240096
4	13312764	13481986	13144964	857660	11391300
5	14349636	11853036	12009676	15253736	97474

A'

	1	2	3	4	5
0	-	Д	—	ї	
1	,	—	о	€	Г
2	О	е	Д	,	
3	€	;	О	а	О
4	к	€	а		Г
5	П	~	Д	,	@

-Д-ї
,_о€Г
ОєД'_
€;ОаО
к€а_Г
П~Д'@

Зашифроване повідомлення:

$P' = \{-Д-ї ,_о€ГОєД'_€;ОаОк€а_ГП ~Д'@\}$

2. Зворотний хід: розшифрування повідомлення

Вхідне повідомлення:

$P' = \{-Д-ї ,_о€ГОєД'_€;ОаОк€а_ГП ~Д'@\}$

A'

	1	2	3	4	5
0	-	Д	—	ї	
1	,	—	о	€	Г
2	О	е	Д	,	
3	€	;	О	а	О
4	к	€	а		Г
5	П	~	Д	,	@

Кількість символів	30
Кількість стовпців	5
Кількість рядків	6,0

L''

	1	2	3	4	5
0	26910	7528948	3374550	6967298	52
1	3111698	857090	1367298	2515048	2146302
2	8741198	12008302	7528948	3111698	857090
3	2515048	205202	8741198	11238752	8741198
4	12812202	2515048	11238752	857090	7414290
5	9198	1999998	7528948	3111698	261952

P

	1	2	3	4	5
0	Т	е	р	п	и
1	,	—	к	о	з
2	а	ч	е	,	—
3	о	т	а	м	а
4	н	о	м	—	б
5	у	д	е	ш	.

Терпи
,_коз
аче,_
отам
аном_б
удеш.

Розшифроване повідомлення:

$P = \{\text{Терпи_козаче_отаманом_будеш.}\}$

На цьому розшифрування зашифрованого повідомлення P' завершено.

Наведену методику шифрування та розшифрування вхідного повідомлення виконано в середовищі MS Excel, а отримані результати показано нижче.

A

	1	2	3	4	5
0	210	229	240	239	232
1	44	95	234	238	231
2	224	247	229	44	95
3	238	242	224	236	224
4	237	238	236	95	225
5	243	228	229	248	46

max= 248
min= 44
S= 6020

$L' = L \otimes m$

	1	2	3	4	5
0	30	196	150	191	4
1	146	95	111	136	129
2	206	229	196	146	95
3	136	59	206	224	206
4	234	136	224	95	195
5	21	126	196	146	64

max= 234
min= 4
S= 4328

L'

	1	2	3	4	5
0	30	196	150	191	4
1	146	95	111	136	129
2	206	229	196	146	95
3	136	59	206	224	206
4	234	136	224	95	195
5	21	126	196	146	64

max= 234
min= 4
S= 4328

$A = L'' \otimes m$

	1	2	3	4	5
0	210	229	240	239	232
1	44	95	234	238	231
2	224	247	229	44	95
3	238	242	224	236	224
4	237	238	236	95	225
5	243	228	229	248	46

max= 248
min= 44
S= 6020

Наведена вище методика шифрування даних з використанням поліномів Люка показала себе ефективною для шифрування поточкових даних тільки як локального методу, яку ми спробували частково вирішити.

3. Метод генерування матриць Люка n -го степеня, елементами яких є поліноми Люка. Головним завданням будь-якого алгоритму шифрування є ефективне перемішування та розсіювання даних. Для цього часто використовують матриці Люка.

Якщо $\bar{X}$ – квадратна матриця, де $\bar{X}^2 = x\bar{X} + \bar{I}$, то $\bar{X}^n = L_n(x)\bar{X} + L_{n-1}(x)\bar{I}, \forall n \in \mathbb{Z}$. Нехай $\bar{L}_2^0(x) = \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix}$ – по-

чаткова матриця Люка [44]. Тоді $\bar{L}_2^n(x)$ -матрицю Люка n -го степеня можна отримати шляхом множення $\bar{Q}_2^n(x)$ -матриці Фібоначчі n -го степеня на початкову матрицю Люка $\bar{L}_2^0(x)$, внаслідок чого отримаємо

$$\bar{L}_2^n(x) = \bar{Q}_2^n(x) \times \bar{L}_2^0(x) = \bar{L}_2^0(x) \times \bar{Q}_2^n(x) = \begin{bmatrix} L_{n+1}(x) & L_n(x) \\ L_n(x) & L_{n-1}(x) \end{bmatrix}, \forall n \in \mathbb{Z}. \quad (26)$$

Її визначник матиме такий вигляд $\det(\bar{L}_2^n(x)) = L_{n-1}(x)L_{n+1}(x) - L_n^2(x) = (-1)^{n+1}(x^2 + 4)$, $n \geq 1$, який ще називають тотожністю Кассіні [48].

З роботи [48] відомо, що $\bar{Q}_m^n(x)$ -матриці Фібоначчі 1-го степеня розміром 2×2 , 3×3 , 4×4 , 5×5 і $m \times m$ матимуть такий вигляд:

$$\begin{aligned} \bar{Q}_2^1(x) &= \begin{bmatrix} x & 1 \\ 1 & 0 \end{bmatrix}_{2 \times 2}; \bar{Q}_3^1(x) = \begin{bmatrix} x & 1 & 0 \\ 0 & x & 1 \\ 0 & 1 & 0 \end{bmatrix}_{3 \times 3}; \bar{Q}_4^1(x) = \begin{bmatrix} x & 1 & 0 & 0 \\ 0 & x & 1 & 0 \\ 0 & 0 & x & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{4 \times 4}; \\ \bar{Q}_m^1(x) &= \begin{bmatrix} x & 1 & 0 & 0 & 0 \\ 0 & x & 1 & 0 & 0 \\ 0 & 0 & x & 1 & 0 \\ 0 & 0 & 0 & x & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5}; \bar{Q}_m^n(x) = \begin{bmatrix} x & 1 & 0 & 0 & \dots & 0 \\ 0 & x & 1 & 0 & \dots & 0 \\ 0 & 0 & x & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & x & 1 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{m \times m}. \end{aligned} \quad (27)$$

З роботи [44, ст. 6] відомо, що початкові матриці Люка 2-го, 3-го, 4-го, 5-го та $m \times m$ -го порядків мають такий вигляд:

$$\begin{aligned} \bar{Q}_2^1(x) &= \begin{bmatrix} x & 1 \\ 1 & 0 \end{bmatrix}; \bar{L}_2^0(x) = \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix}; K_2(x) = x^2 + 4; \\ \bar{L}_2^1(x) &= \bar{Q}_2^1(x)\bar{L}_2^0(x) = \begin{bmatrix} x & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix} = \begin{bmatrix} x^2 + 2 & x \\ x & 2 \end{bmatrix}; \det(\bar{L}_2^1(x)) = x^2 + 4; \\ \bar{L}_2^2(x) &= \bar{Q}_2^2(x)\bar{L}_2^0(x) = \begin{bmatrix} x^2 + 1 & x \\ x & 1 \end{bmatrix} \times \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix} = \begin{bmatrix} x^3 + 3x & x^2 + 2 \\ x^2 + 2 & x \end{bmatrix}; \det(\bar{L}_2^2(x)) = -(x^2 + 4); \\ \bar{L}_2^3(x) &= \bar{Q}_2^3(x)\bar{L}_2^0(x) = \begin{bmatrix} x^3 + 2x & x^2 + 1 \\ x^2 + 1 & x \end{bmatrix} \times \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix} = \begin{bmatrix} x^4 + 4x^2 + 2 & x^3 + 3x \\ x^3 + 3x & x^2 + 2 \end{bmatrix}; \det(\bar{L}_2^3(x)) = x^2 + 4; \\ \bar{L}_2^4(x) &= \bar{Q}_2^4(x)\bar{L}_2^0(x) = \begin{bmatrix} x^4 + 3x^2 + 1 & x^3 + 2x \\ x^3 + 2x & x^2 + 1 \end{bmatrix} \times \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix} = \begin{bmatrix} x^5 + 5x^3 + 5x & x^4 + 4x^2 + 2 \\ x^4 + 4x^2 + 2 & x^3 + 3x \end{bmatrix}; \det(\bar{L}_2^4(x)) = -(x^2 + 4); \\ \bar{L}_2^5(x) &= \bar{Q}_2^5(x)\bar{L}_2^0(x) = \begin{bmatrix} x^5 + 4x^3 + 3x & x^4 + 3x^2 + 1 \\ x^4 + 3x^2 + 1 & x^3 + 2x \end{bmatrix} \times \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix} = \begin{bmatrix} x^6 + 6x^4 + 9x^2 + 2 & x^5 + 5x^3 + 5x \\ x^5 + 5x^3 + 5x & x^4 + 4x^2 + 2 \end{bmatrix}; \det(\bar{L}_2^5(x)) = x^2 + 4; \\ \bar{L}_2^6(x) &= \bar{Q}_2^6(x)\bar{L}_2^0(x) = \begin{bmatrix} x^6 + 5x^4 + 6x^2 + 1 & x^5 + 4x^3 + 3x \\ x^5 + 4x^3 + 3x & x^4 + 3x^2 + 1 \end{bmatrix} \times \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix} = \begin{bmatrix} x^7 + 7x^5 + 14x^3 + 7x & x^6 + 6x^4 + 9x^2 + 2 \\ x^6 + 6x^4 + 9x^2 + 2 & x^5 + 5x^3 + 5x \end{bmatrix}; \det(\bar{L}_2^6(x)) = -(x^2 + 4). \end{aligned} \quad (29)$$

Позаяк поліноміальні $\bar{L}_m^n(x)$ -матриці Люка можна використовувати для реалізації операцій шифрування даних, то аналогічні обернені матриці також потрібні для організації операцій їх розшифрування. Інтуїтивно зрозуміло, що ці так звані поліноміальні $\bar{L}_m^n(x)$ -матриці Люка мають бути оберненими до матриць шифрування даних, які прийнято називати *поліноміальними оберненими $\bar{L}_m^n(x)$ -матрицями Люка*. Подібно до поліноміальних $\bar{L}_2^n(x)$ -матриць Люка 2-го порядку, відповідні полі-

$$\bar{L}_2^0(x) = \begin{bmatrix} x & 2 \\ 2 & -x \end{bmatrix}_{2 \times 2}; \bar{L}_3^0(x) = \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix}_{3 \times 3};$$

$$\bar{L}_4^0(x) = \begin{bmatrix} 7x & 8 & 4 & 3 \\ 3 & 4x & 5 & 1 \\ 1 & 2 & 3x & 4 \\ 4 & -3 & -2 & -1x \end{bmatrix}_{4 \times 4}; \bar{L}_5^0(x) = \begin{bmatrix} 15x & 16 & 11 & 10 & 7 \\ 7 & 8x & 9 & 4 & 3 \\ 3 & 4 & 5x & 6 & 1 \\ 1 & 2 & 3 & 4x & 5 \\ 5 & -4 & -3 & -2 & -1x \end{bmatrix}_{5 \times 5}; \quad (28)$$

$$\bar{L}_m^0(x) = \begin{bmatrix} (2^{m-1}-1)x & 2^{m-1} & 2^{m-1}-k & \dots & 7(2^{m-4}) & 3(2^{m-3}) & 2^{m-2}-1 \\ 2^{m-2}-1 & (2^{m-2})x & 2^{m-2}+1 & \dots & 7(2^{m-5}) & 3(2^{m-4}) & 2^{m-3}-1 \\ 2^{m-3}-1 & 2^{m-3} & (2^{m-3}+1)x & \dots & 7(2^{m-6}) & 3(2^{m-5}) & 2^{m-4}-1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & \dots & m-2 & (m-1)x & m \\ m & 1-m & 2-m & \dots & -3 & -2 & -1x \end{bmatrix}.$$

Нижче наведено $\bar{L}_m^n(x)$ -матриці Люка від 1-го до 6-го степеня розміром 2×2 , 3×3 і 4×4 , утворені шляхом множення $\bar{Q}_m^n$ -матриці Фібоначчі n -го степеня, елементами яких є відповідні поліноми Фібоначчі, на початкову матрицю Люка $\bar{L}_m^0(x)$.

4. Механізм утворення $\bar{L}_m^n$ -матриць Люка n -го степеня та $m=2$ порядку. Продемонструємо механізм утворення $\bar{L}_2^n$ -матриць Люка від 1-го до 6-го степеня розміром 2×2 та відповідно обернених $\bar{L}_2^{-n}$ -матриць Люка. У подальшому матриці $\bar{L}_m^n(x)$ та $\bar{L}_2^{-n}(x)$ будемо називати *поліноміальними матрицями Люка*. З наведених нижче матричних виразів (29) видно, що елементами $\bar{L}_2^n$ -матриці Люка n -го степеня є відповідні поліноми Люка $(n+1)$ -го степеня.

З функції визначника [43] та набору матричних виразів (29) очевидно, що вони матимуть такий вигляд:

$$\det(\bar{L}_2^n(x)) = (-1)^{n+1}(x^2 + 4) = (-1)^{n+1}K_2(x).$$

В нашому випадку для $n = 2$ і 5 отримаємо визначники, які різняться між собою тільки знаком:

$$\begin{aligned} \det(\bar{L}_2^2(x)) &= (x^3 + 3x)x - (x^2 + 2)(x^2 + 2) = -(x^2 + 4); \\ \det(\bar{L}_2^5(x)) &= (x^6 + 6x^4 + 9x^2 + 2)(x^4 + 4x^2 + 2) - \\ &\quad -(x^5 + 5x^3 + 5x)(x^5 + 5x^3 + 5x) = x^2 + 4. \end{aligned}$$

номіальні обернені $\bar{L}_2^{-n}(x)$ -матриці Люка 2-го порядку мають мати також загальний вигляд. Наприклад, якщо поліноміальна матриця Люка 1-го степеня має такий вигляд $\bar{L}_2^1(x) = \begin{bmatrix} x^2 + 2 & x \\ x & 2 \end{bmatrix}$, то відповідна обернена матри-

ця матиме такий вигляд $\bar{L}_2^{-1}(x) = \frac{1}{x^2 + 4} \begin{bmatrix} 2 & -x \\ -x & x^2 + 2 \end{bmatrix}$. Тоді усі матричні вирази (29) легко можна поширити на загальний випадок подання поліноміальних обернених $\bar{L}_2^{-n}(x)$ -матриць Люка, внаслідок чого отримаємо:

$$\begin{aligned}
\bar{L}_2^{-1}(x) &= \frac{1}{x^2+4} \begin{bmatrix} 2 & -x \\ -x & x^2+2 \end{bmatrix}; \\
\bar{L}_2^{-2}(x) &= -\frac{1}{x^2+4} \begin{bmatrix} -x & -(x^2+2) \\ -(x^2+2) & x^3+3x \end{bmatrix}; \\
\bar{L}_2^{-3}(x) &= \frac{1}{x^2+4} \begin{bmatrix} x^2+2 & -(x^3+3x) \\ -(x^3+3x) & x^4+4x^2+2 \end{bmatrix}; \\
\bar{L}_2^{-4}(x) &= -\frac{1}{x^2+4} \begin{bmatrix} x^3+3x & -(x^4+4x^2+2) \\ -(x^4+4x^2+2) & x^5+5x^3+5x \end{bmatrix}; \\
\bar{L}_2^{-5}(x) &= \frac{1}{x^2+4} \begin{bmatrix} x^4+4x^2+2 & -(x^5+5x^3+5x) \\ -(x^5+5x^3+5x) & x^6+6x^4+9x^2+2 \end{bmatrix}; \\
\bar{L}_2^{-6}(x) &= -\frac{1}{x^2+4} \begin{bmatrix} x^5+5x^3+5x & -(x^6+6x^4+9x^2+2) \\ -(x^6+6x^4+9x^2+2) & x^7+7x^5+14x^3+7x \end{bmatrix}.
\end{aligned} \quad (30)$$

Оскільки матричний вираз $\bar{L}_2^n(x) \times \bar{L}_2^{-n}(x) = \bar{I}_{2 \times 2}$, де $\bar{I}_{2 \times 2}$ – одинична матриця 2-го порядку, можна виконати для будь-якого значення n , то в нашому випадку для $n=3$ і $m=2$ отримаємо:

$$\begin{aligned}
\bar{L}_2^3(x) \times \bar{L}_2^{-3}(x) &= \bar{I}_{2 \times 2} \Rightarrow \\
\Rightarrow \begin{bmatrix} x^4+4x^2+2 & x^3+3x \\ x^3+3x & x^2+2 \end{bmatrix} \times \begin{bmatrix} x^2+2 & -x^3-3x \\ -x^3-3x & x^4+4x^2+2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (31)
\end{aligned}$$

5. Механізм утворення $\bar{L}_m^n$ -матриць Люка n -го степеня та $m=3$ порядку. Продемонструємо механізм утворення поліноміальних $\bar{L}_3^n$ -матриць Люка від 1-го до 6-го степеня розміром 3×3 . З наведеного набору матричних виразів (32) видно, що елементами $\bar{L}_3^n$ -матриці

$$\begin{aligned}
\bar{Q}_3^0(x) &= \begin{bmatrix} x & 1 & 0 \\ 0 & x & 1 \\ 0 & 1 & 0 \end{bmatrix}; \bar{L}_3^0(x) = \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix}; K_3(x) = 6x^3 - 16x - 34; \\
\bar{L}_3^1(x) &= \bar{Q}_3^1(x) \bar{L}_3^0(x) = \begin{bmatrix} x & 1 & 0 \\ 0 & x & 1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix} = \begin{bmatrix} 3x^2+1 & 6x & x+3 \\ x+3 & 2x^2-2 & 2x \\ 1 & 2x & 3 \end{bmatrix}; \det(\bar{L}_3^1(x)) = xK_3(x); \\
\bar{L}_3^2(x) &= \bar{Q}_3^2(x) \bar{L}_3^1(x) = \begin{bmatrix} x^2 & 2x & 1 \\ 0 & x^2+1 & x \\ 0 & x & 1 \end{bmatrix} \times \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix} = \begin{bmatrix} 3x^3+2x+3 & 8x^2-2 & x^2+5x \\ x^2+3x+1 & 2x^3 & 2x^2+3 \\ x+3 & 2x^2-2 & 2x \end{bmatrix}; \det(\bar{L}_3^2(x)) = -x^2K_3(x); \\
\bar{L}_3^3(x) &= \bar{Q}_3^3(x) \bar{L}_3^2(x) = \begin{bmatrix} x^3 & 3x^2+1 & 2x \\ 0 & x^3+2x & x^2+1 \\ 0 & x^2+1 & x \end{bmatrix} \times \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix} = \begin{bmatrix} 3x^4+3x^2+6x+1 & 10x^3-2x & x^3+7x^2+3 \\ x^3+3x^2+2x+3 & 2x^4+2x^2-2 & 2x^3+5x \\ x^2+3x+1 & 2x^3 & 2x^2+3 \end{bmatrix}; \det(\bar{L}_3^3(x)) = x^3K_3(x); \quad (32) \\
\bar{L}_3^4(x) &= \bar{Q}_3^4(x) \bar{L}_3^3(x) = \begin{bmatrix} x^4 & 4x^3+3x & 3x^2+1 \\ 0 & x^4+3x^2+1 & x^3+2x \\ 0 & x^3+2x & x^2+1 \end{bmatrix} \times \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix} = \begin{bmatrix} 3x^5+4x^3+9x^2+3x+3 & 12x^4-2 & x^4+9x^3+8x \\ x^4+3x^3+3x^2+6x+1 & 2x^5+4x^3-2x & 2x^4+7x^2+3 \\ x^3+3x^2+2x+3 & 2x^4+2x^2-2 & 2x^3+5x \end{bmatrix}; \det(\bar{L}_3^4(x)) = -x^4K_3(x); \\
\bar{L}_3^5(x) &= \bar{Q}_3^5(x) \bar{L}_3^4(x) = \begin{bmatrix} x^5 & 5x^4+6x^2+1 & 4x^3+3x \\ 0 & x^5+4x^3+3x & x^4+3x^2+1 \\ 0 & x^4+3x^2+1 & x^3+2x \end{bmatrix} \times \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix} = \\
&= \begin{bmatrix} 3x^6+5x^4+12x^3+6x^2+9x+1 & 14x^5+4x^3-4x & x^5+11x^4+15x^2+3 \\ x^5+4x^3+3x^2+9x^2+3x+3 & 2x^6+6x^4-2 & 2x^5+9x^3+8x \\ x^4+3x^3+3x^2+6x+1 & 2x^5+4x^3-2x & 2x^4+7x^2+3 \end{bmatrix}; \det(\bar{L}_3^5(x)) = x^5K_3(x); \\
\bar{L}_3^6(x) &= \bar{Q}_3^6(x) \bar{L}_3^5(x) = \begin{bmatrix} x^6 & 6x^5+10x^3+4x & 5x^4+6x^2+1 \\ 0 & x^6+5x^4+6x^2+1 & x^5+4x^3+3x \\ 0 & x^5+4x^3+3x & x^4+3x^2+1 \end{bmatrix} \times \begin{bmatrix} 3x & 4 & 1 \\ 1 & 2x & 3 \\ 3 & -2 & -1x \end{bmatrix} = \\
&= \begin{bmatrix} 3x^7+6x^5+15x^4+10x^3+18x^2+4x+3 & 16x^6+10x^4-4x^2-2 & x^6+13x^5+24x^3+11x \\ x^6+5x^4+3x^5+12x^3+6x^2+9x+1 & 2x^7+8x^5+4x^3-4x & 2x^6+11x^4+15x^2+3 \\ x^5+3x^4+4x^3+9x^2+3x+3 & 2x^6+6x^4-2 & 2x^5+9x^3+8x \end{bmatrix}; \det(\bar{L}_3^6(x)) = -x^6K_3(x). \\
\det(\bar{L}_3^1(x)) &= \det \begin{bmatrix} 3x^2+1 & 6x & x+3 \\ x+3 & 2x^2-2 & 2x \\ 1 & 2x & 3 \end{bmatrix} = (3x^2+1) \cdot \det \begin{bmatrix} 2x^2-2 & 2x \\ 2x & 3 \end{bmatrix} - 6 \cdot \det \begin{bmatrix} x+3 & 2x \\ 1 & 3 \end{bmatrix} + (x+3) \cdot \det \begin{bmatrix} x+3 & 2x^2-2 \\ 1 & 2x \end{bmatrix} = \\
&= (3x^2+1) \cdot (2x^2-6) - 6 \cdot (x+9) + (x+3) \cdot (6x+2) = 6x^4 - 16x^2 - 34x = x(6x^3 - 16x - 34); \\
\det(\bar{L}_3^2(x)) &= \det \begin{bmatrix} 3x^3+2x+3 & 8x^2-2 & x^2+5x \\ x^2+3x+1 & 2x^3 & 2x^2+3 \\ x+3 & 2x^2-2 & 2x \end{bmatrix} = (3x^3+2x+3) \cdot \det \begin{bmatrix} 2x^3 & 2x^2+3 \\ 2x^2-2 & 2x \end{bmatrix} - (8x^2-2) \cdot \det \begin{bmatrix} x^2+3x+1 & 2x^2+3 \\ x+3 & 2x \end{bmatrix} + \\
&+ (x^2+5x) \cdot \det \begin{bmatrix} x^2+3x+1 & 2x^3 \\ x+3 & 2x^2-2 \end{bmatrix} = (3x^3+2x+3) \cdot (-2x^2+6) - (8x^2-2) \cdot (-x-9) + (x^2+5x) \cdot (-6x-2) = \\
&= -6x^5 + 16x^3 + 34x^2 = -x^2(6x^3 - 16x - 34).
\end{aligned} \quad (33)$$

Люка n -го степеня є відповідні поліноми $(n+1)$ -го степеня, тому її й називають поліноміальною матрицею.

З функції визначника [43] та набору матричних виразів (32) очевидно, що вони матимуть такий вигляд:

$$\det(\bar{L}_3^n(x)) = (-1)^{n+1} x^n (6x^3 - 16x - 34) = (-1)^{n+1} x^n K_3(x).$$

В нашому випадку для $n=1$ і 2 отримаємо відповідні визначники такого вигляду (33).

Для поліноміальних $\bar{L}_3^n(x)$ -матриць Люка 3-го порядку відповідні поліноміальні обернені $\bar{L}_3^{-n}(x)$ -матриці Люка 3-го порядку матимуть також загальний вигляд. Матричні вирази (32) можна поширити на загальний випадок подання поліноміальних обернених матриць Люка, внаслідок чого отримаємо матричні вирази (34).

Для обчислення поліноміальних обернених матриць було використано відповідні математичні методи, один із яких метод Монтанте (англ. *Montante Method*) [39]. Це метод лінійної алгебри для розв'язання системи лінійних рівнянь, знаходження обернених матриць та визначників. Його головна особливість – працює, використовуючи винятково цілочисельну арифметику для цілочисельних матриць, що дає змогу отримувати точні результати в комп'ютерних реалізаціях.

Покажемо, що матричний вираз $\bar{L}_3^n(x) \times \bar{L}_3^{-n}(x) = \bar{I}_{3 \times 3}$, де $\bar{I}_{3 \times 3}$ – одинична матриця 3-го порядку, можна виконати для будь-якого значення n , наприклад $n=1, 2$ і 3 , внаслідок чого отримаємо такі матричні вирази (35).

$$\begin{aligned}
M_3(x) &= 3x^3 - 8x - 17; \\
\bar{L}_3^{-1}(x) &= \frac{1}{M_3(x)} \begin{bmatrix} \frac{x^2-3}{-x-9} & \frac{x-6}{9x-1} & \frac{-x^3+3x^2+x+3}{-6x^3+x^2+4x+9} \\ \frac{2x}{3x+1} & \frac{-3x^2+2}{3x^4-5x^2-9x-1} & \frac{2x}{3x^4-5x^2-9x-1} \end{bmatrix}; \\
\bar{L}_3^{-2}(x) &= \frac{1}{M_3(x)} \begin{bmatrix} \frac{x^2-3}{-x-9} & \frac{-x^3+3x^2+x+3}{-6x^3+x^2+4x+9} & \frac{x^5-3x^4-10x^2+3}{6x^5-x^4+5x^3-10x^2+x+9} \\ \frac{2x^2}{3x+1} & \frac{2x}{3x^4-5x^2-9x-1} & \frac{2x^2}{-3x^6+2x^4+9x^3+3x^2-3x-1} \end{bmatrix}; \\
\bar{L}_3^{-3}(x) &= \frac{1}{M_3(x)} \begin{bmatrix} \frac{x^2-3}{-x-9} & \frac{x^3}{6x^5-x^4+5x^3-10x^2+x+9} & \frac{x^2}{-6x^7+x^6-11x^5+11x^4+3x^3+x+9} \\ \frac{2x^3}{3x+1} & \frac{2x^2}{-3x^6+2x^4+9x^3+3x^2-3x-1} & \frac{2x^3}{3x^8+x^6-9x^5-8x^4-6x^3-3x-1} \end{bmatrix}; \\
\bar{L}_3^{-4}(x) &= \frac{1}{M_3(x)} \begin{bmatrix} \frac{x^2-3}{-x-9} & \frac{x^4}{-6x^7+x^6-11x^5+11x^4+3x^3+x+9} & \frac{x^3}{6x^9-x^8+17x^7-12x^6+2x^5-10x^4+x+9} \\ \frac{2x^4}{3x+1} & \frac{2x^3}{3x^8+x^6-9x^5-8x^4-6x^3-3x-1} & \frac{2x^4}{-3x^{10}-4x^8+9x^7+10x^6+15x^5+3x^4-3x-1} \end{bmatrix}; \\
\bar{L}_3^{-5}(x) &= \frac{1}{M_3(x)} \begin{bmatrix} \frac{x^2-3}{-x-9} & \frac{x^5}{6x^9-x^8+17x^7-12x^6+2x^5-10x^4+x+9} & \frac{x^4}{-6x^{11}+x^{10}-23x^9+13x^8-13x^7+21x^6+3x^5+x+9} \\ \frac{2x^5}{3x+1} & \frac{2x^4}{-3x^{10}-4x^8+9x^7+10x^6+15x^5+3x^4-3x-1} & \frac{2x^5}{3x^{12}+7x^{10}-9x^9-9x^8-24x^7-11x^6-6x^5-3x-1} \end{bmatrix}; \\
\bar{L}_3^{-6}(x) &= \frac{1}{M_3(x)} \begin{bmatrix} \frac{x^2-3}{-x-9} & \frac{x^6}{-6x^{11}+x^{10}-23x^9+13x^8-13x^7+21x^6+3x^5+x+9} & \frac{x^5}{6x^{13}-x^{12}+29x^{11}-14x^{10}+30x^9-33x^8-x^7-10x^6+x+9} \\ \frac{2x^6}{3x+1} & \frac{2x^5}{3x^{12}+7x^{10}-9x^9-9x^8-24x^7-11x^6-6x^5-3x-1} & \frac{2x^6}{-3x^{14}-10x^{12}+9x^{11}+5x^{10}+33x^9+21x^8+21x^7+31x^6-3x-1} \end{bmatrix}.
\end{aligned} \tag{34}$$

$$\begin{aligned}
\bar{L}_3(x) \times \bar{L}_3^{-1}(x) &= \bar{I}_{3 \times 3} \Rightarrow \begin{bmatrix} 3x^2+1 & 6x & x+3 \\ x+3 & 2x^2-2 & 2x \\ 1 & 2x & 3 \end{bmatrix} \times \begin{bmatrix} \frac{x^2-3}{x(3x^3-8x-17)} & \frac{x-6}{3x^3-8x-17} & \frac{-x^3+3x^2+x+3}{x(3x^3-8x-17)} \\ \frac{-x-9}{2x(3x^3-8x-17)} & \frac{9x-1}{2(3x^3-8x-17)} & \frac{-6x^3+x^2+4x+9}{2x(3x^3-8x-17)} \\ \frac{2x}{3x+1} & \frac{-3x^2+2}{3x^4-5x^2-9x-1} & \frac{2x}{3x^4-5x^2-9x-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \\
\bar{L}_3(x) \times \bar{L}_3^{-2}(x) &= \bar{I}_{3 \times 3} \Rightarrow \begin{bmatrix} 3x^3+2x+3 & 8x^2-2 & x^2+5x \\ x^2+3x+1 & 2x^3 & 2x^2+3 \\ x+3 & 2x^2-2 & 2x \end{bmatrix} \times \begin{bmatrix} \frac{x^2-3}{x^2(3x^3-8x-17)} & \frac{-x^3+3x^2+x+3}{x(3x^3-8x-17)} & \frac{x^5-3x^4-10x^2+3}{x^2(3x^3-8x-17)} \\ \frac{-x-9}{-6x^3+x^2+4x+9} & \frac{9x-1}{6x^5-x^4+5x^3-10x^2+x+9} & \frac{-6x^7+x^6-11x^5+11x^4+3x^3+x+9}{6x^9-x^8+17x^7-12x^6+2x^5-10x^4+x+9} \\ \frac{2x^2}{3x+1} & \frac{2x}{3x^4-5x^2-9x-1} & \frac{2x^2}{-3x^6+2x^4+9x^3+3x^2-3x-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \\
\bar{L}_3(x) \times \bar{L}_3^{-3}(x) &= \bar{I}_{3 \times 3} \Rightarrow \\
\Rightarrow \begin{bmatrix} 3x^4+3x^2+6x+1 & 10x^3-2x & x^3+7x^2+3 \\ x^3+3x^2+2x+3 & 2x^4+2x^2-2 & 2x^3+5x \\ x^2+3x+1 & 2x^3 & 2x^2+3 \end{bmatrix} \times \begin{bmatrix} \frac{x^2-3}{x^3(3x^3-8x-17)} & \frac{x^5-3x^4-10x^2+3}{x^2(3x^3-8x-17)} & \frac{-x^7+3x^6-x^5+13x^4+x^3-x^2+3}{x^3(3x^3-8x-17)} \\ \frac{-x-9}{6x^5-x^4+5x^3-10x^2+x+9} & \frac{9x-1}{-6x^7+x^6-11x^5+11x^4+3x^3+x+9} & \frac{-6x^3+x^2+4x+9}{6x^9-x^8+17x^7-12x^6+2x^5-10x^4+x+9} \\ \frac{2x^3}{3x+1} & \frac{2x^2}{-3x^6+2x^4+9x^3+3x^2-3x-1} & \frac{2x^3}{3x^8+x^6-9x^5-8x^4-6x^3-3x-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{35}$$

6. Механізм утворення $\bar{L}_m^n$ -матриць Люка n -го степеня та $m=4$ порядку. Продемонструємо механізм утворення поліноміальних $\bar{L}_4^n$ -матриць Люка від 1-го до 7-го степеня розміром 4×4 . З наведеного набору матричних виразів (36) видно, що елементами $\bar{L}_4^n$ -матриці Люка n -го степеня є відповідні поліноми $(n+1)$ -го степеня.

З функції визначника [43] та набору матричних виразів (36) очевидно, що вони матимуть такий вигляд:

$$\begin{aligned}
\det(\bar{L}_4^n(x)) &= (-1)^{n+1} x^{2n} (84x^4 - 301x^2 + 265x + 515) = \\
&= (-1)^{n+1} x^{2n} K_4(x).
\end{aligned}$$

В нашому випадку для $n = 1$ і 2 отримаємо відповідні визначники такого вигляду (37).

Для обчислення відповідних визначників було використано математичні методи, один із яких метод елімінації Гауса (англ. *Gaussian Elimination*) [39]. Це алгоритм розв'язування систем лінійних алгебраїчних рівнянь, під яким розуміють деяку послідовність операцій, що виконують над відповідною матрицею коефіцієнтів, для приведення її до трикутного вигляду, з наступним вираженням базисних змінних через небазисні. Його використовують для обчислення визначника матриці, а також для обернення невідродженої квадратної матриці.

$$\begin{aligned}
\overline{\overline{Q}}_4(x) &= \begin{bmatrix} x & 1 & 0 & 0 \\ 0 & x & 1 & 0 \\ 0 & 0 & x & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \overline{\overline{L}}_4(x) = \begin{bmatrix} 7x & 8 & 4 & 3 \\ 3 & 4x & 5 & 1 \\ 1 & 2 & 3x & 4 \\ 4 & -3 & -2 & -1x \end{bmatrix}; K_4(x) = 84x^4 - 301x^2 + 265x + 515; \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x & 1 & 0 & 0 \\ 0 & x & 1 & 0 \\ 0 & 0 & x & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 7x^2+3 & 12x & 4x+5 & 3x+1 \\ 3x+1 & 4x^2+2 & 8x & x+4 \\ x+4 & 2x-3 & 3x^2-2 & 3x \\ 1 & 2 & 3x & 4 \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = x^2 K_4(x); \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x^2 & 2x & 1 & 0 \\ 0 & x^2 & 2x & 1 \\ 0 & 0 & x^2+1 & x \\ 0 & 0 & x & 1 \end{bmatrix} \times \begin{bmatrix} 7x^3+6x+1 & 16x^2+2 & 4x^2+13x & 3x^2+2x+4 \\ 3x^2+2x+4 & 4x^3+4x-3 & 11x^2-2 & x^2+7x \\ x^2+4x+1 & -3x+2x^2+2 & 3x^3+x & 3x^2+4 \\ x+4 & 2x-3 & 3x^2-2 & 3x \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = -x^4 K_4(x); \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x^3 & 3x^2 & 3x & 1 \\ 0 & x^3 & 3x^2+1 & 2x \\ 0 & 0 & x^3+2x & x^2+1 \\ 0 & 0 & x^2+1 & x \end{bmatrix} \times \begin{bmatrix} 7x^4+9x^2+3x+4 & 20x^3+6x-3 & 4x^3+24x^2-2 & 3x^3+3x^2+11x \\ 3x^3+3x^2+8x+1 & 4x^4+6x^2-6x+2 & 14x^3-x & x^3+10x^2+4 \\ x^3+4x^2+2x+4 & 2x^3-3x^2+4x+3 & 3x^4+4x^2-2 & 3x^3+7x \\ x^2+4x+1 & -3x+2x^2+2 & 3x^3+x & 3x^2+4 \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = -x^6 K_4(x); \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x^4 & 4x^3 & 6x^2+1 & 3x \\ 0 & x^4 & 4x^3+3x & 3x^2+1 \\ 0 & 0 & x^4+3x^2+1 & x^3+2x \\ 0 & 0 & x^3+2x & x^2+1 \end{bmatrix} \times \begin{bmatrix} 7x^5+12x^3+6x^2+12x+1 & 24x^4+12x^2+2-9x & 4x^4+38x^3-3x & 3x^4+4x^3+21x^2+4 \\ 3x^4+4x^3+12x^2+3x+4 & 4x^5+8x^3-9x^2+6x+3 & 17x^4+3x^2-2 & x^4+13x^3+11x \\ x^4+4x^3+3x^2+8x+1 & 2x^4+6x^2+2-3x^3+6x & 3x^5+7x^3-x & 3x^4+10x^2+4 \\ x^3+4x^2+2x+4 & 2x^3-3x^2+4x+3 & 3x^4+4x^2-2 & 3x^3+7x \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = -x^8 K_4(x); \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x^5 & 5x^4 & 10x^3+4x & 6x^2+1 \\ 0 & x^5 & 5x^4+6x^2+1 & 4x^3+3x \\ 0 & 0 & x^5+4x^3+3x & x^4+3x^2+1 \\ 0 & 0 & x^4+3x^2+1 & x^3+2x \end{bmatrix} \times \begin{bmatrix} 7x^6+15x^4+10x^3+24x^2+4x+4 & 28x^5+20x^3+8x-18x^2-3 & 4x^5+55x^4-2 & 3x^5+5x^4+34x^3+15x \\ 3x^5+5x^4+6x^2+1 & 4x^6+10x^4-12x^3+12x^2+9x+2 & 20x^5+10x^3-3x & x^5+16x^4+21x^2+4 \\ x^5+4x^3+3x+4x^4+12x^2+4 & 2x^5-3x^4+8x^3+9x^2+6x+3 & 3x^6+10x^4+3x^2-2 & 3x^5+13x^3+11x \\ x^4+4x^3+3x^2+8x+1 & 2x^4-3x^3+6x^2+6x+2 & 3x^5+7x^3-x & 3x^4+10x^2+4 \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = x^{10} K_4(x); \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x^6 & 6x^5 & 15x^4+10x^2+1 & 10x^3+4x \\ 0 & x^6 & 6x^5+10x^2+4x & 5x^4+6x^2+1 \\ 0 & 0 & x^6+5x^4+6x^2+1 & x^5+4x^3+3x \\ 0 & 0 & x^5+4x^3+3x & x^4+3x^2+1 \end{bmatrix} \times \begin{bmatrix} 7x^7+18x^5+15x^4+40x^3+10x^2+16x+1 & 32x^6+30x^4-30x^3+20x^2-12x+2 & 4x^6+75x^5+10x^3-5x & 3x^6+6x^5+50x^4+36x^2+4 \\ 3x^6+6x^5+20x^4+10x^3+24x^2+4x+4 & 4x^7+2x^5-15x^4+20x^3+18x^2+8x+3 & 23x^6+20x^4-2 & x^6+19x^5+34x^3+15x \\ x^6+4x^5+5x^4+16x^3+6x^2+12x+1 & 2x^6-3x^5+10x^4+12x^3+12x^2+9x+2 & 3x^7+13x^5+10x^3-3x & 3x^6+16x^4+21x^2+4 \\ x^5+4x^4+4x^3+12x^2+3x+4 & 2x^5-3x^4+8x^3+9x^2+6x+3 & 3x^6+10x^4+3x^2-2 & 3x^5+13x^3+11x \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = -x^{12} K_4(x); \\
\overline{\overline{L}}_4(x) &= \overline{\overline{Q}}_4(x) \overline{\overline{L}}_4(x) = \begin{bmatrix} x^7 & 7x^6 & 21x^5+20x^3+5x & 15x^4+10x^2+1 \\ 0 & x^7 & 7x^6+15x^4+10x^2+1 & 6x^5+10x^3+4x \\ 0 & 0 & x^7+6x^5+10x^3+4x & x^6+5x^4+6x^2+1 \\ 0 & 0 & x^6+5x^4+6x^2+1 & x^5+4x^3+3x \end{bmatrix} \times \begin{bmatrix} 7x^8+21x^6+21x^5+60x^4+20x^3+40x^2+5x+4 & 36x^7+42x^5-45x^4+40x^3-30x^2+10x-3 & 4x^7+98x^6+30x^4-5x^2-2 & 3x^7+7x^6+69x^5+70x^3+19x \\ 3x^7+7x^6+24x^5+15x^4+40x^3+40x^2+16x+1 & 4x^8+14x^6-18x^5+30x^4-30x^3+20x^2-12x+2 & 26x^7+33x^5+10x^3-5x & x^7+22x^6+50x^4+36x^2+4 \\ x^7+4x^6+6x^5+20x^4+10x^3+24x^2+4x+4 & 2x^7-3x^6+12x^5-15x^4+20x^3-18x^2+8x-3 & 3x^8+16x^6+20x^4-2 & 3x^7+19x^5+34x^3+15x \\ x^6+4x^5+5x^4+16x^3+6x^2+12x+1 & 2x^6-3x^5+10x^4-12x^3+12x^2-9x+2 & 3x^7+13x^5+10x^3-3x & 3x^6+16x^4+21x^2+4 \end{bmatrix}; \det(\overline{\overline{L}}_4(x)) = x^{14} K_4(x).
\end{aligned}$$

(36)

$$\det(\bar{L}_4^1(x)) = \det \begin{bmatrix} 7x^2+3 & 12x & 4x+5 & 3x+1 \\ 3x+1 & 4x^2+2 & 8x & x+4 \\ x+4 & 2x-3 & 3x^2-2 & 3x \\ 1 & 2 & 3x & 4 \end{bmatrix} = (7x^2+3) \cdot \det \begin{bmatrix} 4x^2+2 & 8x & x+4 \\ 2x-3 & 3x^2-2 & 3x \\ 2 & 3x & 4 \end{bmatrix} - 12x \cdot \det \begin{bmatrix} 3x+1 & 8x & x+4 \\ x+4 & 3x^2-2 & 3x \\ 1 & 3x & 4 \end{bmatrix} +$$

$$+ (4x+5) \cdot \det \begin{bmatrix} 3x+1 & 4x^2+2 & x+4 \\ x+4 & 2x-3 & 3x \\ 1 & 2 & 4 \end{bmatrix} - (3x+1) \cdot \det \begin{bmatrix} 3x+1 & 4x^2+2 & 8x \\ x+4 & 2x-3 & 3x^2-2 \\ 1 & 2 & 3x \end{bmatrix} =$$

$$= (7x^2+3) \cdot (12x^4 - 51x^2 + 64x) - 12x \cdot (9x^3 + 7x^2 - 102x) + (4x+5) \cdot (-4x^3 - 58x^2 - 25x) -$$

$$- (3x+1) \cdot (-48x^3 - 35x^2 + 67x) = 84x^6 - 301x^4 + 265x^3 + 515x^2 = x^2(84x^4 - 301x^2 + 265x + 515); \quad (37)$$

$$\det(\bar{L}_4^2(x)) = \det \begin{bmatrix} 7x^3+6x+1 & 16x^2+2 & 4x^2+13x & 3x^2+2x+4 \\ 3x^2+2x+4 & 4x^3+4x-3 & 11x^2-2 & x^2+7x \\ x^2+4x+1 & -3x+2x^2+2 & 3x^3+x & 3x^2+4 \\ x+4 & 2x-3 & 3x^2-2 & 3x \end{bmatrix} = -84x^8 + 301x^6 - 265x^5 - 515x^4 = -x^4(84x^4 - 301x^2 + 265x + 515).$$

Для поліноміальних $\bar{L}_4^n(x)$ -матриць Люка 4-го порядку відповідні поліноміальні обернені $\bar{L}_4^{-n}(x)$ -матриці Люка 4-го порядку матимуть загальний вигляд. Тому матричні вирази (36) можна легко поширити на загальний випадок подання поліноміальних обернених мат-

риць Люка, внаслідок чого отримаємо такий набір математичних виразів (38).

Зрозуміло, матричний вираз $\bar{L}_4^n(x) \times \bar{L}_4^{-n}(x) = \bar{I}_{4 \times 4}$, де $\bar{I}_{4 \times 4}$ – одинична матриця $m=4$ -го порядку, можна виконати для будь-якого значення n , наприклад, для $n = 1$, внаслідок чого отримаємо такі матричні вирази:

$$\bar{L}_4^1(x) \times \bar{L}_4^{-1}(x) = \bar{I}_{4 \times 4} \Rightarrow \begin{bmatrix} 7x^2+3 & 12x & 4x+5 & 3x+1 \\ 3x+1 & 4x^2+2 & 8x & x+4 \\ x+4 & 2x-3 & 3x^2-2 & 3x \\ 1 & 2 & 3x & 4 \end{bmatrix} \times$$

$$\begin{bmatrix} \frac{12x^3-51x+64}{x(84x^4-301x^2+265x+515)} & \frac{-36x^3+35x^2+55x-64}{x^2(84x^4-301x^2+265x+515)} & \frac{36x^2-88x+138}{(84x^4-301x^2+265x+515)} & \frac{-36x^5+72x^4-38x^3-84x^2-55x+64}{x^2(84x^4-301x^2+265x+515)} \\ \frac{-9x^2-7x+102}{x(84x^4-301x^2+265x+515)} & \frac{21x^4-15x^2-51x-102}{x^2(84x^4-301x^2+265x+515)} & \frac{21x^2-167x+59}{(84x^4-301x^2+265x+515)} & \frac{-21x^5+111x^4-33x^3-47x^2+51x+102}{x^2(84x^4-301x^2+265x+515)} \\ \frac{-4x^2-58x-25}{x(84x^4-301x^2+265x+515)} & \frac{-14x^3+96x^2+153x+25}{x^2(84x^4-301x^2+265x+515)} & \frac{112x^2-26x-70}{(84x^4-301x^2+265x+515)} & \frac{-84x^5+26x^4+87x^3-101x^2-153x-25}{x^2(84x^4-301x^2+265x+515)} \\ \frac{48x^2+35x-67}{x(84x^4-301x^2+265x+515)} & \frac{-63x^3-116x^2-7x+67}{x^2(84x^4-301x^2+265x+515)} & \frac{-84x^3+158x-64}{84x^4-301x^2+265x+515} & \frac{84x^6-214x^4+168x^3+288x^2+7x-67}{x^2(84x^4-301x^2+265x+515)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (39)$$

Отже, розроблено метод генерування поліноміальних $\bar{L}_m^n(x)$ -матриць Люка n -го степеня та m -го порядку, а також їх обернених матриць, елементами яких є поліноми Люка n -го степеня. Оскільки поліноміальні матриці Люка можна використовувати для реалізації операції шифрування даних, то аналогічні поліноміальні обернені $\bar{L}_m^{-n}(x)$ -матриці також потрібні для процесу розшифрування даних, генерування яких має мати загальний вигляд. Розроблено ПЗ, яке дає змогу генерувати поліноміальні матриці $\bar{L}_m^n(x)$ -Люка n -го степеня та m -го порядку, а також їхні обернені поліноміальні матриці аналогічного степеня та порядку.

Під час розроблення ПЗ автори дослідження зіткнулися з декількома проблемами чисто математичного характеру. Справа в тому, що навіть під час генерування поліноміальних $\bar{L}_m^n(x)$ -матриць Люка $n = 6$ -го степеня та $m=2$ -го порядку отримуємо поліноми Люка 7-го степеня, утворені внаслідок множення рядка на стовпець, де знаходяться поліноми Люка дещо нижчих степенів. Для їх правильного подання доводиться спочатку перемножувати відповідні поліноми Люка нижчих степенів, потім отриманий результат треба додати і спростити. Інша проблема, це обчислення визначників матриць, починаючи з 4-го та вищих порядків, де доводиться використовувати спеціальні математичні методи, наприклад метод Монтанте (алгоритм Барейсса) (англ. *Montante method* – *Bareiss algorithm*), метод елімінації Гауса (англ. *Gaussian elimination*) чи формулу Лейбніца (англ. *Leibniz formula*).

Наступна математична проблема, це обчислення оберненої матриці, для вирішення якої також доводиться використовувати відповідні математичні методи, такі

як метод Монтанте (алгоритм Барейсса), метод елімінації Гауса-Джордана (англ. *Gauss-Jordan elimination*) та за допомогою ад'югрованої матриці (англ. *Adjugate matrix*). Всі ці методи та алгоритми детально описано в мережі Інтернет, тому в цьому дослідженні не будемо на них зосереджувати увагу. І остання проблема – уважність запису відповідних матриць, елементами яких є поліноми Люка, насамперед тих, які стосуються вищих порядків і степенів їх піднесення. Адже запис не того степеню змінної, коефіцієнта при ній чи його знаку – подальша робота даремна.

5. Матричний метод шифрування даних поліномами Люка. Далі наведено особливості реалізації матричного методу шифрування даних *поліномами Люка*. Передбачимо, що ця процедура практично нічим не відрізнятиметься від матричного методу шифрування даних *поліномами Фібоначчі* [13, 20, 22, 37]. Щоб використовувати цей метод, початкове повідомлення подамо у вигляді квадратної $\bar{M}$ -матриці m -го порядку, яку називають *матрицею повідомлення*. Немає обмежень щодо такого подання (тобто, можна й у вигляді прямокутної матриці), тому користувачеві залишається тільки визначати розташування елементів у матриці – рядками чи стовпцями. Наприклад, повідомлення "Поліномів" можна подати матрицею 3-го порядку, заповнюючи її поелементно рядками:

$$\bar{M}_3 = \begin{bmatrix} \Pi & o & l \\ i & n & o \\ m & i & v \end{bmatrix} = \begin{bmatrix} 207 & 238 & 235 \\ 179 & 237 & 238 \\ 236 & 179 & 226 \end{bmatrix}.$$

Після узгодження цілого числа $x \neq 0$ і цілого числа n між відправником і одержувачем $\bar{L}_3^n(x)$ -матрицю шифрування даних отримують з виразу (32). Потім цю матрицю потрібно помножити справа на матрицю повідом-

[illegible]

лення $\bar{M}$, щоб отримати зашифроване повідомлення у вигляді матриці $\bar{E}$. Наприклад, для $m = 3$ і $n = 3$ маємо

$$\bar{E} = \bar{L}_3(x) \times \bar{M}_3 = \begin{bmatrix} 3x^2+1 & 6x & x+3 \\ x+3 & 2x^2-2 & 2x \\ 1 & 2x & 3 \end{bmatrix} \times \begin{bmatrix} 207 & 238 & 235 \\ 179 & 237 & 238 \\ 236 & 179 & 226 \end{bmatrix} = \begin{bmatrix} 621x^2+1310x+915 & 714x^2+1601x+775 & 705x^2+1654x+913 \\ 358x^2+679x+263 & 474x^2+596x+240 & 476x^2+687x+229 \\ 358x+915 & 474x+775 & 476x+913 \end{bmatrix}.$$

Елементи матриці зашифрованого повідомлення $\bar{E}$ надсилають рядками в порядку $e_1, e_2, \dots, e_9$ каналом зв'язку, за якими слідує окремо значення визначника поліноміальної $\bar{L}_3(x)$ -матриці Люка, а саме:

$$\det(\bar{L}_3(x)) = \det \begin{bmatrix} 3x^2+1 & 6x & x+3 \\ x+3 & 2x^2-2 & 2x \\ 1 & 2x & 3 \end{bmatrix} = 6x^4 - 16x^2 - 34x.$$

Припускаючи, що передана послідовність чисел отримана без помилок, тоді, шляхом множення оберненої $\bar{L}_3^{-1}(x)$ -матриці на матрицю зашифрованого повідомлення $\bar{E}$, отримують початкову матрицю повідомлення:

$$\begin{aligned} \bar{M} &= \bar{L}_3^{-1}(x) \times \bar{E} = \\ &= \begin{bmatrix} \frac{x^2-3}{3x^4-8x^2-17x} & \frac{x-6}{3x^3-8x-17} & \frac{-x^3+3x^2+x+3}{3x^4-8x^2-17x} \\ \frac{-x-9}{6x^4-16x^2-34x} & \frac{9x-1}{6x^3-16x-34} & \frac{-6x^3+x^2+4x+9}{6x^4-16x^2-34x} \\ \frac{3x+1}{3x^4-8x^2-17x} & \frac{-3x^2+2}{3x^3-8x-17} & \frac{3x^4-5x^2-9x-1}{3x^4-8x^2-17x} \end{bmatrix} \times \\ &\times \begin{bmatrix} 621x^2+1310x+915 & 714x^2+1601x+775 & 705x^2+1654x+913 \\ 358x^2+679x+263 & 474x^2+596x+240 & 476x^2+687x+229 \\ 358x+915 & 474x+775 & 476x+913 \end{bmatrix} = \\ &= \begin{bmatrix} 207 & 238 & 235 \\ 179 & 237 & 238 \\ 236 & 179 & 226 \end{bmatrix}. \end{aligned}$$

З огляду на наведене вище, можна констатувати факт того, що запропоновано матричний метод шифрування даних на підставі поліномів Люка n -го степеня над алфавітом $\{0, 1, \dots, 255\}$. Йдеться про те, що для цілих чисел $m \geq 2$, $x \geq 1$ і $n \geq 1$, а також використаної поліноміальної $\bar{L}_m^n(x)$ -матриці Люка можна шифрувати блокові дані розміром $m \times m$. Кожне таке вхідне блочне повідомлення є $\bar{M}$ -матрицею розміром $m \times m$, яку можна перетворити в зашифроване повідомлення, використовуючи такий матричний вираз $\bar{E} = \bar{M} \times \bar{L}_m^n(x)$. Щоб розшифрувати це повідомлення, потрібно поліноміальну обернену $\bar{L}_m^{-n}(x)$ -матрицю Люка помножити на зашифроване повідомлення з використанням такого матричного виразу $\bar{M} = \bar{L}_m^{-n}(x) \times \bar{E}$.

Отже, показана можливість вирішення проблеми генерування послідовностей поліномів Люка n -го степеня, які є основою шифрування поточкових і блокових даних, що дає можливість здійснювати ефективний їх захист. Наведено відомі способи подання поліномів Люка n -го степеня, а також запропоновано метод матричного їх подання та обернених еквівалентів, використання яких показано на конкретному прикладі шифрування даних. Розроблено метод генерування поліноміальних $\bar{L}_m^n(x)$ -матриць Люка n -го степеня та m -го порядку, елементами яких є відповідні поліноми Люка, та їх обернених еквівалентів, особливості застосування яких показана на прикладі традиційного методу шифрування даних.

Обговорення результатів дослідження. Числа Фібоначчі та Люка є популярною темою для математичного збагачення дослідників і їхньої популяризації серед них [48]. Водночас, різні послідовності поліномів під назвами поліномів Люка трапляються в науковій літературі хоча і давно, позаяк вони тісно пов'язані між собою, проте вони не так широко досліджені, як відповідні послідовності чисел Люка. Зазначені поліноми з'являються в різних областях застосування, в тому числі й для шифрування даних, насамперед матричним методом. У математиці поліноми Люка [30, 47] – це поліноміальна послідовність, яку можна розглядати як узагальнення чисел Люка [2, 10, 16, 27, 45, 47].

У роботі [18] наведено тотожності Фібоначчі та Люка, отримані з матриць Тепліца–Гессенберга. Автори розглянули визначники для деяких сімейств матриць Тепліца–Гессенберга, що мають різні переклади чисел Фібоначчі та Люка для ненульових елементів. Формули цих визначників можна також переписати як тотожності, що містять суми добутків чисел Фібоначчі та Люка, а також їхніх мультиноміальних коефіцієнтів. У дослідженні наведено комбінаторні докази кількох визначників, які використовують знакозмінні інволюції та визначення визначника як суми зі знаком над симетричною групою. Це призвело до загального узагальнення формул Фібоначчі та Люка в термінах так званих чисел Гібоначчі.

У роботі [34] наведено особливості генерування p -матриці Люка, попередньо вивчаючи p -числа Фібоначчі та Люка. Було введено p -матрицю Люка та супутні матриці для сум p -чисел Фібоначчі та Люка, щоб отримати деякі їхні цікаві детерміновані тотожності. Водночас, у роботі [42] наведено узагальнену теорію шифрування даних з використанням p -чисел Люка. Якщо в роботі [34] тільки наведено p -матрицю Люка $R_p Q_p^n$, елементами якої є p -числа Люка, то в цій роботі розроблено новий метод (роз)шифрування даних на підставі p -матриці Люка $R_p Q_p^n$. Також автори встановили зв'язки між елементами матриці шифрування даних, з'ясували особливості виявлення та виправлення помилок у зашифрованих повідомленнях. Під час обговорення результатів свого дослідження автори стверджують, що коригувальна здатність запропонованого ними методу для $p=1$ становить 93,33 %, а для $p=2$ – 99,80 % і зростає зі збільшенням цього значення.

У роботі [48] розроблено матрицю Фібоначчі, наведено узагальнення формули Кассіні та нову теорію шифрування даних. Автори розглядали новий клас квадратних матриць розміром $(p+1) \times (p+1)$, елементами яких були p -числа Фібоначчі ($p = 0, 1, 2, 3, \dots$), а їхній визначник становить ± 1 . Ця унікальна властивість матриць Фібоначчі приводить до узагальнення формули Кассіні. Запропонований оригінальний метод (роз)шифрування даних числами Фібоначчі впливає з аналогічних матриць n -порядку. На відміну від класичних надлишкових кодів основною особливістю запропонованого методу є те, що він дає змогу коригувати зашифровані елементи матриці, які теоретично можуть бути необмеженими цілими числами. Для найпростішого випадку коригувальна здатність цього методу становить 93,33 %, що значно перевищує відомі коригувальні можливості інших методів.

У роботі [14] розроблено новий метод шифрування даних на підставі поліномів Фібоначчі з високою швид-

кістю їх генерування. Для цілих чисел m, n і $x \geq 1$ можна згенерувати квадратну $Q_{2m}^n(x)$ -матрицю n -го степеня $2m$ -го порядку, елементами якої є поліноми Фібоначчі, і відповідну обернену $Q_{2m}^n(x)$ -матрицю для їх дешифрування [25, 26, 28]. Також показано, що швидкість (роз)шифрування даних за допомогою цих матриць значно вища, ніж з оригінальними матрицями на підставі звичайних чисел Фібоначчі.

У роботі [41] наведено привабливі, як на перший погляд, показники надійності теорії шифрування даних поліномами Фібоначчі. Розроблено новий клас квадратних $Q_{pm}^n(x)$ -матриць n -го степеня та pm -го порядку для шифрування даних, де $p \geq 3, m \geq 1, n \geq 1, x \geq 1$, елементами яких є поліноми Фібоначчі. Запропонований метод дешифрування даних впливає з можливості отримання відповідних обернених $Q_{pm}^n(x)$ -матриць, визначник яких становить ± 1 . При цьому спостерігалися як великі швидкості виконання таких процедур, так і отриманих зашифрованих повідомлень, стійких до криптографічних атак [21, 27].

Проаналізовані вище роботи стосовно наявних підходів до генерування поліномів Люка, а також особливостей їх використання для шифрування даних вказали на те, що навіть за останнє десятиліття виконано багато різноманітних досліджень, в кожному з яких тією чи іншою мірою обґрунтовано різні підходи до генерування таких поліномів і доведено доцільність їх використання для шифрування даних. Водночас було з'ясовано, що як окрема процедура захисту поточкових і блокових даних поліномами Люка у теорії та практиці криптографії трапляються вкрай рідко. Тому проведене нами дослідження щодо ефективних методів генерування поліномів Люка, а також встановлення певних особливостей їх використання для шифрування поточкових і блокових даних тільки як локального методу є актуальною проблемою, яку було частково вирішено в цьому дослідженні.

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – розроблено метод генерування поліноміальних матриць Люка n -го степеня та їх обернених матриць, елементами яких, на відміну від елементів традиційних матриць Люка, є відповідні поліноми Люка n -го степеня, які є основою шифрування блокових даних, що дає можливість здійснювати їх ефективний захист.

Практична значущість результатів дослідження – запропонований метод матричного подання поліномів Люка n -го степеня та їх обернених еквівалентів можна використовувати для шифрування поточкових даних, а розроблений метод генерування поліноміальних матриць Люка n -го степеня, елементами яких є відповідні поліноми Люка, та їх обернених матриць, можна застосовувати в традиційному методі шифрування блокових даних, що загалом дасть можливість ефективно передавати каналами зв'язку поточкові та блокові повідомлення різної величини.

Висновок / Conclusions

Розроблено методи генерування послідовностей поліномів Люка n -го степеня як основи шифрування поточкових і блокових даних, що дасть змогу ефективно передавати каналами зв'язку відповідні повідомлення різ-

ної величини. За результатами виконаного дослідження можна зробити такі основні висновки.

1. Внаслідок проведеного аналізу наявних досліджень та публікацій з'ясовано, що навіть за останнє десятиліття надруковано значну кількість різноманітних публікацій, в кожній з яких тією чи іншою мірою обґрунтовано різні підходи до генерування поліномів Люка n -го степеня та доведено доцільність їх використання для шифрування даних. Проте, більшість досліджень стосується окремих процедур захисту поточкових і блокових даних, що в теорії та практиці криптографії трапляються вкрай рідко. Виявлено основну складність проблеми генерування послідовностей поліномів Люка n -го степеня, які стануть основою шифрування поточкових і блокових даних, що дасть можливість здійснювати ефективний їх захист.

2. Наведено відомі способи подання чисел Фібоначчі та Люка, а також поліномів Люка, що мають багато корисних властивостей, використано тільки ті із них, які можна застосовувати у традиційному методі шифрування даних. Встановлено, що в послідовностях чисел Фібоначчі та Люка, де відношення двох послідовних доданків наближається до золотого перерізу, їхні члени фактично є наближеннями цілих степенів золотого перерізу.

3. Запропоновано метод матричного подання поліномів Люка та їх обернених еквівалентів, які можна застосовувати у традиційному методі шифрування даних, наведено конкретний приклад їх застосування. Для цього необхідно перемножити спеціальну матрицю n -го порядку з відповідними коефіцієнтами на поліном n -го степеня, внаслідок чого отримаємо набір поліномів Люка $L_n(x)$ відповідного степеня. Зазвичай, спеціальна матриця є нижньою трикутною матрицею, всі елементи головної діагоналі якої є одиничними.

4. Розроблено метод генерування поліноміальних матриць Люка n -го степеня та їх обернених матриць, елементами яких є поліноми Люка, які можна застосовувати у традиційному методі шифрування даних, наведено конкретний приклад їх застосування. Оскільки поліноміальні матриці Люка можна використовувати для реалізації операції шифрування даних, то аналогічні поліноміальні оберненні матриці також потрібні для процесу розшифрування даних, генерування яких має мати загальний вигляд. Розроблено ПЗ, яке дає змогу генерувати поліноміальні матриці Люка n -го степеня та m -го порядку, а також їхні обернені поліноміальні матриці аналогічного степеня та порядку.

5. За результатами виконаного дослідження зроблено висновки та надано відповідні рекомендації щодо їх практичного використання як основи для шифрування поточкових і блокових даних, що дасть змогу ефективно передавати каналами зв'язку відповідні повідомлення різної величини.

References

1. Abd-Elhameed, Waleed Mohamed, & Napoli, Anna. (2023). Some Novel Formulas of Lucas Polynomials via Different Approaches. *Symmetry*, 15(1), article ID 185. <https://doi.org/10.3390/sym15010185>
2. Ait-Amrane, L., & Behloul, D. (2022). Generalized hyper-Lucas numbers and applications. *Indian Journal of Pure and Applied Mathematics*, 53, 62–75. <https://doi.org/10.1007/s13226-021-00013-y>
3. Asci, M., & Gurel, E. (2017). Some properties of k -order Gaussian Fibonacci and Lucas numbers. *Ars Combinatoria*, 135, 345–356. URL: <https://hdl.handle.net/11499/23655>

4. Asci, M., & Tasci, D. (2007). On Fibonacci, Lucas and special orthogonal polynomials. *Journal of Computational and Applied Mathematics*. <https://doi.org/10.1016/J.CAM.2007.01.026>
5. Basu, M., & Prasad, B. (2009). The Generalized relations among the code elements for Fibonacci coding theory. *Chaos, Solitons and Fractals*, Vol. 41, issue 5(14), 2517–2525. <https://doi.org/10.1016/j.chaos.2008.09.030>
6. Basu, M., & Prasad, B. (2011). Coding theory on the (m, t) -extension of the Fibonacci p -numbers. *Discrete Mathematics, Algorithms and Applications*, Vol. 3, 259–267. <https://doi.org/10.1142/S1793830911001097>
7. Basu, Manjusri, & Das, Monojit. (2017). Coding theory on generalized Fibonacci n -step polynomials. *Journal of Information and Optimization Sciences*, Vol. 38, issue 1, 83–131. <https://doi.org/10.1080/02522667.2016.1160618>
8. Basu, Manjusri, & Prasad, Bandhu. (2009, November). Coding theory on the m -extension of the Fibonacci p -numbers. *Chaos, Solitons, & Fractals*, Vol. 42, issue 4, 30, 2522–2530. <https://doi.org/10.1016/j.chaos.2009.03.197>
9. Bednar, Urszula, & Wołowicz-Musiał, Małgorzata. (2020). Distance Fibonacci Polynomials. *Symmetry*, 12(9), article ID 1540. <https://doi.org/10.3390/sym12091540>
10. Bellini, Emanuele, Marcolla, Chiara, & Murru, Nadir. (2020, March). On the decoding of 1-Fibonacci error correcting codes. *Discrete Mathematics, Algorithms and Applications*, Vol. 13, No. 05, article ID 2150056. <https://doi.org/10.13140/RG.2.2.27280.97281>
11. Chasnov, Jeffrey R. (2008). Fibonacci numbers and the golden ratio. *The Hong Kong University of Science and Technology*, 87 p. URL: <https://www.math.hkust.edu.hk/~machas/fibonacci.pdf>
12. Esmaeili, M., & Esmaeili, M. (2009). Polynomial Fibonacci-Hessenberg matrices. *Chaos, Solitons and Fractals*, Vol. 41, issue 5, 2820–2827. <https://doi.org/10.1016/j.chaos.2008.10.012>
13. Esmaeili, M., & Esmaeili, M. (2010). A Fibonacci-polynomial based coding method with error detection and correction. *Computers and Mathematics with Applications*, Vol. 60, issue 10, 2738–2752. <https://doi.org/10.1016/j.camwa.2010.08.091>
14. Esmaeili, M., Esmaeili, M., & Gulliver, T. A. (2011). High-rate Fibonacci polynomial codes. In: *Proceedings of IEEE International Symposium on Information Theory Proceedings*, St. Petersburg, Russia, pp. 1921–1924. <https://doi.org/10.1109/ISIT.2011.6033886>
15. Esmaili, M., Moosavi, M., & Gulliver, T. A. (2017, January). A new class of Fibonacci sequence based error correcting codes. *Cryptography and Communications*, Vol. 9, 379–396. <https://doi.org/10.1007/s12095-015-0178-x>
16. Falcon, S., & Plaza, A. (2009). On k -Fibonacci sequences and polynomials and their derivatives. *Chaos, Solitons and Fractals*, Vol. 39, issue 3, 1005–1019. <https://doi.org/10.1016/j.chaos.2007.03.007>
17. Flaut, Cristina, & Savin, Diana. (2017, April). Some remarks regarding generalized Fibonacci-Lucas numbers and generalized Fibonacci-Lucas quaternions. arXiv:1705.00361v2 [math.RA]. <https://doi.org/10.48550/arXiv.1705.00361>
18. Goy, Taras, & Shattuck, Mark. (2019, December). Fibonacci and Lucas Identities from Toeplitz–Hessenberg Matrices. *Applications and Applied Mathematics: An International Journal*, 14(2), 699–715. URL: https://www.researchgate.net/publication/337906242_Fibonacci_and_Lucas_Identities_from_Toeplitz-Hessenberg_Matrices
19. Grytsiuk, Yuriy, Grytsyuk, Pavlo. (2015). Perfecting of the matrix Affine cryptosystem information security. Computer Science and Information Technologies: Proceedings of Xth International Scientific and Technical Conference (CSIT2015), 14–17 September, 2015, pp. 67–69. <https://doi.org/10.1109/stc-csit.2015.7325433>
20. Grytsiuk, P. Y., & Hrytsiuk, Yu. I. (2024). Methods for generating Fibonacci polynomials and features of their use for data encryption. *Scientific Bulletin of UNFU*, 34(7), 161–173. <https://doi.org/10.36930/40340720>
21. Grytsiuk, P. Yu., & Hrytsiuk, Yu. I. (2015). Peculiarities of the implementation of the matrix Athena cryptosystem of information protection. *Scientific Bulletin of UNFU*, 25(5), 346–356. URL: <https://nv.nltu.edu.ua/index.php/journal/article/view/1092>
22. Hoggat, V. E. (1969). Fibonacci and Lucas numbers. Palo Alto (CA): Houghton-Mifflin. URL: <https://www.peliti.org/Notes/fibonacciLucas.pdf>
23. Hoggat, V. E., Bicknell, Marjorie. (1973). Roots of Fibonacci polynomials. *The Fibonacci Quarterly*, Vol. 11, issue 3, 271–274. <https://doi.org/10.1080/00150517.1973.12430825>
24. Hrytsiuk, Yu. I., & Grytsiuk, P. Yu. (2015). Implementation of cryptographic transformations using Fibonacci $G(\lambda)$ -matrices. *Mathematical and software support of intelligent systems: materials of the 13th International Scientific and Practical Conference*, (pp. 53–54), November 18–20, 2015, Dnipropetrovsk, Ukraine. Dnipropetrovsk: Department of Dnipropetrovsk National University named after Olesya Honchara.
25. Hrytsiuk, Yu. I., & Grytsiuk, P. Yu. (2015). Methods and means of generating Fibonacci Q_p -matrices – keys for implementing cryptographic transformations. *Scientific Bulletin of UNFU*, 25(6), 334–351. URL: <https://nv.nltu.edu.ua/index.php/journal/article/view/974>
26. Hrytsiuk, Yu. I., & Grytsiuk, P. Yu. (2016). Features of generating Fibonacci Q_p -matrices – keys for implementing cryptographic transformations. *Bulletin of the National University "Lviv Polytechnic". Series: Computer Science and Information Technology*, Vol. 843, 251–263. URL: <https://vlp.com.ua/node/16094>
27. Hrytsiuk, Yu. I., & Grytsiuk, P. Yu. (2016). Features of generating Fibonacci $G_p(\lambda)$ -matrices for implementation of cryptographic transformations. *Information extraction and processing: interdepartmental collection of scientific papers*, 43(119), 86–95. URL: <http://vidbir.ipm.lviv.ua/vidbir-zm-2016-43e.htm>
28. Hrytsiuk, Yuriy, & Grytsyuk, Pavlo. (2016). Generation of Fibonacci $Q_p(\lambda)$ -matrices – keys for data encryption. *Information protection and security of information systems: materials of the 5th International Scientific and Technical Conference*, (pp. 39–40), June 02–03, 2016, Lviv, Ukraine. Lviv: Lviv Polytechnic State University. URL: <http://dw1.lviv.ua/art/nulp/istc.pdf>
29. Hrytsiuk, Yuriy, Grytsyuk, Pavlo, Dyak, Tetiana, & Hrynyk, Heorhiy. (2019). Software Development Risk Modeling. IEEE 2019 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT 2019), (Vol. 2, pp. 134–137), 17–20 September, Lviv, Ukraine. Lviv: Lviv Polytechnic National University, 206 p. <https://doi.org/10.1109/stc-csit.2019.8929778>
30. Jin, Z. (2018). On the Lucas polynomials and some of their new identities. *Adv Differ Equ*, 126 (2018). <https://doi.org/10.1186/s13662-018-1527-9>
31. Kargin, Alparslan, Kişi, Emre, & Özdemir, Halim. (2019, September). Some Notes on Odd or Even Indexed Fibonacci and Lucas Sequences. *Sakarya University Journal of Science*, 23(5), 929–933. <https://doi.org/10.16984/soaufenbilder.536642>
32. Knuth, Donald. (1968). The art of computer programming, Vol. 1: Fundamental algorithms 2: Semi Numerical algorithms 3: *Sorting and searching*, MA: Addison-Wesley 30. URL: <https://www.amazon.com/Art-Computer-Programming-Vol-Fundamental/dp/0201038099>
33. Koshy, Thomas. (2001). Fibonacci and Lucas Numbers with Applications. A Wiley-Interscience Publication, 654 p. <https://doi.org/10.1002/9781118033067>
34. Kuhapatanakul, K. (2015). The Lucas p -matrix. *International Journal of Mathematical Education in Science and Technology*. <https://doi.org/10.1080/0020739X.2015.1026612>
35. Neeraj Kumar, Paul, & Saikia, Helen K. (2023). A generalization of Lucas sequence and associated identities. *Boletim da Sociedade Paranaense de Matematica*, Vol. 41, 1–20. <https://doi.org/10.5269/bspm.53068>
36. Nihal Tas, Sumeyra Ucar, Nihal Yilmaz Ozgur, & Oztunc Kaymak. (2018). A new coding/decoding algorithm using Fibonacci numbers. *Discrete Mathematics, Algorithms and Applications*, Vol. 10, No. 02, article ID 1850028. <https://doi.org/10.1142/S1793830918500283>; <https://doi.org/10.48550/arXiv.1712.02262>

37. Özkan, Engin, Taştan, Merve & Aydoğdu, Ali. (2019). Fibonacci Sayılarının Ailesinde 3-Fibonacci Polinomları. *Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi*. Cilt: 12 Sayı: 2, 926–933. <https://doi.org/10.18185/erzifed.512100>
38. Perumali, Sundarayya, & Prasad, M. G. Vara. (2019, October). Coding theory on Pell-Lucas p -numbers. *Journal of Physics Conference Series*, 1344(1), article ID 012017. <https://doi.org/10.1088/1742-6596/1344/1/012017>
39. Postavaru, Octavian. (October 2023). An efficient numerical method based on Fibonacci polynomials to solve fractional differential equations. *Mathematics and Computers in Simulation*, Vol. 212, 406–422. <https://doi.org/10.1016/j.matcom.2023.04.028>
40. Prasad, Bandhu. (2014). Coding theory on $(h(x), g(y))$ -extension of Fibonacci p -numbers polynomials. *Universal Journal of Computational Mathematics*, Vol. 2(1), 6–10. <https://doi.org/10.13189/ujcmj.2014.020102>
41. Prasad, Bandhu. (2014). High rates of Fibonacci polynomials coding theory. *Discrete Mathematics, Algorithms and Applications*, Vol. 06, No. 04, article ID 1450053. <https://doi.org/10.1142/S1793830914500530>
42. Prasad, Bandhu. (2016). Coding theory on Lucas p -numbers. *Discrete Mathematics, Algorithms and Applications*, Vol. 08, No. 04, article ID 1650074. <https://doi.org/10.1142/S1793830916500749>
43. Prasad, Bandhu. (2019). The generalized relations among the code elements for a new complex Fibonacci matrix. *Discrete Mathematics, Algorithms and Applications*, Vol. 11, No. 02, article ID 1950026. <https://doi.org/10.1142/S1793830919500265>
44. Prasad, Kalika, Kumari, Munesh, & Mahat, Hrishikesh. (2024). A modified public key cryptography based on generalized Lucas matrices. *Communications in Combinatorics and Optimization*. <https://doi.org/10.22049/cco.2024.28022.1419>
45. Sentürk, G. Y., Gürses, N., & Yüce, S. (2022). Construction of dual-generalized complex Fibonacci and Lucas quaternions. *Carpathian Mathematical Publications*, 14(2), 406–418. <https://doi.org/10.15330/cmp.14.2.406-418>
46. Sikhwal, Omprakash, & Vyas, Yashwant. (2014). Fibonacci Polynomials and Determinant Identities. *Turkish Journal of Analysis and Number Theory*, 2(5), 189–192. <https://doi.org/10.12691/tjant-2-5-6>
47. Singh, B., Sikhwal, O., & Panwar, Y. K.. (2009). Generalized Determinantal Identities Involving Lucas Polynomials. *Applied Mathematical Sciences*, 3(8), 377–388. URL: https://www.researchgate.net/publication/228526069_Generalized_Determinantal_Identities_Involving_Lucas_Polynomials
48. Stakhov, A. P. (2006, October). Fibonacci matrices, a generalization of the "Cassini formula", and a new coding theory. *Chaos, Solitons, & Fractals*, Vol. 30, issue 1, 56–66. <https://doi.org/10.1016/j.chaos.2005.12.054>
49. Stakhov, A., & Rozin, B. (2005). Theory of Binet formulas for Fibonacci and Lucas p -numbers. *Chaos, Solitons and Fractals*, 27, No. 5, 1162–1177. <https://doi.org/10.1016/j.chaos.2005.04.106>
50. Stakhov, Alexey, & Olsen, Scott. (2009). The Mathematics of Harmony: From Euclid to Contemporary Mathematics and Computer Science. *Series on Knots and Everything*, 22. World Scientific Publishing Company; First Edition, 748 p. URL: <https://www.amazon.com/Mathematics-Harmony-Contemporary-Computer-Everything/dp/981277582X>
51. Uçar, Sümeýra, Tas, Nihal, & Özgür, Nihal. (2019, May). A New Application to Coding Theory via Fibonacci and Lucas Numbers. *Mathematical Sciences and Applications E-Notes*, 7(1), 62–70. <https://doi.org/10.36753/mathenot.559251>
52. Vajda, S. (1989). Fibonacci and Lucas Numbers and the Golden Section Theory and Applications. *Ellis Harwood Limited*, 190 p. <https://doi.org/10.2307/3619858>
53. Wang, Weiping, & Wang, Hui. (2017, August). Generalized Humbert polynomials via generalized Fibonacci polynomials. *Applied Mathematics and Computation*, Vol. 307, 204–216. <https://doi.org/10.1016/j.amc.2017.02.050>
54. Yang, Jizhen, & Zhang, Zhizheng. (2018, December). Some identities of the generalized Fibonacci and Lucas sequences. *Applied Mathematics and Computation*, Vol. 339, 451–458. <https://doi.org/10.1016/j.amc.2018.07.054>

P. Yu. Grytsiuk, Yu. I. Hrytsiuk

Lviv Polytechnic National University, Lviv, Ukraine

METHODS FOR GENERATING LUCAS POLYNOMIALS AND FEATURES OF THEIR USE FOR DATA ENCRYPTION

Methods for generating sequences of Lucas polynomials of the n th degree as the basis for encrypting stream and block data have been developed, which makes it possible to effectively transmit corresponding messages of various sizes over communication channels. It has been found that a significant number of publications have been published over the past decade, each of which substantiates different approaches to generating Lucas polynomials and proves the feasibility of their use for data encryption. However, most of the research concerns individual data protection procedures that are extremely rare in the theory and practice of cryptography. The main complexity of the problem of generating sequences of Lucas polynomials of the n th degree, which are the basis for encrypting stream and block data, has been established, which will make it possible to effectively protect them. Known ways of representing Fibonacci and Lucas numbers, as well as Lucas polynomials, which can be used in the traditional method of data encryption, are given. It is established that in the sequences of Fibonacci and Lucas numbers, where the ratio of two consecutive terms approaches the golden ratio, their terms are approximations of integer powers of the golden ratio. A method is proposed for matrix representation of Lucas polynomials and their inverse counterparts for use in traditional data encryption. To do this, it is necessary to multiply a special matrix of the n th order with the corresponding coefficients by a polynomial of the n th degree, as a result of which we obtain a set of Lucas polynomials $L_n(x)$ of the corresponding degree. Typically, a special matrix is a lower triangular matrix, all elements of the main diagonal of which are units. A method of generating Lucas polynomial matrices of the n th degree and their inverse matrices, the elements of which are Lucas polynomials, which can be used in the traditional method of data encryption, is developed. Since polynomial inverse matrices are used to decrypt data, their generation must have a general form. Software has been developed that allows generating Lucas polynomial matrices of the n th degree and m th order, as well as their inverse polynomial matrices of the similar degree and order. Based on the results of the research, conclusions were drawn and appropriate recommendations were given regarding their practical use as a basis for encrypting stream and block data, which will allow for the effective transmission of corresponding messages of various sizes via communication channels.

Keywords: Lucas numbers; recurrent sequence of polynomials; golden ratio; encryption of stream and block data; Lucas polynomial matrix; inverse polynomial matrix.