УДК 004.056.5 **DOI:10.30837/rt.2025.3.222.04**

І.В. ЛИСИЦЬКА, д-р техн. наук, К.Є. ЛИСИЦЬКИЙ, І.М. ГАЛЬЦЕВА,
Є.П. КОЛОВАНОВА, канд. техн. наук

ОСОБЛИВОСТІ ПОБУДОВИ НЕЛІНІЙНИХ ПЕРЕТВОРЕНЬ БЛОКОВИХ СИМЕТРИЧНИХ ШИФРІВ

Вступ

Нелінійність є життєво важливою для криптографічної стійкості, оскільки вона ускладнює лінійні та алгебраїчні атаки. Без нелінійних перетворень шифри були б вразливими до відносно простих криптоаналітичних методів, які могли б ефективно зламати шифр.

Історія виникнення нелінійних перетворень у блокових симетричних шифрах бере свій початок у фундаментальних принципах Шеннона. У роботі "Communication Theory of Secrecy Systems" Шеннон заклав основи сучасної криптографії. Він розглянув важливі загальні концепції розсіювання і перемішування. Для досягнення цих властивостей Шеннон пропонував використання "комбінованих шифрів", що містять простіші операції, такі як шифри заміни (substitution) і шифри перестановки (permutation).

У 1970 р. Хорст Файстель став ключовою фігурою у розробці перших сучасних блокових шифрів. Його робота над шифром LUCIFER в IBM (1971) заклала основу для архітектури, відомої як мережа Файстеля. У цих мережах шифрування відбувається в кілька раундів, де кожен раунд включає нелінійну функцію, яка залежить від ключа.

У ранніх розробках блокових шифрів Файстель вже використовував механізм S-блоків. S-блоки є основним компонентом, що реалізує нелінійні заміни у симетричних шифрах. Вони приймають певну кількість вхідних бітів і перетворюють їх у певну кількість вихідних бітів нелінійним чином.

У 1977 р. Національне бюро стандартів США (зараз NIST) запропонувало блоковий симетричний шифр. DES [1], який базується на мережі Файстеля, активно використовує S-блоки як єдиний нелінійний елемент. Конструкція S-блоків DES спочатку трималася в секреті. Пізніше стало відомо, що ці S-блоки були ретельно спроєктовані, щоб забезпечити стійкість до нового тоді методу криптоаналізу – диференціального криптоаналізу.

Цей потужний метод криптоаналізу був запропонований Біхамом і Шаміром у 1990 р., хоча його використання було відоме розробникам DES раніше [2]. Лінійний криптоаналіз був відкритий Мацуї в 1993 р. [3].

Еволюція блокових симетричних шифрів після DES рухалася в напрямку збільшення довжини ключа, розміру блоку та вдосконалення внутрішньої структури для стійкості до нових криптоаналітичних атак. З'являлись нові ідеї побудови нелінійних перетворень з покращеними криптографічними показниками. Національний інститут стандартів і технологій США (NIST) ініціював відкритий конкурс на новий стандарт блокового шифру, який мав би замінити DES. Цей конкурс тривав з 1997 по 2000 р. і сприяв розробці безлічі нових та інноваційних алгоритмів. Деякі з помітних кандидатів: Serpent, Mars, Twofish, RC6, Blowfish, IDEA, Skipjack, CAST-128 / CAST-256 [4].

Крім структури типу мережа фейстеля були запропоновані структура "мережа Лая-Массі" (Lay-Massey network) [5] та SPN-мережа.

SPN (Substitution-Permutation Network), або підстановочно-перестановочна мережа, є однією з фундаментальних архітектур, що використовуються для побудови симетричних блокових шифрів. Найвідомішим прикладом шифру, що базується на SPN, є AES (Advanced Encryption Standard) – фіналіст конкурсу NIST [6].

S-блоки (Substitution-boxes) є серцем нелінійності в багатьох симетричних блокових шифрах (особливо в SPN-структурах). Їхня якість має вирішальне значення для стійкості шифру до криптоаналітичних атак.

Основні властивості S-блоків, які забезпечують криптографічно стійке нелінійне перетворення

1. Висока нелінійність (High Non-linearity): Це найважливіша властивість. S-блок має бути максимально нелінійним. Це протистоїть лінійному криптоаналізу.

2. Низька диференціальна однорідність (Low Differential Uniformity): Для будь-якої ненульової вхідної різниці Δx, вихідні різниці $\Delta y = S(x) \oplus S(x \oplus \Delta x)$ повинні розподілятися якомога рівномірніше. Це протистоїть диференціальному криптоаналізу. Ідеально – мати диференціальну однорідність 2 (це означає, що для будь-якої вхідної різниці та будь-якої вихідної різниці, існує не більше двох пар вхідних значень, що дають таку різницю).

3. Бієктивність (Bijectivity): S-блок має бути бієктивною функцією, тобто кожне вхідне значення повинне відображатися на унікальне вихідне значення, і навпаки. Це забезпечує оборотність процесу дешифрування.

4. Збалансованість (Balancedness): Кожне можливе вихідне значення S-блоку має з'являтися однакову кількість разів. Це запобігає витоку інформації через статистичний зсув.

5. Лавинний ефект (Avalanche Effect). Невелика зміна на вході S-блока (наприклад, зміна одного біта) повинна призводити до значної зміни на виході (багато вихідних бітів повинні змінити своє значення). Це властивість важлива для поширення змін по всьому шифротексту, що робить його стійким до статистичних атак.

6. Відсутність фіксованих точок та антифіксованих точок (No Fixed Points and Antifixed Points).

7. Високий алгебраїчний ступінь (High Algebraic Degree): Функція S-блока повинна мати високий алгебраїчний ступінь та бути складною для опису за допомогою простих алгебраїчних виразів. Це протистоїть алгебраїчним атакам [7].

S-блоки можуть бути сконструйовані різними способами, і кожен підхід має свої переваги та недоліки.

Різні структури (методи побудови) S-блоків

1. Детерміновані S-блоки (Fixed S-boxes) [6, 8, 9]

Це S-блоки, які є жорстко заданими таблицями або математичними функціями, і їхні значення не змінюються під час шифрування. Типовими прикладами таких S-блоків є S-блоки алгоритмів DES, Serpent, Camellia, ARIA, AES та деякі інші.

Переваги цього підходу:

- простота реалізації та швидкість. Це робить їх придатними для апаратних реалізацій та високопродуктивних систем;

- ретельне проєктування та перевірка: S-блоки, які використовуються в широко відомих та перевірених алгоритмах розроблені таким чином, щоб мати бажані криптографічні властивості, такі як висока нелінійність, низька диференціальна однорідність та стійкість до відомих криптоаналітичних атак (лінійний та диференціальний криптоаналіз);

- стабільність та передбачуваність властивостей: Оскільки S-блоки є фіксованими, їхні криптографічні властивості є постійними та передбачуваними. Це дозволяє розробникам шифрів точно оцінювати безпеку алгоритму;

- сумісність та стандартизація: Використання фіксованих S-блоків полегшує стандартизацію алгоритмів шифрування, забезпечуючи сумісність між різними реалізаціями та системами.

Недоліки:

- вразливість до атак на основі структури: якщо S-блоки мають приховану математичну структуру, це може бути використано в алгебраїчних атаках або інших видах криптоаналізу для пошуку вразливостей;

- потенційна вразливість до атак по стороннім каналам (Side-channel attacks): оскільки S-блоки є таблицями пошуку, їх використання може бути вразливим до атак по стороннім каналам, таким як атаки по часовим затримкам (cache-timing attacks). Це пов'язано з тим, що

час доступу до пам'яті може залежати від того, до якої частини S-блоку відбувається звернення, що може дати зловмиснику інформацію про дані або ключ;

- відсутність адаптивності: фіксовані S-блоки не змінюються залежно від ключа або раунду шифрування. Це може зробити криптоаналіз простішим у порівнянні з системами, які використовують динамічні (ключозалежні) S-блоки, оскільки атакуючому не потрібно обчислювати або дізнаватися структуру S-блоку для кожного окремого випадку;

- ризик "бекдорів" (Backdoors): хоча це малоймовірно для широко відомих стандартів, теоретично існує ризик того, що фіксовані S-блоки могли бути розроблені з прихованими "бекдорами", які дозволяють полегшити криптоаналіз за певних умов;

- розробка криптографічно сильних фіксованих S-блоків є дуже складним завданням.

Пошук S-блоків, які задовольняють всім необхідним криптографічним критеріям (висока нелінійність, низька диференціальна однорідність, хороший лавинний ефект тощо), вимагає значних досліджень та обчислювальних ресурсів.

2. Динамічні S-блоки (Dynamic/Key-Dependent S-boxes) [10, 11, 12]

Це S-блоки, які генеруються або змінюються під час виконання алгоритму, зазвичай, на основі секретного ключа. Типовими представниками цього підходу є Blowfish, Twofish, Khufu/Khafre.

Переваги динамічних S-блоків:

- підвищена стійкість до криптоаналізу "білої скриньки" (White-box Cryptanalysis). Оскільки структура S-блоків змінюється для кожного ключа, криптоаналітику набагато складніше провести атаки, що базуються на статичній структурі, оскільки він не знає, які саме S-блоки використовуються для конкретного ключа. Це ускладнює пошук "бекдорів" або прихованих вразливостей, які могли би бути вбудовані у фіксовані S-блоки [13];

- ускладнення універсальних атак. Атаки, які покладаються на фіксовані властивості S-блоків (наприклад, лінійні або диференціальні апроксимації), стають значно складнішими, оскільки ці властивості змінюються з кожним ключем [14];

- зменшення ризику "бекдора". Якщо S-блоки генеруються криптографічно надійною функцією з використанням ключа, то розробник алгоритму не може навмисно вбудувати приховану вразливість, оскільки вона залежатиме від ключа, який йому невідомий. Це підвищує довіру до прозорості розробки.

Недоліки динамічних S-блоків:

- високі обчислювальні витрати на генерацію. Процес генерації ключозалежних S-блоків може бути обчислювано дорогим і займати значний час, особливо якщо S-блоки повинні мати високі криптографічні властивості. Це може уповільнити фазу ініціалізації шифру [10];

- складність забезпечення криптографічних властивостей. Гарантувати, що згенеровані S-блоки завжди матимуть бажані криптографічні властивості (високу нелінійність, низьку диференціальну однорідність тощо) для будь-якого ключа, є складним завданням. Якщо механізм генерації може створювати "слабкі" S-блоки для певних ключів, це може створити вразливість [15];

- складність криптографічного аналізу. Хоча динамічні S-блоки ускладнюють атаки для криптоаналітика, вони також ускладнюють криптографічний аналіз шифру. Довести безпеку шифру зі змінними S-блоками набагато важче, оскільки необхідно аналізувати алгоритм генерації S-блоків та їхні властивості в цілому, а не лише одну фіксовану таблицю [16];

- збільшення складності реалізації. Алгоритми з динамічними S-блоками є більш складними для реалізації порівняно з тими, що використовують фіксовані таблиці, оскільки вони вимагають додаткового коду для процедури генерації S-блоків.

Таким чином, використання динамічних S-блоків є компромісом. Хоча вони додають шар складності для криптоаналітика, вони також створюють проблеми для розробника шифру у забезпеченні гарантованої безпеки для всіх можливих випадків.

3. Хаотично згенеровані S-блоки

Деякі дослідження пропонують використовувати хаотичні системи для генерації S-блоків. Хаотично згенеровані S-блоки (Substitution-boxes) використовують властивості хао-

тичних систем (такі як чутливість до початкових умов, непередбачуваність, ергодичність та топологічне перемішування) для створення S-блоків, що демонструють високі криптографічні властивості. Це робить їх привабливим напрямком у створенні сучасних шифрів. Існують розробки у таких напрямках:

S – блоки, засновані на логістичному відображенні (Logistic Map);

S – блоки, засновані на системах Чуа (Chua's Circuit);

S – блоки, засновані на Хенон-відображенні (Hénon Map) або інших багатовимірних хаотичних відображеннях;

S – блоки, що використовують гіперхаотичні системи.

Переваги хаотично згенерованих S-блоків:

- висока нелінійність та стійкість до криптоаналізу. Хаотичні системи за своєю природою є нелінійними, що дозволяє генерувати S-блоки з високими показниками нелінійності. Це є ключовою властивістю для забезпечення стійкості до лінійного та диференціального криптоаналізу [17, 18];

- підвищена чутливість до ключа та розповсюдження помилок (Avalanche Effect): навіть мінімальні зміни у вхідних даних (наприклад, у секретному ключі) призводять до значно різних вихідних даних. Це забезпечує сильний лавинний ефект (avalanche effect), де зміна одного біта у відкритому тексті або ключі призводить до зміни близько половини бітів у шифротексті, що ускладнює криптоаналіз [17, 19];

- додаткове заплутування (Confusion): Хаотичні відображення, хоча й детерміновані, демонструють надзвичайно складну динаміку. Ця складність посилює властивість заплутування (confusion), яка є однією з двох фундаментальних властивостей безпечного шифрування (за Шенноном) [17, 20];

- ергодичність та статистична однорідність: Властивість ергодичності хаотичних відображень дозволяє їм з часом охоплювати весь простір станів, що сприяє статистичній однорідності зашифрованих вихідних даних. Це підвищує стійкість до криптоаналізу, який базується на статистичних властивостях шифротексту [17];

- потенціал для ефективності (у певних реалізаціях). Хаотичні системи можуть бути використані для створення S-блоків, які є ефективними з точки зору використання ресурсів та пропускної здатності [17, 18].

Недоліки:

- складність гарантування криптографічних властивостей: важко довести, що хаотичні відображення завжди генерують S-блоки з бажаними криптографічними властивостями (нелінійність, диференціальна однорідність тощо) для всіх можливих початкових умов;

- реалізаційні складнощі: вимагають точної реалізації хаотичних систем, що може бути чутливим до помилок округлення в комп'ютерних системах;

- менш досліджені: хоча є багато досліджень, вони поки не так широко інтегровані в основні стандарти, як фіксовані S-блоки.

Більшість цих "хаотичних" шифрів залишаються в царині досліджень і академічних публікацій. Вони демонструють цікаві концепції та потенціал, але їм, як правило, бракує суворого криптоаналізу та стандартизації, які необхідні для прийняття в якості надійних криптографічних примітивів для широкого використання. Ці шифри часто є "одноразовими" пропозиціями в наукових статтях і рідко отримують загальновизнані назви.

Загалом, використання хаотичних систем у генерації S-блоків є перспективним напрямом для розробки криптографічно стійких алгоритмів, особливо для застосувань, де потрібна висока нелінійність та стійкість до відомих атак.

4. Використання випадкових S-блоків

Окремий напрямок, який було розроблено у чисельних роботах, серед яких [21 – 23]. Використання "випадкових" S-блоків у блочних шифрах – це концепція, де таблиця заміни не є жорстко зафіксованою або детерміновано математично побудованою, натомість генерується або вибирається таким чином, що її конкретна структура виглядає випадковою або є непередбачуваною. Запропоновано в якості S-блоків шифрів використовувати випадкові підстановки з виходу генератора випадкових підстановок, що проходять перевірку на відповідність додатним критеріям відбору.

Важливий висновок міститься в тому, що для випадково взятої булевої функції більшість її криптографічних параметрів близькі до оптимальних.

В результаті набуває актуальності задача побудування шифрів, в яких без зниження стійкості можуть бути використані випадкові S-блоки, яка знайшла своє перше вирішення в роботі [24] та ряді інших.

Переваги використання випадкових S-блоків:

- захист від "бекдорів" та прихованих вразливостей. Якщо S-блоки генеруються випадковим або псевдовипадковим чином, це усуває можливість для розробника шифру навмисно вбудувати приховану слабкість або "бекдор", оскільки навіть сам дизайнер не знатиме точної структури S-блоків для конкретного ключа. Це підвищує довіру до шифру;
- стійкість до криптоаналізу. Для випадково взятої булевої функції більшість її криптографічних параметрів близькі до оптимальних;

Недоліки використання випадкових S-блоків:

- існує ризик, що для певних випадкових початкових значень (або ключів) можуть бути згенеровані "слабкі" S-блоки, що скомпрометують шифр.
- обчислювальні витрати на генерацію. Процес генерації S-блоків на основі ключа або випадкових даних може займати значний час.

Порівняння за основними властивостями різних типів S-блоків надано у табл. 1.

Таблиця 1

Властивостями різних типів S-блоків

Властивість / Тип S-блоку	Детерміновані S-блоки	Динамічні S-блоки	Хаотично згенеровані S-блоки	Випадкові S-блоки
Генерація	Фіксовані, заздалегідь визначені таблиці підстановки розроблені з використанням складних математичних конструкцій або пошукових алгоритмів для досягнення бажаних криптографічних властивостей	Генеруються або модифікуються під час виконання алгоритму шифрування, часто залежать від ключа або стану	Генеруються за допомогою хаотичних систем (наприклад, хаотичних відображень, клітинних автоматів), використовують детерміновану, але непередбачувану поведінку хаосу	Генеруються випадково (наприклад, як випадкові підстановки)
Змінюваність під час шифрування	Не змінюються	Змінюються	Зазвичай змінюються	Можуть бути статичними або динамічними
Нелінійність	Висока: зазвичай розроблені з максимально можливою нелінійністю для протидії лінійному криптоаналізу	Може бути високою: залежить від методу динамічної генерації	Залежить від хаотичної системи	Висока але не гарантовано оптимальна
Диференціальна однорідність	Низька	Може бути низькою: залежить від методу динамічної генерації	Залежить від хаотичної системи: повинна бути низькою через властивості хаосу, але може вимагати перевірки	В середньому, випадкові S-блоки мають прийнятну диференціальну однорідність, але не гарантовано оптимальну
Бієктивність (зворотність)	Так	Може бути так/ні	Може бути так/ні	Так
Складність реалізації	Низька	Висока	Висока	Низька
Складність проєктування	Висока	Середня	Середня	Низька
Залежність від ключа	Ні	Так	Так:	Може бути так/ні

У контексті використання симетричних блокових шифрів (БСШ) в умовах постквантової криптографії основна увага зосереджена на протидії алгоритму Гровера. Цей алгоритм дозволяє квантовому комп'ютеру здійснити пошук по неструктурованій базі даних (що включає перебір ключів) з квадратичним прискоренням. Тобто, якщо для класичного комп'ютера потрібно операцій для повного перебору n-бітного ключа, то для квантового комп'ютера – приблизно операцій. Це означає, що для забезпечення "постквантової стійкості" симетричних шифрів найпростішим і найпоширенішим рішенням є збільшення (подвоєння) довжини ключа. Щодо S-блоків, у постквантовій криптографії основні вимоги до них залишаються тими ж, що й у класичній.

Висновки

Вибір структури S-блоку завжди є компромісом між безпекою, продуктивністю та простотою реалізації. Вибір підходу до побудови S-блоку залежить від багатьох факторів, включаючи бажаний рівень безпеки, обчислювальні ресурси, доступні для проєктування, та специфічні вимоги до продуктивності. Сучасні шифри, як правило, віддають перевагу алгебраїчним S-блокам через їхні доведені криптографічні властивості та елегантність.

Однак гібридні підходи, що поєднують алгебраїчні та евристичні методи, також можуть бути використані для досягнення оптимальних результатів.

Щодо умов квантової криптографії. Підходи до розробки самих S-блоків кардинально не змінюються. Умови квантової криптографії впливають на вибір серед існуючих, добре відомих S-блоків. Якщо деякі з них виявляться значно складнішими для квантової реалізації та менш вразливими до певних типів квантових атак.

Постквантові БСШ, що пропонуються для стандартизації або широкого використання, як правило, покладаються на ретельно проаналізовані, фіксовані S-блоки. Причини ті ж самі, що й у класичному випадку: складність аналізу безпеки, обчислювальні витрати на генерацію та проблеми з відтворюваністю.

Таким чином, "різні типи S-блоків" (фіксовані, ключово-залежні, хаотично згенеровані, випадкові) залишаються предметом досліджень, але в контексті практичних постквантових БСШ переважає підхід використання перевірених S-блоків з достатньою довжиною ключа для захисту від атаки Гровера.

Список літератури:

1. FIPS PUB 46-3 (Federal Information Processing Standards Publication 46-3), Data Encryption Standard (DES). [Електронний ресурс]. Режим доступу: https://csrc.nist.gov/publications/detail/fips/46/3/archive/1999-10-25
2. Biham E. and Shamir A. Differential Cryptanalysis of DES-Like Cryptosystems ; Menezes A.J. and Vanstone S.A., Eds. // Advances in Cryptology-CRYPTO'90, Springer. Berlin, 1990. P. 2–21. [Електронний ресурс]. Режим доступу: https://www.scirp.org/reference/referencespapers?referenceid=2235215
3. Matsui M. Linear Cryptanalysis Method for DES Cipher ; T. Helleseth (Ed.) // Advances in Cryptology-EUROCRYPT '93 (LNCS. 1994. Vol. 765. P. 386–397). Springer, Berlin, Heidelberg.
4. National Institute of Standards and Technology. (n.d.). AES Development. Computer Security Resource Center. Retrieved from https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development
5. Lai X., Massey J. L., & Murphy S. A proposal for a new block encryption standard ; T. Helleseth (Ed.). // Advances in Cryptology-EUROCRYPT '90 (LNCS. 1991. Vol. 473. P. 389–404). Springer, Berlin, Heidelberg.
6. Daemen J., Rijmen V. The Design of Rijndael-The Advanced Encryption Standard. Springer-Verlag; Berlin Heidelberg, New York, 2002. 10.1007/978-3-662-04722-4.
7. Chew L., C. N., & Ismail E. S. S-box Construction Based on Linear Fractional Transformation and Permutation Function // Sensors. 2020. No 20(5). P. 826.
8. Canteaut A., Duval S., and Leurent G. Construction of LightweightS-Boxes Using Feistel and MISTY Structures // Lecture Notes in Com-puter Science. Springer International Publishing. 2016. P. 373–393. doi:10.1007/978-3-319-31301-6_22.
9. Jing-Mei L., Bao-Dian W., Xiang-Guo C., Xin-Mei W. Cryptanalysis of Rijndael S-box and improvement // Appl Math Comput. 2005. No 170. P. 958–975. 10.1016/j.amc.2004.12.043.
10. Blowfish Algorithm with Examples. (n.d.). GeeksforGeeks. Retrieved June 28, 2025, from https://www.geeksforgeeks.org/blowfish-algorithm-with-examples/.

11. The Twofish Encryption Algorithm: A 128-Bit Block Cipher. (1999). Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson.
12. Farwa S., Sohail A., Muhammad N. A novel application of elliptic curves in the dynamical components of block ciphers // Wirel. Pers. Commun. 2020. Vol. 115. P. 1309–1316.
13. Mustafa R. A., Jalab H. A., & Al-Qurashi M. S. Dynamic S-Box Construction Using Mordell Elliptic Curves over Galois Field and Its Applications in Image Encryption // Mathematics. 2023. Vol. 12(4). P. 587. https://doi.org/10.3390/math12040587
14. Keliher L., Meijer D., & Tavares E. Key-dependent S-boxes and differential cryptanalysis // International Workshop on Fast Software Encryption. 2004. P. 37–51. Springer, Berlin, Heidelberg.
15. Husain I., & Mahmood M. Construction of High Quality Key-dependent S-boxes // International Journal of Computer Science and Engineering (IJCSE). 2017. Vol. 6(3). P. 47–53.
16. Mustafa R. A., & Ahmad N. R. SECURITY ANALYSIS BETWEEN STATIC AND DYNAMIC S-BOXES IN BLOCK CIPHERS // Journal of Information Systems and Technology Management. 2018. Vol. 11(1). P. 19–32.
17. Huda M. A. A., & Al-Qurashi M. S. Chaos-Based S-Boxes as a Source of Confusion in Cryptographic Primitives // Electronics. 2025. Vol.14(11). P. 2198. https://doi.org/10.3390/electronics14112198
18. Zhang M., Han F., Sun K., Zhou C., & Xu F. A Novel S-Box Design Algorithm Based on a New Compound Chaotic System // Symmetry. 2020. Vol. 12(9). P. 1469.
19. Singh A. P., Kumar R., & Kumar R. Creation of S-Box based One-Dimensional Chaotic Logistic Map: Colour Image Encryption Approach // Open Access Journal of Information Technology. 2022. Vol. 1(1). P.1–11.
20. Belkhode V., & Raut R. A Chaos Based Method for Efficient Cryptographic S-box Design // International Journal of Computer Applications. 2016. Vol.133(1). P. 18–21.
21. Dolgov V.I., Lisitska I.V., Lisitskyi K.Ye. The new concept of block symmetric ciphers design // Telecommunications and Radio Engineering. 2017. Vol. 76, issue 2. P. 157–184.
22. Лисицький К., & Лисицька І. Mathematical model of random substitution // Radiotekhnika. 2020. No 3(202). P. 116–124. DOI: https://doi.org/10.30837/rt.2020.3.202.12
23. Lisickiy, K., Dolgov, V., Lisickaya, I., & Kuznetsova, K. Block Symmetric Cipher with Random S-boxes // International Journal of Computing. 2019. Vol. 18, iss. 1. P. 89–100. DOI: 10.47839/ijc.18.1.1278.
24. Dolgov V.I., Lisitska I.V., Lisitskyi K.Ye. The new concept of block symmetric ciphers design // Telecommunications and Radio Engineering. 2017. Vol. 76, issue 2. P. 157–184.

Надійшла до редколегії 17.06.2025

Відомості про авторів:

Лисицька Ірина Вікторівна – д-р техн. наук, професор, Харківський національний університет імені В. Н. Каразіна, професор кафедри кібербезпеки інформаційних систем, мереж і технологій, навчально-наукового інституту комп'ютерних наук та штучного інтелекту; Харківський національний університет радіоелектроніки, професор кафедри безпеки інформаційних технологій, Україна; e-mail: ivlisitska@karazin.ua; ORCID: https://orcid.org/0000-0001-6758-9516

Лисицький Костянтин Євгенійович – PhD, Харківський національний університет імені В. Н. Каразіна, доцент кафедри математичного моделювання і аналізу даних навчально-наукового інституту комп'ютерних наук та штучного інтелекту; Національний аерокосмічний університет "Харківський авіаційний інститут", старший викладач кафедри комп'ютерних систем, мереж і кібербезпеки; Україна; e-mail: constantin.lisickiy@gmail.com; ORCID: https://orcid.org/0000-0002-7772-3376;

Гальцева Ірина Михайлівна – Харківський національний університет імені В. Н. Каразіна, старший викладач кафедри кібербезпеки інформаційних систем, мереж і технологій навчально-наукового інституту комп'ютерних наук та штучного інтелекту; Україна; e-mail: irina.galceva@karazin.ua

Колованова Євгенія Павлівна – канд. техн. наук, Харківський національний університет імені В.Н. Каразіна, доцент кафедри кібербезпеки інформаційних систем, мереж і технологій навчально-наукового інституту комп'ютерних наук та штучного інтелекту; Харківський національний університет радіоелектроніки, доцент кафедри безпеки інформаційних технологій; Україна; e-mail: e.kolovanova@gmail.com; ORCID: https://orcid.org/0000-0002-0326-2394