КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (20), 2023
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2023.20.205219
УДК 003.26: 629.7.05

**Гнатюк Сергій Олександрович**
д.т.н., професор, науковий керівник НДЛ протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0003-4992-0564
*s.gnatyuk@nau.edu.ua*

**Поліщук Юлія Ярославівна**
аспірантка, м.н.с. НДЛ протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0002-0686-2328
*yu.polishchuk@nau.edu.ua*

**Кінзерявий Василь Миколайович**
к.т.н., доцент, доцент кафедри безпеки інформаційних технологій
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0002-7697-1503
*v.kinzeryavyy@nau.edu.ua*

**Горбаха Богдан Миколайович**
лаборант НДЛ протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID-ID: 0000-0003-0713-4426
*4591078@stud.nau.edu.ua*

**Проскурін Дмитро Петрович**
аспірантка, м.н.с. НДЛ протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID-ID: 0000-0002-2835-4279
*proskurin.d@stud.nau.edu.ua*

# ФОРМУВАННЯ ДАТАСЕТУ КРИПТОАЛГОРИТМІВ ДЛЯ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ, ЯКІ ПЕРЕДАЮТЬСЯ З РОЗВІДУВАЛЬНО-ПОШУКОВОГО БПЛА

**Анотація.** Швидкий розвиток безпілотних літальних апаратів (БПЛА) суттєво змінив проведення військових операцій та стратегії ведення війни, пропонуючи численні переваги з точки зору розвідки, спостереження та бойових можливостей. Використання БПЛА у військовій сфері забезпечує більш повну обізнаність про ситуацію, оперативну ефективність і знижує ризики для персоналу. Окрім цього, у сфері розвідки та спостереження БПЛА зробили революцію у контексті збирання розвідувальних даних. Обладнані новітніми системами обробки зображень, давачами та камерами високої роздільної здатності, вони можуть здійснювати аерофотозйомку в реальному часі, стежити за діяльністю противника та збирати важливі розвідувальні дані, не наражаючи військових на небезпеку. БПЛА дають можливість проводити довготривалі операції в умовах секретності, надаючи командирам цінну інформацію для прийняття стратегічних рішень. Проте, залишається не вирішеним питання забезпечення конфіденційності критичних даних, зібраних за допомогою БПЛА. З огляду на це, у роботі було сформовано універсальний датасет криптографічних алгоритмів, що використовує нейронну мережу для вибору оптимального алгоритму шифрування. Для формування такого датасету необхідно було оцінити швидкість роботи алгоритмів, їх криптостійкість та інші параметри. Розроблений датасет у синтезі з нейронною мережею може використовуватись для вибору оптимального криптоалгоритму залежно від умов функціонування. У подальших

дослідженнях авторами планується визначити критерії застосування сформованого датасету нейронними мережами та розробити базу знань для навчання нейронної мережі.

**Ключові слова:** БПЛА; конфіденційність; криптографія; криптоалгоритм; шифрування; передавання даних; штучна нейронна мережа; датасет.

## ВСТУП

У сучасному світі безпілотні літальні апарати (БПЛА) відіграють ключову роль у багатьох сферах діяльності людини – від сфери обслуговування та логістики до розвідки та ведення бойових дій [1]. Проте, таке застосування БПЛА вимагає надійного захисту даних, що передаються з бортового комп'ютера та зчитувальних пристроїв до пункту управління авіаційним комплексом. Зокрема, важливим є забезпечення конфіденційності як базової характеристики інформаційної безпеки (кібербезпеки).

## АНАЛІЗ ПУБЛІКАЦІЙ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

У роботі [2] було проведено аналіз криптоалгоритмів за низкою важливих критеріїв. Результати аналізу показали, що кожен криптографічний алгоритм має переваги та недоліки – тобто, не існує універсального криптоалгоритму, який здатен вирішити усі проблеми конфіденційності в БПЛА. З огляду на обмеженість ресурсів у процесі експлуатації БПЛА, існує необхідність створення універсального набору даних – так званого, датасету (бібліотеки) криптографічних алгоритмів, який зміг би розв'язувати різного роду задачі в умовах, що постійно змінюються. Крім того, актуальним і новітнім підходом на сьогодні є застосування методів штучного інтелекту (зокрема, нейронних мереж) для вибору оптимального криптоалгоритму з датасету за певними параметрами [3-5], а також інших варіацій синтезу моделей штучного інтелекту та криптографічних методів [6-8].

З огляду на це, *метою роботи* є забезпечення конфіденційності даних при передаванні з БПЛА за рахунок застосування нейронних мереж та створення універсального датасету сучасних криптографічних алгоритмів. Для ефективного формування датасету необхідно також оцінити швидкість роботи криптоалгоритмів, їх криптостійкість тощо. У перспективі розроблений датасет може бути використаний у поєднанні з нейронною мережею для вибору оптимального алгоритму шифрування в залежності від умов функціонування БПЛА.

## РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### Опис обраних алгоритмів шифрування даних

Для забезпечення захисту інформації в сучасних інформаційно-комунікаційних системах представлено велику кількість криптографічних алгоритмів, які різняться своїми базовими характеристиками (параметрами) – криптостійкість (до різних відомих методів криптоаналізу), швидкість криптографічної обробки даних, зручність апаратної реалізації тощо. З огляду на апаратні обмеження сучасних БПЛА, чинник швидкості та обсяг необхідних ресурсів для проведення шифрування інформації стають ключовими характеристиками при виборі алгоритму. Серед алгоритмів, що на думку авторів мають бути представленими в датасеті, були обрано наступні симетричні потокові та блокові

шифри: Salsa20; PANAMA; HC-256; AES; DES; Triple DES; Serpent; Blowfish; Twofish; MARS; RC2; RC6; ГОСТ 28147-89 (ДСТУ ГОСТ 28147:2009); Калина.

Розглянемо більш детально обрані *потокові алгоритми шифрування з датасету:*

**Salsa20**

Криптоалгоритм, що був спроєктованим Д. Бернштейном і представлений на конкурсі eSTREAM [9], метою якого було створення європейських стандартів шифрування даних, які передаються поштовими системами. Алгоритм став переможцем конкурсу в першій категорії (потокові шифри для програмного застосування з великою пропускною здатністю) [9]. Для реалізації необхідно створити ключ довжиною 128 або 256 біт, а також вектор ініціалізації довжиною 64 біти [9]. У самому алгоритмі використовуються такі операції [9]:

- додавання 32-бітних чисел;
- побітове додавання за модулем 2 (XOR);
- зсув бітів.

В основі цього алгоритму лежить хеш-функція на 64 байти, яка працює разом з лічильником і становить 20 циклів, що виконуються над внутрішнім станом. Недоліком Salsa20 є те, що її можна використовувати лише для захисту особистих даних, які зберігаються на ПК або на жорсткому диску [9]. Оскільки цілісність зашифрованого тексту не перевіряється, тому, для більш важливих даних необхідно використовувати автентифіковане шифрування [9].

**PANAMA**

Основні перетворення шифру PANAMA оперують 32-розрядними словами. Цей алгоритм може використовуватись для хешування інформаційних масивів великого розміру [19]. При використанні як потокового шифру та генератора псевдовипадкових послідовностей алгоритм має досить довгу процедуру ініціалізації. Галузь використання алгоритму – шифрування відеоінформації, наприклад сфера платного телебачення – у цій галузі, де інтенсивність потоку даних дуже висока і для його обробки застосовується високопродуктивний процесор, потрібен алгоритм, що найменшою мірою використовує і так надмірно завантажений процесор. Сам алгоритм базується на 544-розрядному регістрі *state* (стан) і 8192-розрядному регістрі *buffer* (буфер). Стан регістру *state* оновлюється за допомогою паралельних нелінійних перетворень. Регістр *buffer* – це LFSR, який схожий на регістр, який використовується в алгоритмі хешування SHA. Регістр *state* складається із сімнадцяти 32-розрядних слів. Стан регістрів може змінюватися за допомогою двох ітерацій [19]:

- ітерація Push приймає вхідні дані, але не генерує вихідних;
- ітерація Pull (виштовхування) не приймає вхідних даних, але генерує вихідні.

Також існує ітерація Blank Pull (порожнє виштовхування), яка аналогічна до ітерації Pull, але при цьому вихідні дані відкидаються [19].

**HC-256**

HC-256 – алгоритм потокового шифрування, розроблений У Хунцзюнем, криптографом із сінгапурського Інституту інфокомунікаційних досліджень і вперше опублікований у 2004 році [10]. 128-бітний варіант шифру був представлений на згаданому конкурсі eSTREAM. Алгоритм став одним із чотирьох фіналістів конкурсу в одній із категорій. Цей алгоритм генерує ключову послідовність довжиною $2^{128}$ біт за допомогою 256-бітного ключа і 256-бітного вектору ініціалізації. У складі шифру є дві секретні таблиці, в кожній з яких 1024 32-бітних елементи. При кожному кроці

оновлюється один елемент з таблиці за допомогою нелінійної функції зворотного зв'язку, і через кожні 2048 кроків всі елементи двох таблиць будуть оновлені. Також використовуються такі операції як побітове виключне АБО, конкатенація, зсув вліво / вправо, циклічний зсув вправо [10].

Далі, розглянемо *блокові симетричні алгоритми з датасету:*

**AES (Rijndael)**

Симетричний алгоритм блокового шифрування, фіналіст (переможець) конкурсу AES і прийнятий як американський стандарт шифрування урядом США [22]. Сам алгоритм, прийшов на заміну попереднього стандарту шифрування – DES [23]. Розмір блоку AES має фіксовану довжину у 128 біт, а розмір ключа може бути 128/192/256 біт. Дані представляються по 8 байт. Сам алгоритм включає в себе такі операції: *SubBytes* (операція підстановки, кожних 8 байт заміняються відповідно до таблиці підстановки), *ShiftRows* (зсув елементів квадрату), *MixColumns* (множення на многочлен за модулем), *AddRoundKey* (побітове додавання даних із раундовим ключем за модулем 2 (XOR)), розширення ключа. AES є досить швидким алгоритмом шифрування, що дозволяє розглядати його як гідного кандидата для застосування у сучасних інформаційно-комунікаційних системах, зокрема в системах БПЛА [22].

**DES**

Симетричний алгоритм шифрування, який був стандартом шифрування США із 1976 року до кінця 1990-х років і з часом набув міжнародного застосування в різних державах світу. Ще з часу свого розроблення, алгоритм викликав неоднозначні відгуки, оскільки містив засекречені елементи своєї структури – породжувались побоювання щодо можливості контролю з боку Агентства національної безпеки США. Алгоритм шифрує дані блоками по 64 біти, а довжина ключа складає 56 біт. У процесі шифрування використовуються такі операції: перемішування, підстановка, XOR, розширення ключа, циклічний зсув [23]. Зараз DES вважається ненадійним в основному через малу довжину ключа (56 біт) та розмір блоку (64 біти) [23], проте його використання у створеному датасеті обумовлено експериментальним дослідженням з метою порівняння його з іншими алгоритмами. У 1999 році ключ DES було публічно зламано за 22 год. 15 хв., а також доведено що DES не стійкий до лінійного криптоаналізу. Вважається, що алгоритм достатньо надійний для застосування у модифікації 3-DES, хоча існують розроблені теоретичні атаки.

**Triple DES (3DES)**

Блоковий симетричний шифр, який тричі застосовує згаданий алгоритм DES до кожного блоку даних. Створений у 1978 році на основі алгоритму DES з метою усунення головного недоліку останнього – малої довжини ключа (56 біт), що може бути зламаним (підібраним) методом перебору. Алгоритм 3DES працює в 3 рази повільніше ніж DES, але його криптостійкість є набагато більшою – час необхідний на проведення криптоаналізу 3DES є в $10^9$ разів більшим ніж для DES. Довжина ключа складає 192 біти, проте насправді його довжина складає 168 біт, оскільки як і в DES, в якому 64-розрядний ключ ділиться на 8 байтів, у кожному байті використовується тільки 7 бітів, тому насправді довжина ключа дорівнює 56 бітів [18]. Аналогічно до DES, в процесі шифрування використовуються операції перемішування, підстановки, XOR, розширення ключа, циклічний зсув. Основними перевагами алгоритму можна назвати його високу криптостійкість, проте він має низьку швидкість шифрування даних [18].

**RC2**

Алгоритм було розроблено наприкінці 1980-х років, він є власністю компанії RSA Data Security. Розробку алгоритму ініціювала і частково спонсорувала компанія Lotus, якій був потрібен стійкий алгоритм шифрування для використання в системі Lotus Notes. Стійкість алгоритму перевірена Агентством Національної Безпеки США, також було сформульовано певні рекомендації, впроваджені розробниками [21]. Шифрування відбувається блоками по 64 біти з використанням ключів змінного розміру: від 8 до 1024 бітів включно (рекомендований розмір ключа – 64 біти) [21]. Алгоритм RC2 є мережею Фейстеля, в якому виконується 18 раундів перетворень, які поділяються на 2 типи:

- раунди що змішують *(mix);*
- раунди що об'єднують *(mesh).*

Також, в алгоритмі використовуються такі операції: побітова логічна операція I, побітовий комплемент до *x*, циклічний зсув вліво на кількість бітів, що визначається таблицею підстановки, процедура розширення ключа. Згідно з дослідженнями впливу диференціального та лінійного криптоаналізу на алгоритм, було отримано наступний результат: алгоритм не уразливий до атаки методом лінійного криптоаналізу, проте він може бути теоретично розкритий методом диференціального криптоаналізу [21].

**Serpent**

Блоковий алгоритм шифрування, що був одним з фіналістів другого етапу конкурсу AES [15]. Розмір блоку складає 128 біт, довжина ключа може бути 128/192/256 біт, а сам алгоритм має 32 раунди (початково планувалося 16, але з метою протидії ще не відомим методам криптоаналізу, збільшили до 32). Serpent є SP-мережею, тобто в його складі є таблиці перестановок, а також таблиці підстановок. Алгоритм має операції з розширення ключа, лінійних перетворень, а також зворотних лінійних перетворень (при розшифруванні) [15]. При розробці та аналізі алгоритму Serpent не було виявлено будь-яких уразливостей у повній 32-раундовій версії (які і інших алгоритмів-фіналістів). На думку авторів алгоритму, для зламу шифру, необхідна нова математична теорія [15].

**Blowfish**

Блоковий симетричний алгоритм, розроблений Б. Шнайєром в 1993 році, шо не є запатентованим і вільно розповсюджується [13]. Довжина ключа в ньому може складати від 32 до 448 біт, а довжина блоку 32 біти. Алгоритм представляє собою мережу Фейстеля і, відповідно, включає такі операції як: XOR, таблиці підстановки, додавання [13]. Щодо криптостійкості алгоритму можна відмітити розроблену атаку яка дозволяла зламати 3-ітераційний Blowfish – вона спирається на факт, що операції додавання за модулем $2^{32}$ і XOR не комутативні. Успішні атаки можливі тільки через помилки реалізації. Саме тому Blowfish зарекомендував себе як надійний алгоритм, він використовується, зокрема, в SSH (транспортний рівень), PuTTY (транспортний рівень), OpenVPN тощо [14].

**Twofish**

Симетричний алгоритм блокового шифрування, розроблений групою фахівців на чолі з Б. Шнайером, який також (як і Serpent) був серед п'яти фіналістів другого етапу конкурсу AES. Алгоритм розроблений на основі Blowfish, SAFER і Square [12], розмір блоку складає 128 біт, а довжина ключа – 256 біт при кількості раундів 16. Однією з його особливостей є таблиці перестановки, які формуються в залежності від ключа. Сам алгоритм був реалізований як змішана мережа Фейстеля з 4 гілками, які модифікують одна одну з використанням криптоперетворень Адамара. Алгоритм включає в себе такі функції як циклічний зсув на 1 біт, а також функцію відбілювання. Вивчення Twofish зі

скороченими числом раундів показало, що алгоритм володіє великим запасом стійкості, і, в порівнянні з іншими фіналістами конкурсу AES, він виявився найстійкішим. Проте його незвичайна структура і відносна складність породили деякі сумніви щодо якості цієї стійкості – саме це і зіграло проти нього на конкурсі AES [12].

**MARS**

Блоковий симетричний алгоритм, розроблений корпорацією IBM. За результатами конкурсу AES, MARS теж вийшов у фінал, але поступився Rijndael. На даний момент MARS поширюється під Royalty Free ліцензією [11]. Розмір блоку в алгоритмі складає 128 біт, а розмір ключа може бути від 128 до 448 біт (має бути кратним 32 бітам). У процесі шифрування алгоритм використовує такі операції: додавання / віднімання, виключаюче АБО, таблиці підстановок, фіксований циклічний зсув і залежний від даних циклічний зсув, множення за модулем $2^{32}$, розширення ключа [11]. За заявою IBM, в алгоритм MARS вкладено 25-річний криптоаналітичний досвід компанії і, поряд з високою криптографічною стійкістю, шифр допускає ефективну реалізацію навіть в таких обмежених рамках, які характерні для смарт-карт, що дозволяє його використовувати в БПЛА. З точки зору криптоаналізу, на цей момент немає ефективних атак на даний алгоритм, проте він має декілька слабких сторін, зокрема:

- підключі з великою кількістю повторюваних нулів або одиниць можуть привести до ефективних атак на MARS, так як на їх підставі будуть згенеровані слабкі підключі.
- два молодших біта, що використовуються при перемножуванні, завжди рівні одиниці, тобто є два вхідних біта, які незмінні в ході процесу множення на ключ, а також два вихідних біти, незалежних від ключа [11].

**RC6**

Симетричний блоковий шифр, один з п'яти фіналістів конкурсу AES, який є пропрієтарним алгоритмом, запатентованим RSA Security, однак дія патентів закінчилась і зараз алгоритм знаходиться у відкритому доступі. У той же час, RC6 залишається зареєстрованою торговою маркою RSA [11]. Довжина ключа може сягати від 0 до 255 біт, а довжина блоку складає 128 біт. Алгоритм працює з такими операціями: циклічні зсуви, операція XOR, розширення ключа тощо. З точки зору безпеки жодних ефективних атак не було виявлено. Були виявлені атаки тільки проти спрощених версій алгоритму, тобто алгоритму зі зменшеною кількістю раундів [11].

**ГОСТ 28147-89**

Радянський стандарт симетричного шифрування, введений в 1990 році [20]. За наявною інформацією, його історія набагато давніша. Алгоритм, покладений згодом в основу стандарту, народився, імовірно, в надрах 8-го Головного управління КДБ СРСР, швидше за все, в одному з підвідомчих йому закритих НДІ, ймовірно, ще в 1970-х роках в рамках проєктів створення програмних та апаратних реалізацій шифру для різних комп'ютерних платформ. З моменту опублікування цього шифру на ньому стояв обмежувальний гриф «Для службового користування», і формально шифр був оголошений «повністю відкритим» тільки в травні 1994 року. У 2009 році ГОСТ 28147-89 з певними змінами був перезатверджений в Україні під назвою ДСТУ ГОСТ 28147:2009 [20]. Довжина ключа складає 256 біт, кількість раундів складає 16 або 32 (у залежності від режиму роботи), а довжина блоку дорівнює 64 біти. В алгоритмі використовуються такі операції: таблиці підстановок, побітовий циклічний зсув вліво, накладання ключа, додавання за модулем 2. Таблиці підстановок можуть бути секретними та змінюватися, що забезпечує алгоритму високий рівень криптостійкості [20], проте він

вважається морально застарілим і не рекомендований до використання в Україні з 2015 року (у тому числі, через початок російсько-української війни).

**Калина (ДСТУ 7624:2014)**

Блоковий симетричний шифр, що на сьогодні є національним стандартом України ДСТУ 7624:2014 «Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення» [16]. Стандарт набрав чинності з 01 липня 2015 року наказом Мінекономрозвитку від 2 грудня 2014 року №1484. Розмір блоку та кількість раундів є залежними від розміру ключа (див. Табл.1).

*Таблиця 1*

**Характеристики алгоритму Калина**

| Довжина блоку | Довжина ключа | Кількість раундів |
|---|---|---|
| 128 | 128 | 10 |
| | 256 | 14 |
| 256 | 256 | 14 |
| | 512 | 18 |
| 512 | 512 | 18 |

У самому алгоритмі використано операції перетворення, таблиці підстановки і перестановки, розгортання ключа, додавання за модулем 2, лінійні перетворення, пре- та постзабілювання [16]. Дослідження криптостійкості алгоритму показали лише деякі ефективні атаки на скорочені версії алгоритму, проте вони не є практичними.

**Вибір типу нейронної мережі для визначення криптоалгоритму**

Штучні нейронні мережі (ШНМ) розглядаються як інструменти, які можуть допомогти аналізувати причинно-наслідкові зв'язки в складних системах [24]. Саме тому, прийнято рішення проаналізувати саме цей підхід для вибору криптографічного алгоритму зі створеного датасету. Типовий алгоритм роботи ШНМ показаний на рис.1:

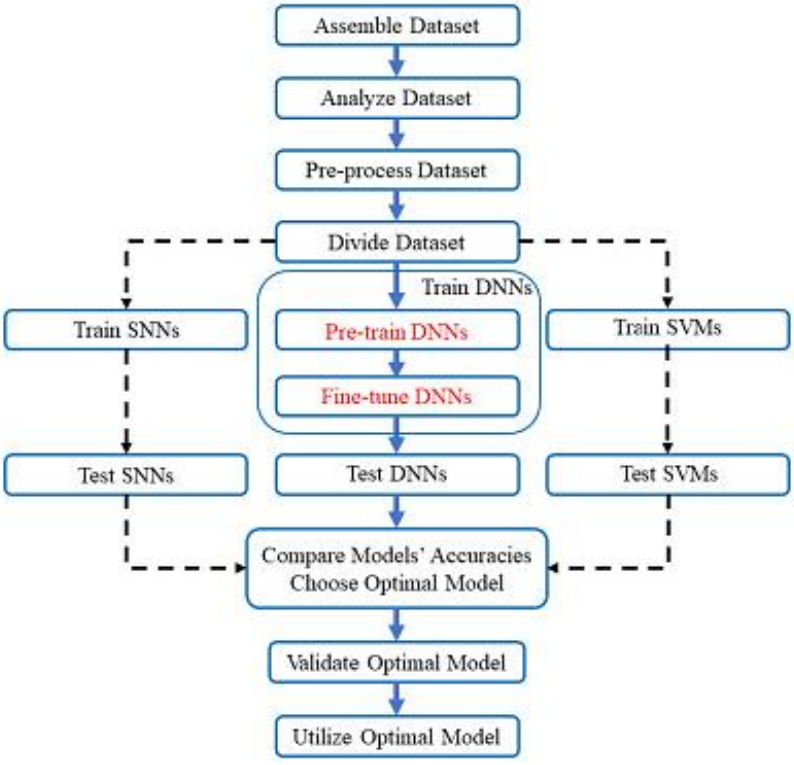


*Рис. 1. Алгоритм роботи ШНМ*

Розглянемо можливість навчання глибокої нейронної мережі (DNN) з багатьма прихованими шарами, як у нейронній системі людини [25]. Успіх нейронної мережі попереднього покоління обмежується SNN з 1 або 2 прихованими шарами, тому що навчання DNN нелегке, а кінцева точність зазвичай гірша, ніж у SNN [25]. Труднощі навчання DNN полягають у зникненні градієнтів із збільшенням номера прихованого шару, тобто глибини мережі, і пасток поганих локальних мінімумів. Глибока нейронна мережа подібна до дрібної нейронної мережі за структурою, але має більше прихованих шарів і більш очевидну ієрархічну структуру. DNN можна сприймати як оновлення SNN. Але, тим не менш, розвиток машинного навчання за останні роки зробив DNN придатним для навчання та надав деякі корисні інструменти, наприклад, попереднє навчання з RBM та SAE, для роботи з невеликим набором даних. З іншого боку, природа поставленого завдання також вказує на доцільність застосування DNN з невеликим набором даних: DNN, які використовуються для розпізнавання зображень, зазвичай більше $10^4$ вхідних змінних (наприклад, невелике зображення розміром 100×100 пікселів потребує $10^4$ вхідних змінних) і необхідно визначити величезну кількість параметрів (наприклад, $10^6$ і більше параметрів). Тому необхідні великі набори даних, але DNN для поставленого завдання зазвичай не мають менше 100 вхідних змінних і потрібно визначити менше параметрів, тому малих / вузьких DNN (кілька прихованих шарів і невелика кількість нейронів у кожному шарі) може бути достатньо [17].

Потенціал застосування DNN із невеликими наборами даних для визначення криптографічного алгоритму очевидний: великі проблеми регресії/класифікації, які раніше розглядалися традиційними методами машинного навчання (наприклад, SNN, SVM тощо) з невеликим набором даних може бути розв’язано за допомогою DNN з вищою точністю та кращою продуктивністю узагальнення [17]. У цьому дослідженні використано прогнозування SCS як приклад, щоб показати, що попереднє навчання SAE є ефективним методом роботи з невеликими наборами даних у регресії DNN, а повністю підключена DNN демонструє вищу точність і кращу продуктивність узагальнення, ніж SNN і SVM [17]. Робочий процес наступного аналізу включатиме:

- складання та аналіз набору даних;
- попередню обробку та розподіл набору даних для навчання/тестування;
- навчання SVM/SNN/DNN;
- тестування навчених моделей машинного навчання;
- порівняння моделей;
- точність і вибір оптимальної моделі;
- застосування оптимальної моделі для прогнозування.

Навчання DNN включає два етапи: попереднє навчання та тонке налаштування. Машина опорних векторів і неглибока нейронна мережа також будуть навчені та протестовані в нашому аналізі для підтвердження переваги точності DNN. Після навчання SVM, SNN і DNN їх точність навчання/тестування підсумовується та порівнюється. Попередній процес PCA має негативний вплив на точність навчання та тестування, що може бути спричинено втратою нелінійної інформації. Решта цього дослідження базується на наборі даних без попередньої обробки. Повністю зв’язана DNN, яка складається з 3 або більше прихованих рівнів, показує свою перевагу перед неглибокою ШНМ та машиною допоміжних векторів у тому, що вона може досягти вищої точності прогнозування та кращої продуктивності узагальнення [17]. Хоча DNN із великими наборами даних є найкращим вибором, DNN із малими наборами даних і попереднім навчанням може бути розумним вибором, коли великі набори даних

недоступні. У даній роботі невеликі набори даних є звичайним явищем, і проблеми, які потрібно вирішити, мають менше вхідних змінних.

## ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

Для проведення експериментальних досліджень було розроблено спеціальну бібліотеку, в якій були програмно реалізовані усі наведені алгоритми. Для розроблення було використано мову програмування Python, а для реалізації алгоритмів шифрування використано бібліотеку *Pycryptodome,* яка має в собі заготовлені алгоритми шифрування (у форматі *.pyc* скомпільовані на вихід файли, написані на мові Python). Сам застосунок представляє собі утиліту з командним інтерфейсом, де користувач має змогу вказати файли для шифрування, а також алгоритм що буде використано і режим його роботи.

Робота застосунку представляє собою наступну послідовність:

1. Перед початком роботи необхідно зайти у віртуальне середовище *venv* щоб підвантажити алгоритми з бібліотеки *Pycryptodome;*

2. Якщо при запуску утиліти не було вказано жодних аргументів, або утиліта запущена з прапором *-h,* користувач отримує коротку довідку про використання утиліти у наступному форматі:

```
python main.py -[enc/dec] [Enc/Dec method] [Mode] [Key] [InFile] [OutFile]
To see supported methods use -> python main.py -s
To see supported modes use -> python main.py -m
```

де, [enc/dec] – зашифрування чи розшифрування, [Enc/Dec method] – вибір алгоритму для роботи, [Mode] – режим роботи алгоритму, [Key] – ключ, [InFile] – файл на вході, [OutFile] – файл на виході.

Також, користувач має можливість запустити утиліту з прапором *-s* або *-m,* щоб побачити доступні алгоритми та режими роботи відповідно.

3. Після успішного запуску процедури зашифрування / розшифрування файлу, користувач може знайти оброблені файлі в кореневій папці програми.

Для зручності реалізації та подальшого розширення програмної реалізації датасету, в директорії *Libs/CryptoAbstract.py* було створено абстрактні класи (рис. 2), які в подальшому наслідуються при реалізації кожного окремого криптографічного алгоритму, таким чином, що кожен алгоритм має однаковий інтерфейс входу та виходу даних. Також, за шляхом *Libs/env.py* знаходиться файл в якому зберігаються словники з доступними алгоритмами для роботи, а також режимами роботи.

ISSN 2663 - 4023

```
27          pass
28
29      @abstractmethod
30      def _decryptBlock(self, block128):
31          pass
32
33      @abstractmethod
34      def encfile(self, file_path_in, file_path_out):
35          pass
36
37      @abstractmethod
38      def decfile(self, file_path_in, file_path_out):
39          pass
40
41
42  class Crypto192(ABC):
43
44      @abstractmethod
45      def _encryptblock(self, block128):
46          pass
47
48      @abstractmethod
49      def _decryptBlock(self, block128):
50          pass
51
52      @abstractmethod
53      def encfile(self, file_path_in, file_path_out):
54          pass
55
56      @abstractmethod
```

*Рис. 2. Реалізація абстрактних класів (рядки 27-56)*

Кожен алгоритм та різні його версії, зберігаються в директорії з відповідною назвою, та реалізовані у вигляді класу з наступними методами:

- ініціалізація;
- зашифрування/дешифрування блоку;
- зчитування файлу побайтово для подальшого шифрування даних.

Програмна реалізація датасету створена таким чином, щоб в подальшому її можна було досить швидко модифікувати та доповнювати.

Для проведення експерименту було виконано шифрування файлів різного розміру (від декількох кілобайтів до декількох гігабайтів). Кожен файл шифрувався кожним алгоритмом до 10 разів (для підвищення точності дослідженні).

Експерименти проводилися на такій апаратній платформі: процесор – Intel core i7 (9th gen); відеокарта – Nvidia Geforce GTX 1060; оперативна пам’ять – 16GB RAM DDR3; операційна система – Windows 10.

Результати експерименту було оброблені та представлені в датасеті у вигляді кількісної оцінки – було створено порівняльну таблицю (табл. 2) з експертними оцінками (дані про криптостійкість *(CRS)* і запас криптостійкості *(FCRS)* бралися з відкритих джерел, швидкість шифрування *(ECR)* і розширення ключа *(KEA)* перевірялася

експериментальним дослідженням авторами). У табл. 2 числа від 1 до 10 визначають ефективність алгоритму за вказаним критерієм (1 – найгірша оцінка, а 10 - найкраща).

*Таблиця 2*

**Порівняння криптоалгоритмів датасету**

| *Алгоритм* | *CRS* | *FCRS* | *ECR* | *KEA* |
|---|---|---|---|---|
| Salsa20 (П) | 7 | 8 | 9 | 6 |
| Panama (П) | 6 | 7 | 7 | - |
| HC-256 (П) | 6 | 4 | 7 | - |
| DES (Б) | 2 | 2 | 4 | 4 |
| Triple DES (Б) | 7 | 5 | 2 | 2 |
| RC2 (Б) | 5 | 4 | 6 | 5 |
| AES (Б) | 9 | 8 | 10 | 7 |
| RC6 (Б) | 8 | 7 | 10 | 7 |
| Blowfish (Б) | 6 | 7 | 7 | 6 |
| Twofish (Б) | 9 | 7 | 7 | 7 |
| Serpent (Б) | 9 | 9 | 7 | 8 |
| ГОСТ (Б) | 7 | 6 | 6 | 10 |
| Калина (Б) | 10 | 9 | 9 | 7 |

Отже, було створено датасет криптоалгоритмів, що може бути використаний для забезпечення конфіденційності даних під час передавання з БПЛА. Даний датасет може також бути використаний ШНМ для вибору того чи іншого алгоритму шифрування в залежності від заданих вимог. Далі авторами планується обрати критерії застосування даного датасету ШНМ, після чого буде створена база знань для навчання ШНМ. У подальшому буде також розроблено метод підвищення продуктивності ШНМ.

## ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Таким чином, у роботі був сформований відкритий датасет криптографічних алгоритмів для забезпечення ефективності захисту інформації при передачі з БПЛА на основі застосування ШНМ. Наразі датасет містить низку блокових та потокових криптоалгоритмів (опис алгоритмів наведено у статті), криптостійкість та швидкість роботи яких визначена на основі результатів наукових досліджень різних авторів.

У наступних роботах планується удосконалення датасету: додавання більшої кількості алгоритмів (у тому числі, розроблених авторами), оптимізація програмної реалізації алгоритмів, проведення додаткових експериментів, спрямованих на формування більш повної бази алгоритмів для використання у БПЛА, а також взаємодії в контексті ШНМ. Крім того, планується також дослідити інші задачі (у контексті функціонування БПЛА), що можуть бути розв’язані сучасними засобами ШНМ.

## ACKNOWLEDGEMENT

Робота виконана у рамках науково-дослідного проєкту «Інтелектуалізована система захищеного передавання пакетних даних на базі розвідувально-пошукового безпілотного літального апарату» (№0122U002361), що фінансується Міністерством освіти і науки України протягом 2022-2024 років.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1 Du, X., Tang, Y., Gou, Y., & Huang, Z. (2021). Data Processing and Encryption in UAV Radar. У 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE. https://doi.org/10.1109/imcec51613.2021.9482373.

2 Гнатюк, С., Кінзерявий, В., Поліщук, Ю. (2022). Аналіз методів забезпечення конфіденційності даних, які передаються з БПЛА. *Кібербезпека: освіта, наука, техніка*, *1*(17), 167-186.

3 Dong, T., & Huang, T. (2020). Neural Cryptography Based on Complex-Valued Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(11), 4999–5004. https://doi.org/10.1109/tnnls.2019.2955165.

4 Jhajharia, S., Mishra, S., & Bali, S. (2013). Public key cryptography using neural networks and genetic algorithms. У *2013 Sixth International Conference on Contemporary Computing (IC3)*. IEEE. https://doi.org/10.1109/ic3.2013.6612177.

5 Xiao, Y., Hao, Q., & Yao, D. D. (2019). Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers. У *2019 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE. https://doi.org/10.1109/dsc47296.2019.8937659.

6 Niemiec, M., Mehic, M., & Voznak, M. (2018). Security Verification of Artificial Neural Networks Used to Error Correction in Quantum Cryptography. У *2018 26th Telecommunications Forum (TELFOR)*. IEEE. https://doi.org/10.1109/telfor.2018.8612006.

7 Das, G., & Kule, M. (2022). A New Error Correction Technique in Quantum Cryptography using Artificial Neural Networks. У *2022 IEEE 19th India Council International Conference (INDICON)*. IEEE. https://doi.org/10.1109/indicon56171.2022.10040091.

8 Schmidt, T., Rahnama, H., Sadeghian, A. (2008). A review of applications of artificial neural networks in cryptosystems, *World Automation Congress.* Waikoloa, HI.

9 Bernstein, D. J. (б. д.). The Salsa20 Family of Stream Ciphers. У *Lecture Notes in Computer Science* (с. 84–97). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68351-3_8.

10 Wu, H. (2004). A New Stream Cipher HC-256. У *Fast Software Encryption* (с. 226–244). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-25937-4_15.

11 Kelsey, J., Kohno, T., & Schneier, B. (2001). Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. У *Fast Software Encryption* (с. 75–93). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44706-7_6.

12 Ferguson, N., Hall, C., Kelsey, J., Wagner, D., Whiting, D., & Schneier, B. (1999). *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*. Wiley.

13 Gonzalez, T. (2007). A reflection attack on Blowfish. http://karbalus.free.fr/sat/docsat/PaperGonzalezTom.pdf.

14 Kara, O., & Manap, C. (б. д.). A New Class of Weak Keys for Blowfish. У *Fast Software Encryption* (с. 167–180). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74619-5_11.

15 Anderson, R., Biham, E., Knudsen, L. (2000). Serpent: A Proposal for the Advanced Encryption Standard. https://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf

16 ДСТУ 7624:2014 «Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення». http://online.budstandart.com/ua/catalog/doc-page?id_doc=65314

17 Bhadeshia, H. K. (1999). Neural Networks in Materials Science. *ISIJ International*, *39*(10), 966-979.

18 Thakur, J., Kumar, N. (2011). DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. *International Journal of Emerging Technology and Advanced Engineering*, *1*(2), 6-12.

19 Panama Stream cipher. http://nightcrowlwing.blogspot.com/2012/10/panama-stream-cipher.html

20 Совин, Я., Хома, В., Наконечний, Ю., Стахів, М. (2019). Ефективна імплементація та порівняння швидкодії шифрів «Калина» та ГОСТ 28147-89 за використання векторних розширень SSE, AVX та AVX-512. *Захист інформації, 21*(4), 207-223. https://doi.org/10.18372/24107840.21.14266

21 RC2. Block cipher with symmetric secret key. (2020). http://www.crypto-it.net/eng/symmetric/rc2.html

22 NIST. Advanced Encryption Standard (AES). (2021). https://csrc.nist.gov/publications/detail/fips/197/final

23 Davis, R. (1978). The data encryption standard in perspective. *IEEE Communications Society Magazine, 16*(6), 5-9. https://doi.org/10.1109/MCOM.1978.1089771

24 Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. *The Knowledge Engineering Review*, 13(4), 409-412.

25 Juracy, L.R., Garibotti, R., Moraes, F.G. (2023). *From CNN to DNN Hardware Accelerators: A Survey on Design.* Exploration, Simulation, and Frameworks.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (20), 2023
ISSN 2663 - 4023

**Sergiy O. Gnatyuk**
DSc, Professor, Scientific Advisor of NAU Cybersecurity R&D Lab
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0003-4992-0564
*s.gnatyuk@nau.edu.ua*

**Yuliia Ya. Polishchuk**
PhD student, Junior Researcher of NAU Cybersecurity R&D Lab
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0002-0686-2328
*yu.polishchuk@nau.edu.ua*

**Vasyl M. Kinzeryavyy**
PhD, Associate Professor, Associate Professor of IT-Security Academic Department
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0002-7697-1503
*v.kinzeryavyy@nau.edu.ua*

**Bohdan M. Horbakha**
Laboratory Assistant of NAU Cybersecurity R&D Lab
National Aviation University, Kyiv, Ukraine
ORCID-ID: 0000-0003-0713-4426
*4591078@stud.nau.edu.ua*

**Dmytro P. Proskurin**
PhD student, Junior Researcher of NAU Cybersecurity R&D Lab
National Aviation University, Kyiv, Ukraine
ORCID-ID: 0000-0002-2835-4279
*proskurin.d@stud.nau.edu.ua*

# FORMATION OF A DATASET OF CRYPTOGRAPHIC ALGORITHMS FOR ENSURING DATA CONFIDENTIALITY TRANSFERRED FROM RECONNAISSANCE AND SEARCH UAV

**Abstract.** The rapid development of unmanned aerial vehicles (UAV) has significantly changed the conduct of military operations and warfare strategies, offering numerous advantages in terms of intelligence, surveillance and combat capabilities. The use of UAV in the military sphere provides more complete situational awareness, operational efficiency and reduces risks to personnel. In addition, in the field of intelligence and surveillance, UAV have revolutionized the context of intelligence gathering. Equipped with the latest image processing systems, sensors and high-resolution cameras, they can conduct real-time aerial photography, monitor enemy activity and gather critical intelligence without putting the military at risk. UAV make it possible to conduct long-term operations in conditions of secrecy, providing commanders with valuable information for making strategic decisions. However, the issue of ensuring the confidentiality of critical data collected using UAV remains unresolved. With this in mind, in this paper universal dataset of cryptographic algorithms was created, it uses a neural network to select the optimal encryption algorithm. To form such a dataset, it was necessary to evaluate the speed of the crypto algorithms, their cryptographic security and other parameters. The developed dataset in synthesis with a neural network can be used to select the optimal crypto algorithm depending on the operating conditions. In further research, the authors plan to determine the criteria for using the generated dataset by neural networks and develop a knowledge base for neural network training.

**Keywords:** UAV; confidentiality; cryptography; crypto algorithm; encryption; data transfer; neural network; dataset.

## REFERENCES

1 Du, X., Tang, Y., Gou, Y., & Huang, Z. (2021). Data Processing and Encryption in UAV Radar. У 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE. https://doi.org/10.1109/imcec51613.2021.9482373.

2 Hnatiuk, S., Kinzeriavyi, V., Polishchuk, Yu. (2022). Analiz metodiv zabezpechennia konfidentsiinosti danykh, yaki peredaiutsia z BPLA. *Kiberbezpeka: osvita, nauka, tekhnika*, *1*(17), 167-186.
3 Dong, T., & Huang, T. (2020). Neural Cryptography Based on Complex-Valued Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(11), 4999–5004. https://doi.org/10.1109/tnnls.2019.2955165.
4 Jhajharia, S., Mishra, S., & Bali, S. (2013). Public key cryptography using neural networks and genetic algorithms. У *2013 Sixth International Conference on Contemporary Computing (IC3)*. IEEE. https://doi.org/10.1109/ic3.2013.6612177.
5 Xiao, Y., Hao, Q., & Yao, D. D. (2019). Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers. У *2019 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE. https://doi.org/10.1109/dsc47296.2019.8937659.
6 Niemiec, M., Mehic, M., & Voznak, M. (2018). Security Verification of Artificial Neural Networks Used to Error Correction in Quantum Cryptography. У *2018 26th Telecommunications Forum (TELFOR)*. IEEE. https://doi.org/10.1109/telfor.2018.8612006.
7 Das, G., & Kule, M. (2022). A New Error Correction Technique in Quantum Cryptography using Artificial Neural Networks. У *2022 IEEE 19th India Council International Conference (INDICON)*. IEEE. https://doi.org/10.1109/indicon56171.2022.10040091.
8 Schmidt, T., Rahnama, H., Sadeghian, A. (2008). A review of applications of artificial neural networks in cryptosystems, *World Automation Congress.* Waikoloa, HI.
9 Bernstein, D. J. (б. д.). The Salsa20 Family of Stream Ciphers. У *Lecture Notes in Computer Science* (с. 84–97). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68351-3_8.
10 Wu, H. (2004). A New Stream Cipher HC-256. У *Fast Software Encryption* (с. 226–244). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-25937-4_15.
11 Kelsey, J., Kohno, T., & Schneier, B. (2001). Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. У *Fast Software Encryption* (с. 75–93). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44706-7_6.
12 Ferguson, N., Hall, C., Kelsey, J., Wagner, D., Whiting, D., & Schneier, B. (1999). *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*. Wiley.
13 Gonzalez, T. (2007). A reflection attack on Blowfish. http://karbalus.free.fr/sat/docsat/PaperGonzalezTom.pdf.
14 Kara, O., & Manap, C. (б. д.). A New Class of Weak Keys for Blowfish. У *Fast Software Encryption* (с. 167–180). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74619-5_11.
15 Anderson, R., Biham, E., Knudsen, L. (2000). Serpent: A Proposal for the Advanced Encryption Standard. https://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf
16 DSTU 7624:2014 «Informatsiini tekhnolohii. Kryptohrafichnyi zakhyst informatsii. Alhorytm symetrychnoho blokovoho peretvorennia». http://online.budstandart.com/ua/catalog/doc-page?id_doc=65314
17 Bhadeshia, H. K. (1999). Neural Networks in Materials Science. *ISIJ International*, *39*(10), 966-979.
18 Thakur, J., Kumar, N. (2011). DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. *International Journal of Emerging Technology and Advanced Engineering*, *1*(2), 6-12.
19 Panama Stream cipher. http://nightcrowlwing.blogspot.com/2012/10/panama-stream-cipher.html
20 Sovyn, Ya., Khoma, V., Nakonechnyi, Yu., Stakhiv, M. (2019). Efektyvna implementatsiia ta porivniannia shvydkodii shyfriv «Kalyna» ta HOST 28147-89 za vykorystannia vektornykh rozshyren SSE, AVX ta AVX-512. *Zakhyst informatsii*, *21*(4), 207-223. https://doi.org/10.18372/24107840.21.14266
21 RC2. Block cipher with symmetric secret key. (2020). http://www.crypto-it.net/eng/symmetric/rc2.html
22 NIST. Advanced Encryption Standard (AES). (2021). https://csrc.nist.gov/publications/detail/fips/197/final
23 Davis, R. (1978). The data encryption standard in perspective. *IEEE Communications Society Magazine, 16*(6), 5-9. https://doi.org/10.1109/MCOM.1978.1089771
24 Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. *The Knowledge Engineering Review*, 13(4), 409-412.
25 Juracy, L.R., Garibotti, R., Moraes, F.G. (2023). *From CNN to DNN Hardware Accelerators: A Survey on Design.* Exploration, Simulation, and Frameworks.