КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (17), 2022
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2022.17.167186
УДК 003.26: 629.7.05

Гнатюк Сергій Олександрович
д.т.н., професор, заступник декана факультету кібербезпеки, комп'ютерної та програмної інженерії, керівник НДЛ протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0003-4992-0564
s.gnatyuk@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

Поліщук Юлія Ярославівна
PhD аспірант, 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

Нечипорук Олена Петрівна
д.т.н., доцент, професор кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0001-8203-7998
olena.nechyporuk@npp.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

АНАЛІЗ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ, ЯКІ ПЕРЕДАЮТЬСЯ З БПЛА

Анотація. Стрімкий розвиток безпілотних літальних апаратів (БПЛА), а також розширення їх функціоналу, зумовило підвищення вимог до безпеки та надійності передавання даних. В умовах ведення військових дій, а також при виконані операцій, під час яких збираються конфіденційні дані, захист такоих даних є першочерговим завданням. Практичний стан проведення повітряної розвідки в зоні бойових дій демонструє нагальну потребу створення БПЛА, що здатні виконувати польотне завдання та аеророзвідку в режимі встановлених радіоперешкод, а також підкреслює важливість забезпечення конфіденційності даних про цільові об'єкти, що передаються оптичним каналом для реалізації їх обробки в автоматизованих системах. У цій статті проведено огляд та порівняльний аналіз сучасних криптоалгоритмів, які використовуються для забезпечення конфіденційності даних під час їх передавання радіоканалом з БПЛА до наземних об'єктів. Для багатокритеріального порівняння алгоритмів використовувалась система критеріїв, подібна до конкурсів AES та NESSIE, що пов'язані з розмірами блоку і ключа, режимами роботи, швидкістю шифрування, вимогами до пам'яті та стійкістю до криптоаналізу. Аналіз показав, що кожен криптоалгоритм має переваги та недоліки, а універсальні алгоритми, здатні вирішити усі проблеми конфіденційності в БПЛА, є відсутніми. З огляду на обмеженість ресурсів у процесі експлуатації БПЛА, необхідно створити датасет криптографічних алгоритмів, які могли б розв'язувати різного роду задачі у різних умовах. Саме цим дослідженням і буде присвячена подальша робота авторів у рамках виконуваного наукового проєкту.

Ключові слова: БПЛА; конфіденційність даних; криптографія; криптографічний алгоритм; шифрування; передавання даних; датасет.

ВСТУП

Поява безпілотних літальних апаратів (БПЛА) зробила можливим виконання надскладних та важливих завдань, таких як пошук і порятунок людей, спостереження за рухом і геодезія, виконання розвідки в умовах бойових дій тощо. Проте, успіх цих операцій визначається не лише можливістю зробити відео чи фото з місця події, але, що є більш важливим, передавання даних на наземні системи та забезпечення конфіденційності цих даних. Крім того, слід пам'ятати, що БПЛА мають обмеження щодо розміру, ваги та потужності, які обмежують кількість корисного навантаження на борту [1]. Таким чином, вирішальне значення для безпечного передавання даних з борту БПЛА на наземні системи є першочерговим завданням, виходячи із загроз, які актуальні у нашій державі в умовах воєнного стану. Існує кілька методів захисту комунікацій (систем зв'язку), одним із них є використання алгоритмів шифрування.

Загалом шифрування можна визначити як процес застосування перетворень до оригінального повідомлення (яке називається відкритим текстом) з метою створення зашифрованого тексту, який неавторизовані користувачі не можуть прочитати або мати до нього доступ. Існує два типи криптографічних алгоритмів – це симетричні та асиметричні криптоалгоритми. У симетричному шифруванні ключі зашифрування та розшифрування збігаються або є трансформацією один одного. Асиметричне шифрування використовує різні ключі для зашифрування та розшифрування. Проте, цей підхід не рекомендовано для БПЛА через їхню високу складність і меншу швидкість порівняно з симетричними алгоритмами [2]. Однак симетричні алгоритми вимагають додаткових безпечних каналів зв'язку для обміну ключами, а асиметричне шифрування – ні. Шифрування може мати прямий вплив на захист даних, тому вкрай важливо вибрати не тільки надійний, але й не ресурсоємний алгоритм.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Питання вибору криптоалгоритму для вирішення проблеми безпечного передавання інформаціїї було розглянуто у багатьох роботах науковців. Авторами [3] проведене порівняння чотирьох найпоширеніших криптоалгоритмів з симетричним ключем: AES, DES, CAST 128 і Blowfish. Метою дослідження було порівняти поведінку та продуктивність алгоритму, у разі використання даних різного розміру, головним завданням було вивчення продуктивності алгоритмів за різних налаштувань. Порівняння здійснювалось на основі таких параметрів: швидкість, розмір блоку та розмір ключа. Виходячи із висновку дослідників, було продемонстровано, що алгоритм AES забезпечує найкращий захист. Експеримент показав, що в обох режимах DES дає сильний лавинний ефект, а AES і CAST 128 дають значну зміну терміну перевірки цілісності порівняно з іншими алгоритмами.

У [4] статті проведено порівняльний аналіз алгоритму AES з різними режимами роботи (блоковий шифр) і алгоритму RC4 (потоковий шифр) з точки зору процесорного часу, часу шифрування, використання пам'яті та пропускної здатності за різних параметрів, таких як змінний розмір ключа та змінний розмір пакета даних. Експерименти показали, що RC4 є швидким і енергоефективним для шифрування. На основі аналізу, проведеного в рамках цього дослідження, RC4 кращий ніж AES.

У статті [5] автори запропонували систему, яка забезпечує цілісність зображень за допомогою AES та Rivest-Shamir-Adleman (RSA). Вони порівняли два алгоритми, обчисливши розмір буфера різних зображень. Зрештою вони з'ясували, що AES більш ефективний для шифрування.

Автори [6] провели аналіз продуктивності алгоритмів DES, 3DES, AES і Blowfish з різними розмірами та носіями. Результати показують, що Blowfish був найшвидшим алгоритмом; однак автори зазначають, що безпека була здебільшого проігнорована і, що її слід розглянути в першу чергу. Автори [7] також провели аналіз продуктивності AES і DES і дійшли висновку, що AES є ефективнішим для захисту каналів зв'язку між вузлами, ніж DES.

Дослідження [8] було спрямоване на порівняння алгоритмів шифрування з точки зору продуктивності та забезпечення безпеки для БПЛА. Порівнювались алгоритми OTP, AES, DES і RSA. У цій статті було проведено порівняння продуктивності OTP, легкого та надійного алгоритму, з продуктивністю AES, DES і RSA. Результати показують, що OTP працює краще, ніж інші алгоритми з точки зору використання оперативної пам'яті, часу обробки та часу шифрування.

У науковій статті [9], поєднуючи шифрування даних за допомогою Elliptic Curve Cryptography (ECC) і The Diffie–Hellman (DH) Algorithm, проводилися випробування обміну ключами за допомогою криптографії з відкритим ключем. Результат випробування показав кращі результати ніж RSA та інші алгоритми.

У роботі [10] проведено порівняння симетричних шифрів, таких як AES, AES-GCM, HMAC, CHACHA20, CHACHA-POLY, POLY 1305 за різними критеріями.

У статті [11] було розроблено програми для шифрування даних між БПЛА та GCS за допомогою алгоритму AES-128. Метою дослідження було зробити дані, надіслані БПЛА наземній системі управління, недоступними для зловмисника. Також проведено тестування атаки грубої сили і виявлено, що зламати ключ AES-128 біт неможливо, тому безпека за допомогою криптоалгоритму AES-128 все ще доцільна для використання сьогодні (до появи суперкомп'ютера, швидшого ніж $10{,}51\times10^{15}$ Flops).

У [12] авторами пропонується гібридна криптографічна схема безпеки, яка має прості обчислення, але високий рівень безпеки. Запропонований алгоритм має багатошарове шифрування з використанням AES-256, ECC і SHA256, які забезпечують високу аутентифікацію даних і сервіси шифрування для кожного вузла. Запропонована схема є ефективною з точки зору часу обробки, тому вона не буде суттєво впливати на дані датчиків з дрона у реальному часі. Результати експерименту показують, що запропонована система досягає 88,61 мс для шифрування повідомлень за допомогою пристрою Raspberry Pi.

У науковому дослідженні [13] проведено порівняння алгоритмів ECC і RSA в пристроях з обмеженими ресурсами. Було порівняно алгоритм криптографії на основі ECC із розміром ключа 160 біт і алгоритм Райвеста-Шаміра-Адлемана (RSA) із розміром ключа 1024 біт. У результаті цього дослідження показано, що використання ECC у пристроях з обмеженими ресурсами має переваги перед RSA, але ECC потребує продовження вдосконалення щоб задовольнити обмеження нових мікросхем.

Авторами в роботі [14] також проведене порівняння криптоалгоритмів на основі ECC і алгоритму RSA.

У статті [15] було запропоновано розроблений алгоритм MOD-ECDH, а також проведене порівняння з іншими алгоритмами, такими як ECDH, RSA та ECS. Детально описано результати емпіричного аналізу та моделювання застосування алгоритмів ECDH та MOD-ECDH. Згідно з аналізом результатів, очевидно, що запропонований алгоритм перевершує інші алгоритми з точки зору часу обробки та розміру ключа.

У [16] аналізується надійність безпеки двох популярних і практичних методів криптографії з відкритим ключем RSA і ECC. Безпека криптосистеми RSA базується на проблемі цілочисельної факторизації (IFP), а безпека ECC – на проблемі дискретного логарифмування еліптичної кривої (ECDLP). Головна перевага ECC порівняно з RSA полягає в тому, що найвідоміший алгоритм для розв'язання ECDLP займає повний експоненціальний час, тоді як для розв'язання IFP RSA потрібен субекспоненціальний час. Це означає, що в ECC можна використовувати значно менші параметри, ніж у RSA, з еквівалентними рівнями безпеки. Наприклад, для досягнення рівня безпеки 112 біт алгоритму RSA потрібен розмір ключа 2048 біт, тоді як ECC – 224-255 біт.

У роботі [17] запропоновано сумісне використання генетичних алгоритмів і алгоритмів асиметричної криптографії, а робота [18] (як і багато інших подібних) пропонує механізм інтеграції штучного інтелекту та блокчейн-технологій, а в роботі [19] авторами розвинено теорію так званої «нейронної криптографії», у якій алгоритми криптографічної обробки даних і розподілу ключів базуються на алгоритмах синхронізації нейронних мереж. Крім того, на сьогодні існує багато публікацій, пов'язаних із використанням штучного інтелекту для задач криптоаналізу [20, 21] з метою підбору найбільш ефективної криптоаналітичної атаки на основі можливостей зловмисника і характеристик перехоплених даних (фрагментів ключа, шифротексту тощо).

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розглянемо більш детально кожен із досліджуваних алгоритмів шифрування (які, як показав проведений аналіз, використовуються для безпечного передавання даних з борту БПЛА на наземні системи), зокрема їх структури і базові процедури перетворень:

Симетричний блоковий криптоалгоритм AES

Advanced Encryption Standard – симетричний алгоритм блокового шифрування (розмір блока 128 біт, ключ 128 / 192 / 256 біт) [22]. Через фіксований розмір блоку AES оперує із масивом 4×4 байт, що називається станом (версії алгоритму із більшим розміром блоку мають додаткові колонки). Для ключа у 128 біт алгоритм має 10 раундів, у яких послідовно виконуються операції: *SubBytes(), ShiftRows(), MixColumns()* (у десятому раунді ця операція пропускається) та *AddRoundKey().* Структура алгоритму AES показана на Рис. 1.

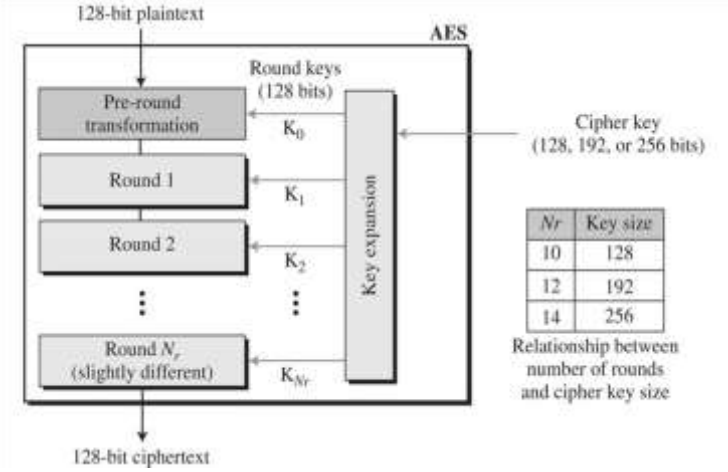

Рис. 1. Структура алгоритму AES

Відповідно до оцінок розробників, шифр стійкий проти таких видів криптоаналітичних атак: диференціального криптоаналізу; лінійного криптоаналізу; криптоаналізу на основі зв'язаних ключів (слабких ключів в алгоритмі немає). Єдиний працюючий спосіб злому шифру AES – це атаки за побічними каналами. Такі атаки не пов'язані з математичними особливостями AES, а використовують певні особливості реалізації систем, що використовують шифр, з метою розкрити частково або повністю секретних даних, у тому числі ключ.

Алгоритм має високу швидкість шифрування. Програмна реалізація на машині з частотою 2 ГГц дозволяє шифрувати дані зі швидкістю 700 Мбіт/с. Такої швидкості достатньо для шифрування відео в форматі MPEG-2 в реальному часі. Апаратні реалізації працюють ще швидше. Останнім часом з'явилася нова версія AEC-NI (New Instructions), яка дозволяє оптимізувати роботу алгоритму (знизити завантаження процесора на 50%). Ця версія може використовуватися і спільно з SSL (протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером). Компанія Intel розробила мікросхему, що реалізує цей алгоритм (серія X5600).

Симетричний блоковий криптоалгоритм DES

Стандарт шифрування даних DES (Data Encryption Standard) – блоковий шифр із симетричними ключами, розроблений Національним Інститутом Стандартів та Технології США (NIST – National Institute of Standards and Technology) [23]. Для шифрування DES (Рис. 2) приймає 64-бітовий відкритий текст і перетворює його в 64-бітовий зашифрований текст і навпаки, отримавши 64 біти зашифрованого тексту – він видає 64 біти розшифрованого. У обох випадках для зашифрування та розшифрування застосовується той самий 56-бітовий ключ.

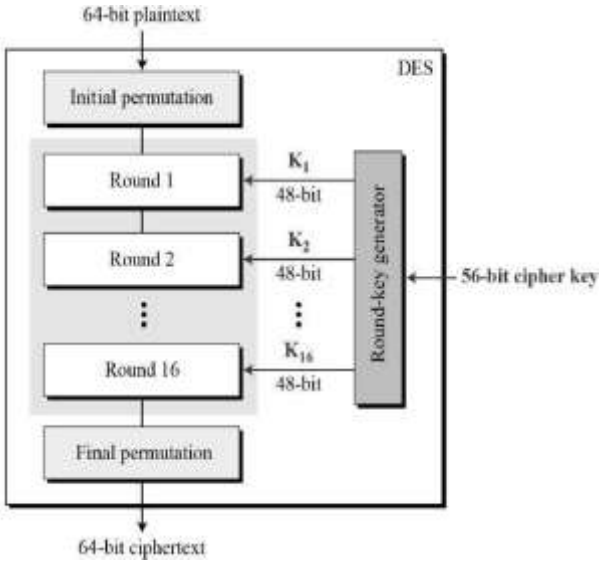

Рис. 2. Структура алгоритму DES

Процес шифрування складається з двох перестановок, які називають початковою і фінальною (кінцевою), та 16 раундів Фейстеля. Кожен раунд використовує різні згенеровані 48-бітові ключі. На вхід кожної з них надходить 64 біти, які потім переставляються відповідно до заданих таблиць (box). Ці перестановки є взаємно оберненими. Інакше кажучи, 58-й біт на вході початкової перестановки перетворюється на 1-шу позицію на виході з неї, а фінальна – 1-й вхідний біт переведе в 58 позицію на

виході. Функція DES за допомогою 48-бітового ключа зашифровує 32 найправіших біт, щоб отримати на виході 32-бітовий результат. Ця функція містить 4 складові: операція XOR, P-box розширення, групу S-box і прямий box. DES створює 16 раундових ключів k_i по 48 бітів із ключа k шифру на 56 бітів. Однак, щоб задати ключ шифру треба серед 56 біт ключа додатково вписати 8 біт в позиції 8,16,...,64 для перевірки парності таким чином, щоб кожен байт містив непарне число одиниць. За допомогою цієї операції виявляють помилки під час обміну та зберігання ключів.

Найбільша проблема DES – розмір ключа (56 бітів). Щоб здійснити атаку грубої сили потрібно перевірити 2^{56} ключів. Якщо перевіряти 10^6 ключів / сек. потрібно, приблизно 2 000 років, щоб виконати атаку грубої сили на DES на одному процесорі. Якщо зробити комп'ютер з мільйоном крипточипів, то така кількість ключів перевіриться за 20 годин. Зазначений криптоалгоритм був зламаний у 1998 році, атака була реалізована протягом 56 годин компанією Electronic Frontier Foundation, використовуючи спеціальний комп'ютер DES Cracker.

Симетричний блоковий криптоалгоритм 3DES

Triple DES (3DES) – симетричний блоковий шифр, створений на основі алгоритму DES з метою усунення головного недоліку останнього – малої довжини ключа (56 біт), який був зламаний методом повного перебору. Швидкість роботи 3DES в 3 рази нижче, ніж у DES, але криптостійкість набагато вища – час, необхідний для криптоаналізу 3DES, може бути в 10^9 разів більше, ніж час, необхідний для розкриття DES [24]. Структура роботи алгоритму показана на Рис. 3.

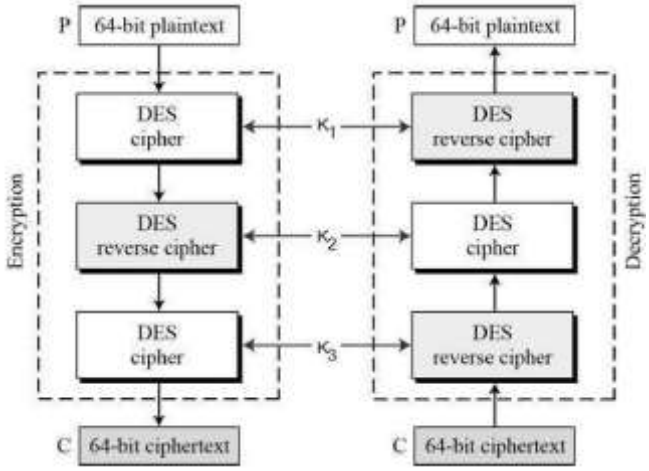

Рис. 3. Структура алгоритму 3DES

Процес шифрування виглядає таким чином: 1) Виконується шифрування блоку відкритого тексту за допомогою DES (одного) з ключем K_1. 2) Проводиться розшифрування вхідних даних кроку 1 за допомогою DES з ключем K_2. 3) Проводиться шифрування вхідних даних кроку 2 за допомогою DES з ключем K_3. Результатом кроку 3 є зашифрований текст.

Розшифрування традиційно є зворотнім процесом – користувач спочатку розшифровує за допомогою K3, потім шифрує за допомогою K2 і, нарешті, розшифровує за допомогою K1. Завдяки такій конструкції Triple DES як процесу зашифрування-розшифрування-зашифрування можна використовувати реалізацію 3TDES (апаратне забезпечення) для одного DES, встановивши K_1, K_2 і K_3 однаковими значеннями. Це

забезпечує зворотну сумісність з DES. Другий варіант Triple DES (2TDES) ідентичний 3TDES за винятком того, що K_3 замінено на K_1. Іншими словами, користувач шифрує блоки відкритого тексту за допомогою ключа K_1, потім розшифровує за допомогою ключа K_2 і, нарешті, знову шифрує за допомогою K_1. Тому 2TDES має довжину ключа 112 біт.

Системи з потрійним DES є значно безпечнішими, ніж одинарний DES, але вони значно повільніші, ніж шифрування з використанням одного алгоритму DES.

Асиметричний криптоалгоритм RSA

RSA – це криптографічний алгоритм з відкритим ключем, що грунтується на обчислювальній складності задачі факторизації великих цілих чисел [25].

Перевагами алгоритму є те, що:

- відсутня проблема розподілу секретних ключів (як і в усіх асиметричних криптоалгоритмах);
- алгоритм дозволяє кільком (багатьом) користувачам обмінюватися інформацією незахищеними каналами зв'язку;
- користувач сам може змінювати як відкритий, так і закритий ключ на власний розсуд, потім він має поширити відкритий ключ у мережі (це дає можливість збільшити криптостійкість).

Недоліками цього алгоритму є:

- невисока швидкість роботи;
- значне навантаження на апаратну платформу.

Структура роботи алгоритму RSA показана на Рис. 4:

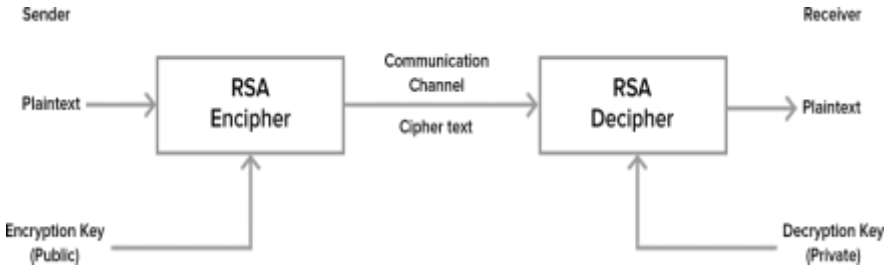

Рис. 4. Структура алгоритму RSA

Безпека алгоритму RSA ґрунтується на трудомісткості розкладання на множники (факторизації) великих чисел. Відкритий та закритий ключ є функціями двох великих простих чисел, розрядністю 100-200 десяткових цифр (або навіть більше). Передбачається, що відновлення відкритого тексту за шифртекстом і відкритим ключем рівносильне розкладанню числа на два великі прості множники.

Симетричний потоковий криптоалгоритм RC4

RC4 (від англ. Rivest Cipher 4 або Ron's Code) – потоковий шифр, що широко застосовується у різних системах захисту інформації в комп'ютерних мережах (наприклад, в протоколах SSL і TLS, алгоритмах безпеки бездротових мереж WEP та WPA) [26]. Алгоритм RC4 (Рис. 5), як і будь-який потоковий шифр, будується на основі генератора псевдовипадкових бітів. На вхід генератора записується ключ, а на виході читаються псевдовипадкові біти. Довжина ключа може становити від 40 до 2048 біт.

ISSN 2663 - 4023

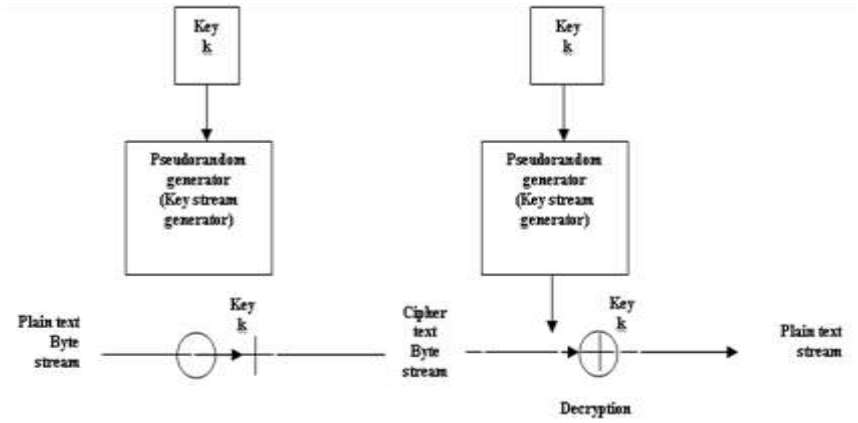

Рис. 5. Структура алгоритму RC4

Генеровані біти мають рівномірний розподіл. Основні переваги – висока швидкість роботи та змінний розмір ключа. Недоліки – вразливість, якщо: використовуються не випадкові чи пов'язані ключі, а також використання одного ключового потіку двічі.

Ядро алгоритму складається з генератора псевдовипадкових бітів (гами), який видає потік бітів ключа (ключовий потік, гаму, послідовність псевдовипадкових бітів). Алгоритм зашифрування складається з наступних кроків: спочатку функція генерує послідовність бітів (k_i); послідовність бітів за допомогою операції «підсумовування за модулем два» (XOR) поєднується з відкритим текстом (m_i) і в результаті виходить шифрограма (c_i): $c_i = m_i + k_i$. Алгоритм розшифрування: повторно створюється (регенерується) потік бітів ключа (ключовий потік) (k_i); потік бітів ключа складається із шифрограмою (c_i) операцією XOR. Завдяки властивостям операції XOR, на виході отримуємо вихідний (незашифрований) текст $m_i = c_i + k_i = (m_i + k_i) + k_i$.

RC4 – фактично клас алгоритмів, що визначаються розміром блоку (S-box). Параметр n є розміром слова алгоритму і визначає довжину S-box. Зазвичай $n = 8$, але з метою аналізу можна його зменшити (відповідно, для підвищення безпеки необхідно збільшити). У алгоритмі відсутні протиріччя збільшення розміру S-box.

Симетричний блоковий криптоалгоритм CAST 128

CAST-128 (або CAST5) – алгоритм блокового шифрування, що використовується кількома прикладними програмами, включаючи деякі версії PGP і GNU Privacy Guard. Він був схвалений в Управлінні безпеки зв'язку Канади для використання урядом. Він шифрує інформацію блоками по 64 біт, використовуючи кілька фіксованих розмірів ключа: 40-128 біт (з кроком 8 біт); підтримує 40-, 64-, 80- і 128-бітні ключі [27]. Цей алгоритм є мережею Фейстеля, в якій виконується 12 або 16 раундів перетворення в залежності від розміру ключа (Рис. 6):

- 12 раундів (при ключах розміром до 80 біт включно);
- 16 раундів (якщо розмір перевищує 80 біт).

№ 1 (17), 2022
ISSN 2663 - 4023

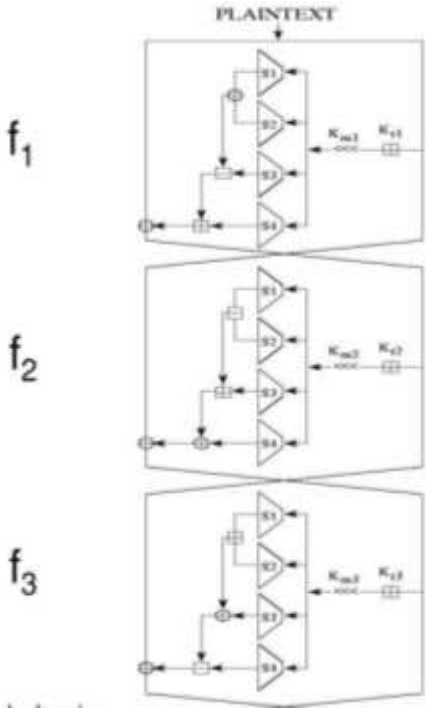

Рис. 6. Структура алгоритму CAST-128

Процедура розширення ключа передбачає формування 32-х 32-розрядних підключів, 16 з яких встановлюються маскувальними підключами Km_i, а інші 16 підключами зсуву Kr_t (в яких використовуються тільки по 5 молодших біт). Перед виконанням розширення ключа виконується доповнення ключа, меншого за 128 біт, нулевими бітами до досягнення 128-бітного розміру. Процедура розширення ключа є досить складною на відміну від алгоритму CAST-256 – CAST-128 має 8 (а не 4) таблиць заміни, причому 4 з них використовуються тільки в процедурі розширення ключа. Це сильно збільшує ресурсоємність алгоритму в частині вимог до енергонезалежної пам'яті. Розшифровування виконується аналогічно процедурі зашифрування, але зі зворотним порядком використання раундових ключів.

На сьогодні не відомо яких-небудь методів зламування алгоритму CAST-128 більш швидких, ніж прямий перебір варіантів ключа. Також, було проведено успішну атаку диференціального криптоаналізу (однак ця атака була спрямована на модифіковану версію алгоритму).

Функція хешування SHA256

Алгоритм SHA-256 (Secure Hash Algorithm 256-bit) – це безпечний алгоритм хешування, розмір вихідних даних якого становить 256 біт. Вихідне повідомлення після доповнення розбивається на блоки, кожен блок – на 16 слів. Алгоритм пропускає кожен блок повідомлення через цикл із 64 або 80 ітерацій (Рис. 7).

ISSN 2663 - 4023

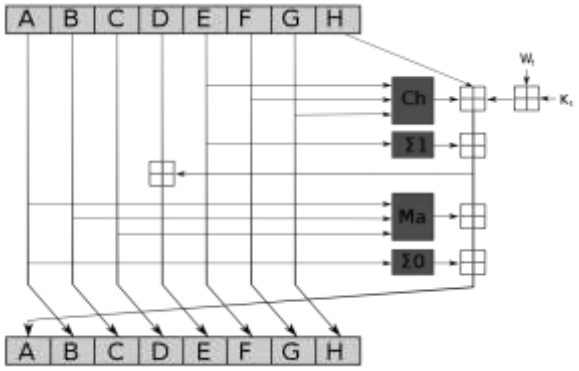

Рис. 7. Схема однієї ітерації алгоритму SHA-256

На кожній ітерації 2 слова перетворюються, функцію перетворення задають інші слова. Результати обробки кожного блоку складаються, сума є значенням хеш-функції. Тим не менш, ініціалізація внутрішнього стану здійснюється результатом обробки попереднього блоку, а тому незалежно обробляти блоки та складати результати не можливо [28].

Симетричний блоковий криптоалгоритм Blowfish

Blowfish – криптографічний алгоритм, що реалізує блокове симетричне шифрування зі змінною довжиною ключа (Рис. 8). Виконаний на простих та швидких операціях: XOR, підстановка, додавання. Алгоритм складається з розширення ключа та шифрування даних. На етапі розширення ключа вихідний ключ (довжиною до 448 біт) перетворюється у 18 32-бітових підключів і в 4 32-бітних S-box, що містять 256 елементів. Загальний обсяг отриманих ключів 33 344 біт або 4 168 байт [23].

Етап 1 – підготовчий (формування ключів шифрування за секретним ключем).

Ініціалізація масивів *P* та *S* за допомогою секретного ключа *K:*

1. Ініціалізація P1−P18 фіксованим рядком, що складається з шістнадцяткових цифр.

2. Проводиться операція XOR над P1 з першими 32 бітами ключа K над P2 з другим 32-бітами і так далі. Якщо ключ K коротший, він накладається циклічно.

Шифрування ключів та таблиць замін:

1. Алгоритм шифрування 64-бітного блоку, використовуючи ініціалізовані ключі P1-P18 та таблицю замін S1-S4, шифрує 64 бітну нульову (0x0000000000000000) рядок. Результат записується в P1, P2.

2. P1, P2 шифруються зміненими значеннями ключів та таблиць замін. Результат записується в P3, P4.

3. Шифрування продовжується до зміни всіх ключів P1-P18 та таблиць замін S1-S4.

Етап 2 – шифрування тексту (отриманими ключами та *F(x)*, з попереднім розбиттям на блоки по 64 біти). Якщо неможливо розбити початковий текст на блоки по 64 біта, використовуються різні режими шифрування для побудови повідомлення, що складається з цілого числа блоків. Сумарна пам'ять 4 168 байт.

ISSN 2663 - 4023

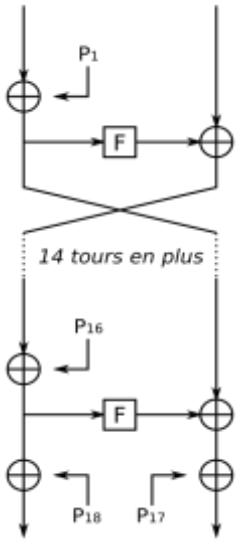

Рис. 8. Структура алгоритму Blowfish

Розшифрування відбувається аналогічно, тільки P1-P18 застосовуються у зворотньому порядку.

Асиметричний криптоалгоритм ECC

Криптографія на основі еліптичних кривих часто обговорюється в контексті криптографічного алгоритму RSA, на відміну від якого ECC базується на тому, як еліптичні криві структуровані алгебраїчно над кінцевими полями. Таким чином, ECC створює ключі, які складніше зламати з математичної точки зору.

ECC підтримує високий рівень як продуктивності, так і безпеки. Це пов'язано з тим, що ECC використовується все ширше, оскільки веб-сайти прагнуть одночасно забезпечити кращу безпеку даних клієнтів в Інтернеті та оптимізацію для мобільних пристроїв [29]. Еліптична крива для цілей ECC – це плоска крива над кінцевим полем, яке складається з точок, що задовольняють рівняння (Рис. 9): $y^2=x^3 + ax + b$. У цьому прикладі будь-яка точка кривої може бути дзеркально відображена на осі *x,* і крива залишиться незмінною. Будь-яка невертикальна лінія перетинатиме криву в трьох місцях (або менше).

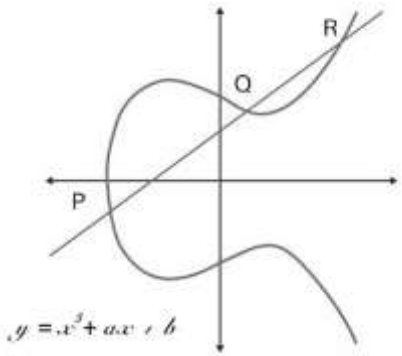

Рис. 9. Схема Elliptic Curve Cryptography

ECC містить менші зашифровані тексти, ключі та підписи, а також швидшу генерацію ключів і підписів. Його швидкість шифрування є помірно високою. ECC забезпечує нижчу затримку, ніж інверсна, завдяки обчисленню підписів у два етапи. ECC має надійні протоколи для автентифікованого обміну ключами (Рис.10).

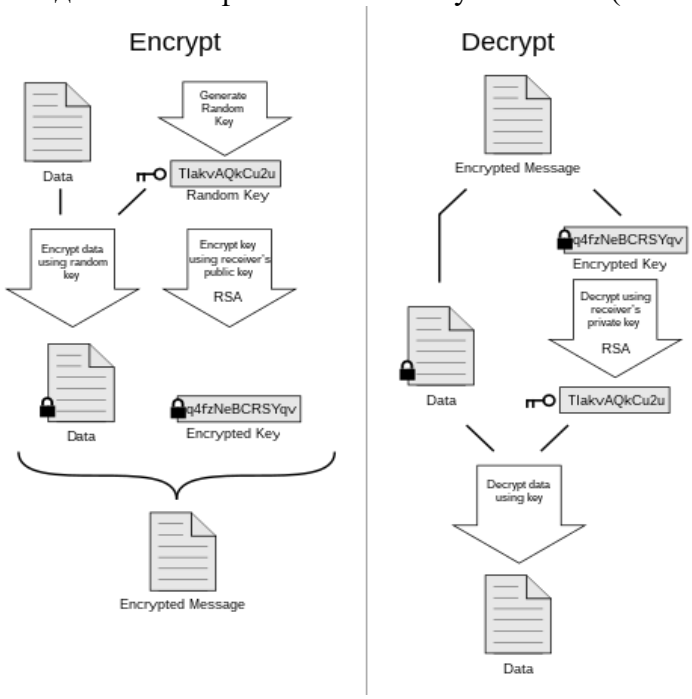

Рис. 10. Схема шифрування з використанням ECC

Є кілька потенційних уразливостей для криптографії на основі еліптичної кривої, включаючи атаки за побічними каналами і атаки зі зміною безпеки. Обидва типи мають на меті зробити недійсним захист ECC для приватних ключів.

Симетричний потоковий криптоалгоритм OTP

OTP – це алгоритм, який використовує структуру симетричного потокового шифру [8]. У передавачі відкритий текст перетворюється на біти та генерується випадковий бітовий ключ. Довжина ключа має бути принаймні такою ж, як і повідомлення. Для отримання зашифрованого тексту між відкритим текстом і ключем виконується операція XOR. Щоб отримати оригінальне повідомлення, одержувач виконує операцію XOR між зашифрованим текстом і ключем (Рис.11).

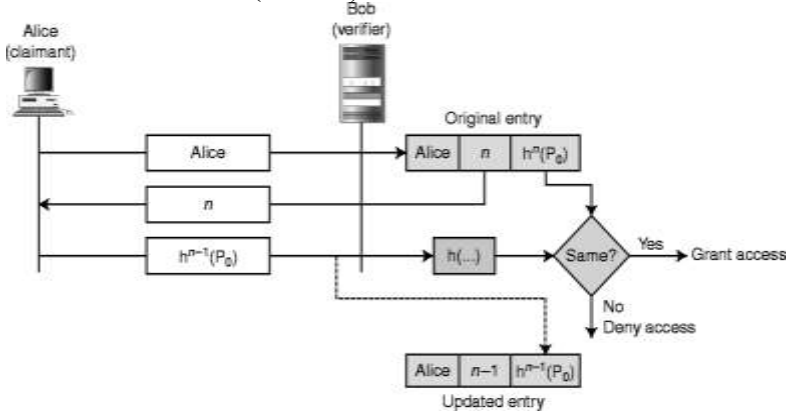

Рис. 11. Структура алгоритму OTP

OTP вимагає, щоб ключ безпечно доставлявся користувачам і було досягнуто справжньої випадковості генерації ключів. Ключ не можна використовувати більше одного разу, а розмір ключа повинен бути невідомий зловмиснику. Якщо ці вимоги виконуються, OTP неможливо зламати. Однак, якщо обмін ключами скомпрометовано, техніка шифрування дає збій, оскільки легко скасувати одну операцію XOR. Крім того, успішне розшифрування ніколи не слід використовувати як форму автентичності, оскільки мережеві атаки (типу атаки «людина посередині» та атаки з відтворенням), можуть розкрити зловмиснику частини ключа. Якщо відомий розмір ключа, ключ можна зламати грубою силою; однак, залежно від розміру ключа, це, швидше за все, буде неможливо. Крім того, якщо ключ використовується кілька разів, зловмисники можуть виконати атаку потокового шифру, щоб розкрити вміст ключа. Через просту складність OTP він одночасно надійний і легкий.

Симетричний потоковий криптоалгоритм CHACHA20

Chacha20 – потоковий шифр, що складається з 256-бітного ключа (*key*), 32-бітного лічильника (*counter*), 96-бітного одноразового номеру (*nonce*) і звичайного тексту (Рис. 12). Його початковий стан – це матриця 4×4 із 32-розрядних слів. Перший рядок – це постійний рядок «*expand 32-byte k*», який ділиться на 4 32-бітові слова. Другий і третій – заповнюються 256-бітним ключем. Перше слово в останньому рядку є 32-бітним лічильником, а інші – 96-бітним *nonce*. Він генерує 512-бітний потік ключів у кожній ітерації для шифрування 512-бітного блоку звичайного тексту. Процес зашифрування та розшифрування однакові, якщо ввести той самий початковий *key, counter* і *nonce* [30].

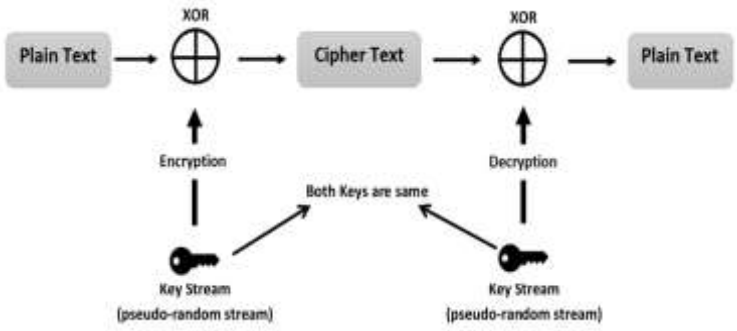

Рис. 12. Структура алгоритму CHACHA20

Chacha20 складається з 2 частин: ініціалізація та шифрування. Початковий стан генерується вхідним 256-бітним ключем, 32-бітним лічильником і 96-бітним одноразовим номером. Під час шифрування генерується новий 512-бітний ключ, який використовується для виконання XOR із 512-бітним простим текстом, а потім виводить блок шифру під час кожної ітерації.

Симетричний потоковий криптоалгоритм CHACHA20-POLY 1305

ChaCha20-Poly1305 – це алгоритм автентифікованого шифрування з додатковими даними (AEAD), який поєднує потоковий шифр ChaCha20 із кодом автентифікації повідомлення Poly1305. Його використання в протоколах IETF стандартизовано в RFC 8439. Він має швидку програмну продуктивність і без апаратного прискорення зазвичай швидше ніж AES-GCM.

Алгоритм ChaCha20-Poly1305 приймає як вхідні дані 256-бітний ключ і 96-бітний одноразовий номер для шифрування відкритого тексту з розширенням зашифрованого тексту 128-біт (Рис.13).

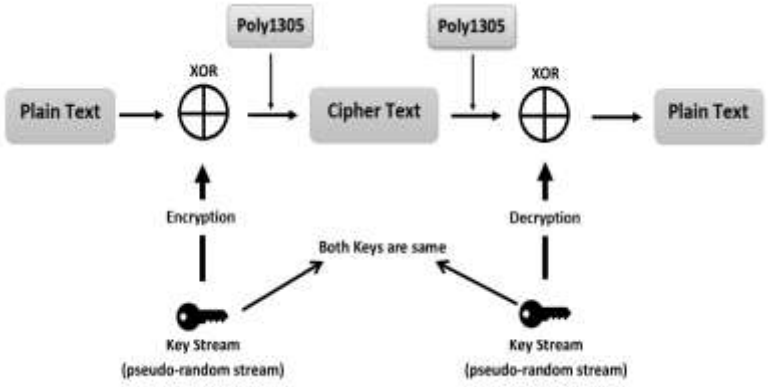

Рис. 13. Структура алгоритму CHACHA20-Poly1305

У конструкції ChaCha20-Poly1305, ChaCha20 використовується в режимі лічильника для отримання потоку ключів, який об'єднується операцією XOR з відкритим текстом. Потім зашифрований текст і пов'язані дані перевіряються за допомогою варіанту Poly1305, який спочатку кодує два рядки в один [31].

ПОРІВНЯННЯ КРИПТОАЛГОРИТМІВ ЗА ВИЗНАЧЕНИМИ КРИТЕРІЯМИ

З огляду на велику кількість алгоритмів, які можуть використовуватись в БПЛА, необхідно провести порівняльний аналізу за певними критеріями (критерій 1 будемо позначати *К1*, критерій 2 – *К2* і т.д.). Такими критеріями є:

1) 128-бітний розмір блоку даних, що шифруються *(К1);*

2) не менше трьох підтримуваних алгоритмом розмірів ключів шифрування: 128, 192 та 256 біт *(К2);*

3) алгоритм має бути стійким проти криптоаналітичних атак, відомих на цей час *(К3);*

4) структура алгоритму має бути зрозумілою, простою та обґрунтованою, що гарантувало б відсутність запроваджених розробниками «закладок» (тобто в даному випадку, особливостей архітектури алгоритму, якими теоретично могли б скористатися його автори у зловмисних цілях) *(К4);*

5) повинні бути відсутніми слабкі та еквівалентні ключі (тобто ключі, що є різними, але призводять до одного і того ж результату шифрування) *(К5);*

6) швидкість шифрування даних має бути високою на всіх потенційних апаратних платформах – від 8-бітових до 64-бітових *(К6);*

7) алгоритм повинен пред'являти мінімальні вимоги до оперативної та енергонезалежної пам'яті *(К7);*

8) не повинно бути обмежень для використання алгоритму; зокрема, алгоритм не повинен обмежувати своє використання в різних стандартних режимах роботи, генераторів псевдовипадкових послідовностей і т.д *(К8).*

Багатокритеріальний аналіз алгоритмів наведено у Табл. 1, де N/A – NOT APPLICABLE (критерій не використовується для асиметричних криптоалгоритмів).

Таблиця 1

Багатокритеріальний аналіз криптоалгоритмів

Критерії / Алгоритми	К1	К2	К3	К4	К5	К6	К7	К8
Симетричне, блокове шифрування								
AES	+	+	+	+	+	+	+	+
DES	-	-	-	+/-	-	-	+	+
3DES	-	-	+/-	+/-	+	-	+	+
CAST 128	-	-	+	+	+	+	+	+
Blowfish	-	+	-	+	-	-	+	+
Симетричне, потокове шифрування								
RC4	+	+	-	+	-	+	+	+
OTP	+	-	+	+	+	+	+	+
ChaCha 20	+	-	+	+	+	+	+	+
ChaCha20-POLY1305	+	-	+	+	+	+	+	+
Асиметричне шифрування								
RSA	N/A	+	+	+	+	-	+/-	+
ECC	N/A	+	+	+	+	-	+/-	+
Хеш-функція								
SHA 256	+	-	+	+	+	+	+	+

Проведений аналіз (Табл. 1) демонструє, що кожен з розглянутих алгоритмів має переваги та недоліки, не існує універсального криптоалгоритму, який здатен вирішити усі проблеми конфіденційності в БПЛА. З огляду на це, необхідно створити базу даних (датасет) криптографічних алгоритмів, які могли б розв'язувати різного роду задачі при застосуванні БПЛА.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У цій роботі проведено огляд та порівняльний аналіз сучасних криптоалгоритмів, які використовуються для забезпечення конфіденційності даних під час їх передавання радіоканалом з БПЛА до наземних об'єктів. Для порівняння криптоалгоритмів використовувалась система критеріїв (подібно конкурсам AES, NESSIE тощо), пов'язаних з: розмірами блоку і ключа; режимами роботи; швидкістю шифрування; вимогами до пам'яті; стійкістю до криптоаналізу.

Аналіз показав, 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, необхідно створити універсальний набір (датасет) криптографічних алгоритмів, які могли б розв'язувати різного роду задачі у різних умовах. Крім того, актуальним і новітнім підходом на сьогодні є застосування методів штучного інтелекту для вибору оптимального алгоритму з датасету [17-21]. Саме цим дослідженням і буде присвячена подальша робота авторів у рамках виконуваного наукового проєкту.

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. *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC),* 1445-1450. DOI: 10.1109/IMCEC51613.2021.9482373.

2 Thompson, R. B., Thulasiraman, P. (2016). Confidential and authenticated communications in a large fixed-wing UAV swarm. *IEEE International Symposium on Network Computing and Applications (NCA),* 375-382.

3 Koukou, Y.M., Othman, S.H., Siraj, M. M., Nkiama, H. (2016) .Comparative Study of AES, Blowfish, CAST-128 and DES Encryption Algorithm. *IOSR Journal of Engineering (IOSRJEN), 6*(6), 1-7.

4 Singhal, N.,.Raina, J.P.S. (2011). Comparative Analysis of AES and RC4 Algorithms for Better Utilization. International *Journal of Computer Trends and Technology (IJCTT), 1*(3), 259-263.

5 Kumar, B. J. S., Raj, V. K. R., Nair, A. (2017). Comparative study on AES and RSA algorithm for medical images. I*nternational Conference on Communication and Signal Processing (ICCSP),* 501-504.

6 Nadeem, A., Javed, M. Y. (2005). A Performance Comparison of Data Encryption Algorithms. *International Conference on Information and Communication Technologies,* 84-89.

7 Mandal, A. K., Parakash, C., Tiwari, A. (2012). A. Performance evaluation of cryptographic algorithms: DES and AES. I*EEE Students' Conference on Electrical, Electronics and Computer Science,* 1-5.

8 Khoei, T., Ghribi, E., Ranganathan, P., Kaabouch, N. (2021). A performance comparison of encryption/decryption algorithms for UAV swarm communications. *Academic Press, 1,* 1-5.

9 Usman, M., Amin, R., Aldabbas, H., Alouffi, B. (2022). Lightweight Challenge-Response Authentication in SDN-Based UAVs Using Elliptic Curve Cryptography. *Electronics 2022, 11,* 1026. https://doi.org/10.3390/ electronics11071026.

10 Muslum Ozgur Ozmen, Rouzbeh Behnia, Attila A Yavuz. (2019). IoD-crypt: A lightweight cryptographic framework for Internet of drones. arXiv, 1.

11 Syafaat, F., Farhan, A. (2019). Implementation of AES-128 Cryptography on Unmanned Aerial Vehicle and Ground Control System. *Teknik Informatika – Universitas Komputer Indonesia,* 10-19.

12 Ronaldo, F., Pramadihanto, D., Sudarsono, A. (2020). Secure Communication System of Drone Service using Hybrid Cryptography over 4G/LTE Network. *2020 International Electronics Symposium (IES),* 116-122. DOI: 10.1109/IES50839.2020.9231951.

13 Bafandehkar, Mohsen et al. (2013). Comparison of ECC and RSA Algorithm in Resource Constrained Devices. 2*013 International Conference on IT Convergence and Security (ICITCS),* 1-3.

14 Bansal, Malti et al. (2021). Comparison of ECC and RSA Algorithm with DNA Encoding for IoT Security. *2021 6th International Conference on Inventive Computation Technologies (ICICT),* 1340-1343.

15 Nagesh, O. S., Vankamamidi, S. N. (2020). Comparative Analysis of MOD-ECDH Algorithm and Various Algorithms. *International Journal of Industrial Engineering & Production Research, 31*(2), 301-308.

16 Mahto, Dindayal et al. (2016). Security Analysis of Elliptic Curve Cryptography and RSA. *Proceedings of the World Congress on Engineering 2016,* I, 1-4.

17 Jhajharia, S., Mishra, S., Bali, S. (2013). Public key cryptography using neural networks and genetic algorithms. *2013 Sixth International Conference on Contemporary Computing (IC3),* 137-142.

18 Chavali, B., Khatri, S. K, Hossain, S. A. (2020). AI and Blockchain Integration. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO),* 548-552.

19 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.

20 Danziger, M., Amaral, M. A., Henriques. (2014). Improved cryptanalysis combining differential and artificial neural network schemes. *2014 International Telecommunications Symposium (ITS),* 1-5.

21 Xiao, Y., Hao, Q., Yao, D. D. (2019). Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers. *2019 IEEE Conference on Dependable and Secure Computing (DSC),* 1-8.

22 Daemen, J., Rijmen, V. (2003). AES Proposal: Rijndael. National Institute of Standards and Technology 2003, 1.

23 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.

24 Kelsey, J., Schneier, B., Wagner, D. (1996). Key-schedule cryptanalysis of IDEA, GDES, GOST, SAFER, and Triple-DES. Advances in Cryptology, Proceedings Crypto ’96, 237-252.

25 Boneh, D. (1999). Twenty Years of Attacks on the RSA. *Notices of the American Mathematical Society 1999, 46*(2), 203–213.

26 Isobe, Takanori, Ohigashi, Toshihiro. (2013). Security of RC4 Stream Cipher. Hiroshima University, 10.

27 Carlisle, M. A. (1997). Constructing Symmetric Ciphers Using the CAST Design Procedure. Designs, Codes and Cryptography, 283-316.
28 Dobraunig, C., Eichlseder, Maria., Mendel, F. (2016). Analysis of SHA-512/224 and SHA-512/256. IACR Cryptology ePrint Archive, 374.
29 Standards for Efficient Cryptography Group (SECG), SEC 1: Elliptic Curve Cryptography, Version 1.0, September 20, 2000. https://www.secg.org/SEC1-Ver-1.0.pdf.
30 Bernstein, D. J.. (2008). ChaCha, a variant of Salsa20. The State of the Art of Stream Ciphers SASC 2008. https://cr.yp.to/chacha/chacha-20080120.pdf.
31 F. De Santis, Schauer, A., Sigl., G. (2017). ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 692-697. doi: 10.23919/DATE.2017.7927078.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (17), 2022
ISSN 2663 - 4023

Sergiy O. Gnatyuk
DSc, Professor, Vice-Dean of the Faculty of Cybersecurity, Computer and Software Engineering
Chair of NAU Cybersecurity R&D Lab
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0003-4992-0564
s.gnatyuk@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

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

Olena P. Nechyporuk
DSc, Associate Professor, Professor of Computerised Control Systems Academic Department
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0001-8203-7998
olena.nechyporuk@npp.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

ANALYSIS OF METHODS FOR DATA CONFIDENTIALITY ENSURING DURING TRANSMITTING FROM UAV

Abstract. The rapid development of unmanned aerial vehicles (UAVs), as well as the expansion of the list of actions performed by modern UAVs, led to increased requirements for the safety and reliability of data transmission. In the context of warfare, when confidential information is collected, the protection of such information is a top priority. The practical level of conducting aerial reconnaissance during current warfare demonstrates the urgent need to create UAV which capable of performing flight tasks and aerial reconnaissance in the mode of installed radio interference, and also emphasizes the importance of ensuring the data confidentiality about target objects transmitted by an optical channel for the implementation of their processing in automated systems. The paper provides a review and comparative analysis of modern cryptoalgorithms that are used to ensure data confidentiality during their transmission by radio channel from UAV to ground objects. There are the system of criteria (multi criteria analysis) was used to compare following cryptographic algorithms (similar to AES, NESSIE, etc competitions): block and key sizes; modes of operation; encryption speed; memory requirements; resistance (security) to cryptanalysis. The conducted analysis showed that each cryptographic algorithm has advantages and disadvantages. Also, there is no universal cryptographic algorithm that capable to resolve all privacy problems in UAV. According to the limited resources in the process of UAV operation, it is necessary to create a universal set (dataset) of cryptographic algorithms that could solve various problems in different conditions including different aspects of UAV exploitation. It is these studies that will be devoted to the further work of the authors within the framework of the ongoing scientific project.

Keywords: UAV; data confidentiality; cryptography; cryptographic algorithm; encryption; data transmission; dataset.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1 Du, X., Tang, Y., Gou, Y., Huang, Z. (2021). Data Processing and Encryption in UAV Radar. *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC),* 1445-1450. DOI: 10.1109/IMCEC51613.2021.9482373.

2 Thompson, R. B., Thulasiraman, P. (2016). Confidential and authenticated communications in a large fixed-wing UAV swarm. *IEEE International Symposium on Network Computing and Applications (NCA),* 375-382.

3 Koukou, Y.M., Othman, S.H., Siraj, M. M., Nkiama, H. (2016) .Comparative Study of AES, Blowfish, CAST-128 and DES Encryption Algorithm. *IOSR Journal of Engineering (IOSRJEN), 6*(6), 1-7.

4 Singhal, N.,.Raina, J.P.S. (2011). Comparative Analysis of AES and RC4 Algorithms for Better Utilization. International *Journal of Computer Trends and Technology (IJCTT), 1*(3), 259-263.

5 Kumar, B. J. S., Raj, V. K. R., Nair, A. (2017). Comparative study on AES and RSA algorithm for medical images. I*nternational Conference on Communication and Signal Processing (ICCSP),* 501-504.

6 Nadeem, A., Javed, M. Y. (2005). A Performance Comparison of Data Encryption Algorithms. *International Conference on Information and Communication Technologies,* 84-89.

7 Mandal, A. K., Parakash, C., Tiwari, A. (2012). A. Performance evaluation of cryptographic algorithms: DES and AES. I*EEE Students' Conference on Electrical, Electronics and Computer Science,* 1-5.

8 Khoei, T., Ghribi, E., Ranganathan, P., Kaabouch, N. (2021). A performance comparison of encryption/decryption algorithms for UAV swarm communications. *Academic Press, 1*, 1-5.

9 Usman, M., Amin, R., Aldabbas, H., Alouffi, B. (2022). Lightweight Challenge-Response Authentication in SDN-Based UAVs Using Elliptic Curve Cryptography. *Electronics 2022, 11,* 1026. https://doi.org/10.3390/ electronics11071026.

10 Muslum Ozgur Ozmen, Rouzbeh Behnia, Attila A Yavuz. (2019). IoD-crypt: A lightweight cryptographic framework for Internet of drones. arXiv, 1.

11 Syafaat, F., Farhan, A. (2019). Implementation of AES-128 Cryptography on Unmanned Aerial Vehicle and Ground Control System. *Teknik Informatika – Universitas Komputer Indonesia,* 10-19.

12 Ronaldo, F., Pramadihanto, D., Sudarsono, A. (2020). Secure Communication System of Drone Service using Hybrid Cryptography over 4G/LTE Network. *2020 International Electronics Symposium (IES),* 116-122. DOI: 10.1109/IES50839.2020.9231951.

13 Bafandehkar, Mohsen et al. (2013). Comparison of ECC and RSA Algorithm in Resource Constrained Devices. 2*013 International Conference on IT Convergence and Security (ICITCS),* 1-3.

14 Bansal, Malti et al. (2021). Comparison of ECC and RSA Algorithm with DNA Encoding for IoT Security. *2021 6th International Conference on Inventive Computation Technologies (ICICT),* 1340-1343.

15 Nagesh, O. S., Vankamamidi, S. N. (2020). Comparative Analysis of MOD-ECDH Algorithm and Various Algorithms. *International Journal of Industrial Engineering & Production Research, 31*(2), 301-308.

16 Mahto, Dindayal et al. (2016). Security Analysis of Elliptic Curve Cryptography and RSA. *Proceedings of the World Congress on Engineering 2016,* I, 1-4.

17 Jhajharia, S., Mishra, S., Bali, S. (2013). Public key cryptography using neural networks and genetic algorithms. *2013 Sixth International Conference on Contemporary Computing (IC3),* 137-142.

18 Chavali, B., Khatri, S. K, Hossain, S. A. (2020). AI and Blockchain Integration. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO),* 548-552.

19 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.

20 Danziger, M., Amaral, M. A., Henriques. (2014). Improved cryptanalysis combining differential and artificial neural network schemes. *2014 International Telecommunications Symposium (ITS),* 1-5.

21 Xiao, Y., Hao, Q., Yao, D. D. (2019). Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers. *2019 IEEE Conference on Dependable and Secure Computing (DSC),* 1-8.

22 Daemen, J., Rijmen, V. (2003). AES Proposal: Rijndael. National Institute of Standards and Technology 2003, 1.

23 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.

24 Kelsey, J., Schneier, B., Wagner, D. (1996). Key-schedule cryptanalysis of IDEA, GDES, GOST, SAFER, and Triple-DES. Advances in Cryptology, Proceedings Crypto ’96, 237-252.

25 Boneh, D. (1999). Twenty Years of Attacks on the RSA. *Notices of the American Mathematical Society 1999, 46*(2), 203–213.

26 Isobe, Takanori, Ohigashi, Toshihiro. (2013). Security of RC4 Stream Cipher. Hiroshima University, 10.

27 Carlisle, M. A. (1997). Constructing Symmetric Ciphers Using the CAST Design Procedure. Designs, Codes and Cryptography, 283-316.
28 Dobraunig, C., Eichlseder, Maria., Mendel, F. (2016). Analysis of SHA-512/224 and SHA-512/256. IACR Cryptology ePrint Archive, 374.
29 Standards for Efficient Cryptography Group (SECG), SEC 1: Elliptic Curve Cryptography, Version 1.0, September 20, 2000. https://www.secg.org/SEC1-Ver-1.0.pdf.
30 Bernstein, D. J.. (2008). ChaCha, a variant of Salsa20. The State of the Art of Stream Ciphers SASC 2008. https://cr.yp.to/chacha/chacha-20080120.pdf.
31 F. De Santis, Schauer, A., Sigl., G. (2017). ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 692-697. doi: 10.23919/DATE.2017.7927078.