Сучасні методи шифрування за допомогою алгоритму RC4

Петрук М. І.[a], Ставицький О. В.[a1]
[a] *Київський інститут бізнесу та технологій, Україна*

Анотація

У наш час шифрування даних та конфіденційність є однією з найважливіших проблем. У даній статті ми розглянемо сучасні методи шифрування та розкриємо всі аспекти, що пов'язані з актуальністю використання та вразливістю алгоритму RC4. Методика шифрування – це процес перетворення даних відкритого тексту в зашифрований текст з метою приховування його значення і таким чином, запобігання несанкціонованому одержувачу отримати вихідні дані. Отже, шифрування в основному використовується для забезпечення секретності. Компанії зазвичай шифрують свої дані перед передачею, щоб переконатися, що дані захищені під час транзиту. Зашифровані дані надсилаються через загальнодоступну мережу і розшифровуються одержувачем. До появи Інтернету криптографія займалася тільки шифруванням повідомлень — перетворенням повідомлень з зрозумілих в незрозумілі, роблячи їх нечитабельним для людини, яка перехопила повідомлення, і зворотним перетворенням одержувачем при збереженні суті повідомлення. Останніми роками криптографія розпочала поширюватися і окрім таємної передачі повідомлень стала включати в себе методи перевірки цілісності повідомлень, технології безпечного спілкування, аутентифікацію відправника та одержувача (за допомогою ключів, цифрових підписів, тощо), і багато іншого. У даній статті буде розглянуто алгоритм RC4, також відомий як ARC4 або ARCFOUR – потоковий шифр, який широко застосовується в різних системах захисту інформації в комп'ютерних мережах (наприклад, в протоколах SSL і TLS, алгоритмах забезпечення безпеки бездротових мереж WEP і WPA).

Ключові слова: *конфіденційність, кібербезпека, криптографія.*

Modern encryption methods using the RC4 algorithm

Petruk M. I.[a], Stavytskyi O. V.[a1]
[a] *Kiev Institute of Business and Technology, Ukraine*

Abstract

Nowadays, data encryption and privacy are one of the most important issues. In this article, we will look at modern encryption methods and reveal all aspects related to the relevance of use and vulnerability of the RC4 algorithm. Encryption techniques are the process of converting plaintext data into encrypted text in order to hide its value and thus prevent an unauthorized recipient from obtaining the original data. Therefore, encryption is mainly used to ensure confidentiality. Companies typically encrypt their data before transferring to make sure the data is secure during transit. Encrypted data is sent over a public network and decrypted by the recipient. Before the advent of the Internet, cryptography dealt only with encrypting messages – converting messages from intelligible to incomprehensible, making them unreadable to the person who intercepted the message, and reverse conversion by the recipient while preserving the essence of the message. In recent years, cryptography has begun to expand and, in addition to secret messaging, has begun to include methods for verifying the integrity of messages, secure communication technology, sender and recipient authentication (using keys, digital signatures, etc.), and more. This article will discuss the RC4 algorithm, also known as ARC4 or ARCFOUR – streaming cipher, which is widely used in various information security systems in computer networks (eg, SSL and TLS, wireless security algorithms WEP and WPA).

Keywords: *confidentiality, cybersecurity, cryptography.*

[1] Corresponding author.
E-mail address: stavitsky@kibit.edu.ua

Отримано / Recieved 12.10.2020
Отримано виправлений варіант / Received in revised form 21.10.2020
Прийнято до друку / Accepted 27.10.2020

https://doi.org/10.37203/kibit.2020.45.10

Вступ

Основною метою при розробці будь-якого з алгоритмів шифрування є повне забезпечення конфіденційності інформації та збереження даних користувача від атак сторонніх осіб. Але для забезпечення необхідного рівня захисту існують певні фактори які впливають на вартість таких послуг, та на ціну впровадження існуючих систем. Розроблений алгоритм шифрування буде не ефективним якщо він буде досить простим та мати невелику продуктивність. Традиційною метою шифрування завжди був захист інформації та забезпечення конфіденційності користувачів.

Криптографія – це інструмент, який може використовуватися для збереження конфіденційності інформації, забезпечення її цілісності та достовірності.

Криптографія є ключовим елементом встановлення довіри та надання послуг у цифровому світі. В даний час криптографія реалізується за допомогою математичних операцій і представляється не такою легкою для розуміння широкого кола користувачів (Halunen & Latvala, 2021).

Для запобігання промислового шпіонажу в конкурентних умовах роботи організацій та підприємств актуальною є проблема розроблення системи шифрування конфіденційної інформації на основі вибору надійного алгоритму шифрування даних. Для вирішення цієї проблеми виділяють два послідовні етапи (White, 1990):

- доведення надійності алгоритму на основі математичних методів;
- перевірка алгоритму на практиці.

Рання криптографія займалася виключно перетворенням повідомлень у нечитабельні групи слів та цифр, щоб захистити вміст повідомлення під час перенесення повідомлення з одного місця в інше. У сучасну епоху криптографія зросла від базової конфіденційності повідомлень, включаючи, зокрема, деякі етапи перевірки цілісності повідомлень, автентифікацію ідентифікації відправника, одержувача та цифрові підписи.

Найбільш ранні форми криптографії були знайдені в регіонах захоплених Єгіптом, Грецією та Римом. Ще в 1900 рр. до н.е. Єгипетські книжки використовували нестандартні ієрогліфи, ймовірно щоб приховати значення зашифрованої інформації (Whitman & Mattord, 2005).

Одним із відомих шифрів є шифр Цезаря. Шифр зміщення Цезаря – приклад моноалфавітного шифру в якому кожна буква відкритого тексту замінюється на ту, що знаходиться від неї в алфавіті на якійсь сталій кількості позицій. Зашифровані тексти, отримані з подібних шифрів, завжди мають однакову інформацію про вхідний текст, через це вони є вразливими до зламу. Після винаходу частотного аналізу в IX столітті усі подібні шифри стали легко зламувалися, через це класичні шифри залишаються популярними у виді головоломок.

У часи другої світової війни декодування зашифрованих повідомлень стала одним із перших використанням криптографії. Союзники стикнулися з невідомим кодом та машиною кодування “Енігма”, яку використовували для шифрування даних та для заміни вхідного повідомлення на зашифроване. Енігма складалася з набору механічних та електричних частин та роторів. Шифрування полягало в пошаговому зміщенню роторів за підміні символів, але мала вагомий недолік, при отриманні великого об’єму зашифрованого повідомлення, надійність криптосистеми зменшувалась.

Менш відомою історією використання криптографії в другій світовій війні є рішення американців для шифрування використовувати мову індіанців Навахо. Через маловідомість та складність мови разом із звичайним шифруванням американці використовували шифрувальників Навахо, які передавали інформацію своєю рідною мовою. Шифрувальники Навахо приймали участь майже у всіх основних операціях в Тихому океані 1942-1945 років.

Відхилимося від теми криптографії та розглянемо криптовалюти. Криптовалюта — це цифровий актив призначений для роботи як засіб обміну та заснований на технології блокчейну. Блокчейн здається складним, і це, безумовно, так і є, але його основна концепція насправді дуже проста. Блокчейн є одним із типів баз даних. Для розуміння що таке блокчейн та криптовалюти спочатку потрібно розглянути основи баз даних.

Бази даних призначені для однієї людини або невеликої групи людей для зберігання та доступу до обмеженого обсягу інформації. Великі бази даних досягають цього, розміщуючи дані на серверах, які створені з потужних комп’ютерів. Ці сервери іноді можуть бути побудовані з використанням сотень або тисяч комп’ютерів, щоб мати обчислювальну потужність та ємність, необхідну багатьом користувачам для одночасного доступу до бази даних.

Однією з ключових відмінностей між типовою базою даних та блокчейном є спосіб структури даних. Блокчейн збирає інформацію разом у групи, також відомі як блоки, які містять набори інформації. Блоки мають певну ємність для зберігання і, заповнюючись, прив’язуються до заповненого раніше блоку, утворюючи ланцюжок даних, відомий як “блокчейн”. Вся нова інформація, що випливає з того, що свіжо додані блоки, компілюється у новостворений блок, який потім також буде доданий до ланцюжка після заповнення. Криптовалюти отримуються за проведення операцій в системі якоїсь даної криптовалюти, а платять користувачам за обчислювальну силу. Незважаючи на назву

“криптовалюти” вона мало пов’язана напряму з самою криптографією і є лише альтернативою платіжних методів.

Бездротовий зв’язок допомагає обмінюватися інформацією від однієї точки до іншої або більше. WEP та WPA — є основними механізмами захисту даних в мережі, що використовується для забезпечення безпеки в середовищі бездротової мережі.

Цей захід безпеки стосується бездротової локальної мережі та є частиною стандарту безпеки IEEE 802.11. У WEP для забезпечення безпеки та цілісності використовується циклічний код резервування (CRC-32), а шифр потоку RC4 використовується для забезпечення конфіденційності. Стандартна специфікація WEP підтримує довжину ключа до 40 біт, тоді як нестандартна специфікація забезпечує шифрування даних 128 та 256 біт довжини ключа. Приклад шифрування наведно на (рис. 1).

В галузі захисту комп’ютера, даних або мережі запропоноване рішення не завжди охоплює рішення для всіх областей, що мають слабкість у відповідному полі. Протокол WEP має деякі слабкі місця в безпеці, такі як:

- Аналізований захоплений мережевий трафік показав, що спільний ключ, який використовується WEP, може легко декодуватися, аналізуючи захоплені дані. Це може призвести до маніпулювання даними та втрати цілісності даних;
- Маленький розмір ключа: розмір ключа у стандарті WEP — лише 40-бітний ключ;

Проблеми автентифікації: Завдяки схемі відповіді на виклик, яка використовується при автентифікації спільного ключа, у WEP може бути здійснена атака “man-in-the-middle”.

Протокол WPA2 є покращеною версією WPA. 802.11i був повністю реалізований у протоколі WPA2. Основна зміна, яка була зроблена в WPA2 на відміну від WPA, стосується алгоритму шифрування даних. Протокол більше не використовує шифрування RC4, йому на заміну прийшов шифр AES який використовує 256 бітну систему шифрування.

Розглянемо процес шифрування на простому прикладі, візьмемо рядок даних “Hello World” і використаємо простий метод шифрування літера-число (letter-number). У цьому методі кожна буква англійського алфавіту буде відповідати певній цифри, (тобто A=1, B=2, C=3 і т.д.) ці дані перекладаються в числа “8 5 12 12 15 23 15 18 12 4”. Після цього цей ряд чисел передається через мережу і приймач може розшифрувати рядок, використовуючи той самий алгоритм, але у зворотному порядку, 8 перетворюється в H, 5 в E і т. д. Зрештою приймач отримає ціле повідомлення “Hello World”. Більшість методів шифрування використовують набагато складніші формули та методи. В нашому прикладі довжина ключа була приблизно 8 біт, але деякі клавіші можуть бути надзвичайно складні і можуть досягати 128 біт. Чим більше ключ у бітах, тим складніше шифрування, та там важче буде його зламати (рис. 2).

Для шифрування повідомлення та щоб розшифрувати зашифроване повідомлення, потріб-

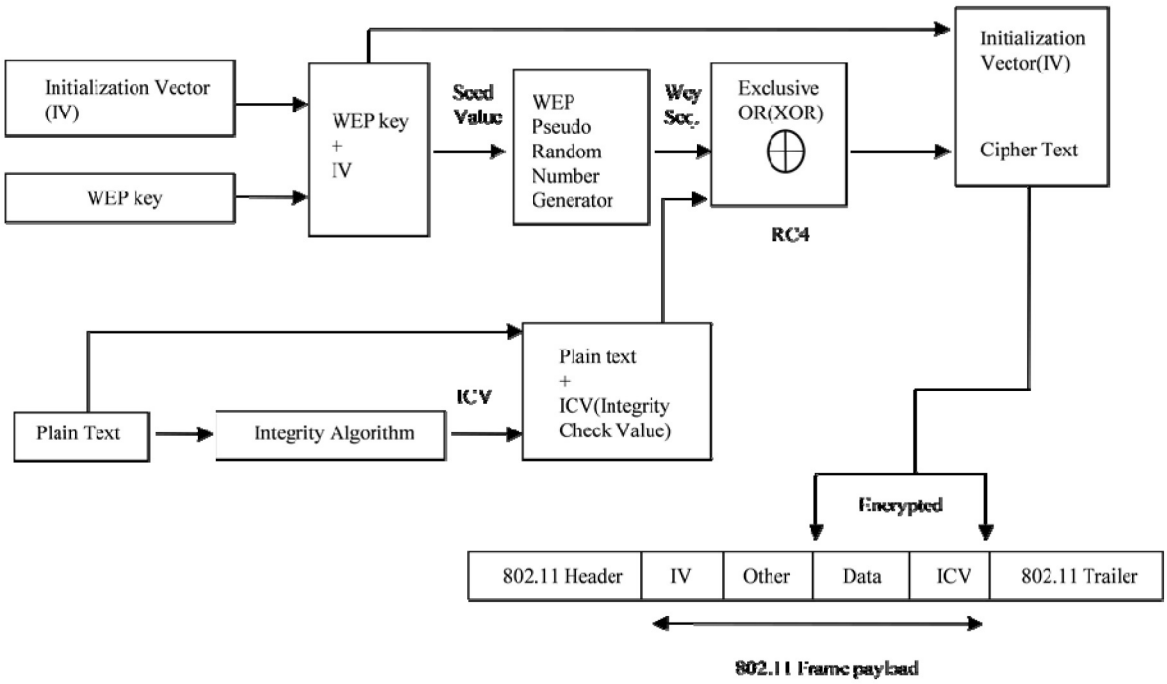

Рис. 1. Приклад шифрування у системі захищеної шифруванням WEP

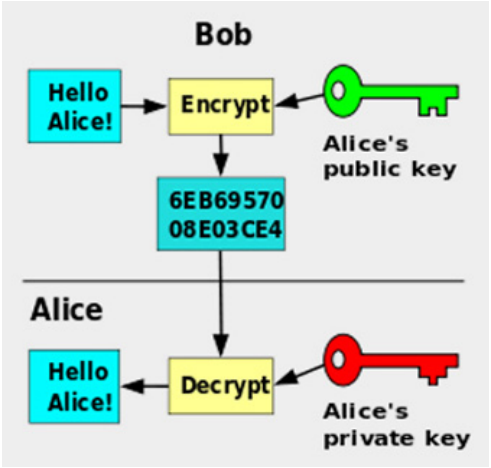

Рис. 2. Приклад шифрування повідомлення у інтернеті

но зашифрувати кожен символ за допомогою ключа. Ключ шифрування — це таблиця або формула, яка визначає спосіб перетворення зашифрованого символу. Ключі поділяються на 2 типи: відкриті (public) та закриті (private).

Криптографічне шифрування поділяють на Симетричну та Асиметричну системи шифрування.

Симетричне шифрування — Ці алгоритми шифрування включають методи шифрування, в яких відправник і одержувач мають однакові ключі. Такі алгоритми шифрування були єдиними відомими до 1976 року.

Криптосистема, в якій для шифрування та дешифрування використовується один і той самий ключ, називається симетричною. Безпека симетричної криптосистеми є функцією двох факторів: надійності алгоритму та довжини ключа (Searle, 1969).

Асиметричне шифрування — такі алгоритми шифрування використовують два типи відкритих і закритих ключів, які пов'язані між собою.

Приклад обох варіантів шифрування наведено на (рис. 3).

Розглянемо деякі сучасні алгоритми шифрування даних.

Алгоритм 3DES або Triple Des. Один із перших алгоритмів шифрування, оснований на розкладання чисел на прості множники. Візьмемо у приклад число 589, ключами цього числа будуть 19 та 31 (19*31=589). Банки до нашого часу використовують даний алгоритм шифрування, але використовують шифри з набагато більшою кількістю цифр.

Алгоритм AES (Advanced Encryption Standard). На даний момент один із найбільш розповсюджених алгоритмів шифрування. Алгоритм використовує 256 бітну систему шифрування. Це означає що пароль може бути довжиною до 32 символів з таблиці на 256 символів. Фактично це шифр простої підстановки даних. Візьмемо у приклад число 19 яке за таблицею перетворюється в зашифроване – d4.

Алгоритм RSA (Рівест, Шамір, Адлеман). Алгоритм заснований на розкладі великих цілих чисел на множники. Один з перших алгоритмів, придатних для шифрування цифрових підписів. Для шифрування даних нам потрібно мати 2 ключі, відкритий та приватний, разом вони складають пару. Відкритий ключ не потрібно приховувати, він потрібен лише для шифрування даних. Після того, як повідомлення зашифровано за допомогою відкритого ключа, його можна розшифрувати лише за допомогою закритого ключа.

DSA (Алгоритм цифрового підпису) — це криптоалгоритм, з відкритим ключем який вико-

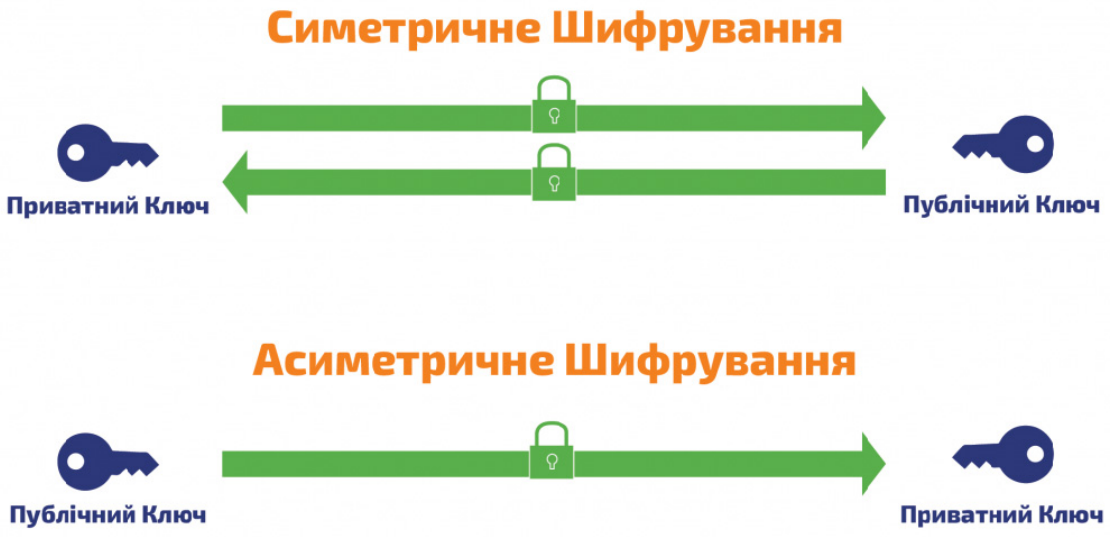

Рис. 3. Приклад симетричного та асиметричного шифрування

ристовують для створення електронного підпису, на відміну від схеми RSA. Створення підпису проводиться таємно, але його можна перевірити публічно. Принцип роботи алгоритму заснований на складності обчислення логарифмів.

Новий легкий криптографічний алгоритм (NLCA) шифрує дані на основі симетричної криптографії. Алгоритм – це 16-байтовий (128-бітний) блоковий шифр, який вимагає 16-байтовий (128-бітний) ключ для шифрування даних. Алгоритм простий і дуже безпечний для шифрування-дешифрування. Він натхненний архітектурними методами Feistel та SP для поліпшення складності шифрування. Запропонований алгоритм порівнював продуктивність з деякими криптографічними алгоритмами, а саме DES, AES, HIGHT, Blowfish, LED, використовуючи різні параметри, такі як розмір блоку, довжина ключа, можливий ключ, математичні операції, тип шифру та потужність захисту. У майбутньому алгоритм NLCA може бути впроваджений в апаратне забезпечення, що може дати набагато кращі результати (Thabit et al., 2021).

Розроблено нову криптосистему шифрування зображень за допомогою двовимірної декомпозиції з частковим розкручуванням (2D-PUD) (Yongfei et al., 2021). Щодо алгоритму шифрування RC5, який розглядають як швидкий симетричний блок-шифр, придатний для апаратних або програмних реалізацій. Новою особливістю RC5 є інтенсивне використання залежних від даних обертань (Ronald, 2005). Запропонований алгоритм (Aljawarneh, Yassein & Talafha, 2017) здатний захистити великі мультимедійні дані від атак в реальному часі. Експериментальні результати та аналіз безпеки підтверджують що алгоритм шифрування зображень, заснований на хаотичній системі та операціях з послідовністю дезоксирибонуклеїнової кислоти (ДНК), має не тільки чудовий результат шифрування, але й протистоїть різним типовим атакам (Xiuli, Yiran, Lucie, 2017). Останнім часом криптологія на основі хаосу стала одним з найпоширеніших методів проектування для розробки нових алгоритмів шифрування за останні два десятиліття (Özkaynak, 2018). Також розроблена основна конструкція, яка може базуватися на функціональних схемах шифрування, які підтримують єдиний функціональний ключ, і де схема шифрування зростає підлінійно за розміром схеми функції (Bitansky, Vaikuntanathan, 2018). А алгоритм роботи машини для зламання шифру „грубою силою" описано в (Wiener, 1993; 1994).

З бурхливим розвитком технологій квантових обчислень криптосистему з відкритим ключем також потрібно періодично оновлювати. Таким чином, криптографу доводиться постійно шукати протоколи з відкритим ключем, щоб протистояти атакам квантових обчислень, що призводить до появи постквантової криптографії. Варто зазначити, що хоча назва постквантової криптографії містить слово «квантова», вона, по суті, складається з протоколів класичної криптографії. Сучасні дослідження пост-квантової криптографії включають переважно протоколи, засновані на наступних чотирьох рамках: протоколи, засновані на функції Хеша, протоколи, засновані на коді виправлення помилок, протоколи, засновані на гратах, і багатоваріантні протоколи. В даний час перевірка відповідних протоколів кандидатів для постквантової криптографії проводиться NIST (Chen et al., 2016; Alagic et al., 2019). Хоча визначення безпеки цих протоколів є складною проблемою, ми не знаємо, чи існує ще квантовий алгоритм, який може змінити майбутню структуру криптосистеми з відкритим ключем (Kai Li et al., 2021).

Вказані алгоритми гарантують надійність захисту завдяки їх обчислювальній стійкості та стійкості до алгоритмічних атак (Kölbl, 2017).

Шифр RC4 був розроблений Роном Рівестом разом з компанією RSA Security у 1987. Офіційним скороченням абревіатури є Rivest Cipher 4, але дехто вважає назву скороченням від Ron's Code.

До вересня 1994 року шифр був комерційною таємницею, але опис алгоритму було анонімно відправлено до Cypherpunks (неформальна, анонімна група людей яка зацікавлена криптографією). Згодом опис RC4 був опублікований до криптографічної групи новин sci.crypt. Після цього код потрапив в Інтернет. Опублікований шифр дав такі ж шифротекста на виході, як і оригінальний RC4. Опублікований шифр був сумісний з продуктами які використовували RC4, а деякі учасники Cypherpunks, що за їх словами працювали з вихідним кодом RC4 підтвердили схожість використаного алгоритму.

Оскільки алгоритм опублікований в вільному доступі, він більше не являє собою комерційною таємницею, проте абревіатура "RC4" досі є торговою маркою компанії RSA Security, через це іноді шифр можуть називати «Alleged RC4» або «ARCFOUR» ("схожий на RC4", оскільки RSA Security досі не опублікувала вихідний код алгоритму).

Шифр RC4 використовують в широко поширених протоколах шифрування, наприклад: WPA, WEP і TLS. Одним із головних факторів поширеності використання алгоритму була простота його застосування та висока швидкість роботи.

Алгоритм RC4 будується на основі генератора генерації псевдовипадкових бітів. На вході записується ключ, а на виході читаються псевдовипадкові біти. Довжина ключа може складати від 40 до 2048 біт. RC4 може бути достатньо вразливим, якщо використовувати не випадкові та пов'язані ключі або якщо ключовий потік використовувати двічі. Ці 2 фактори можуть зробити криптосистему небезпечною.

У 1995 році Андрю Руз (англ. Andrew Roos) експериментально прослідкував, що перший байт ключового потоку пов'язаний з першими трьома байтами ключа, а перші кілька байт перестановки після алгоритму розкладу ключів (англ. KSA) корельований з деякою лінійною комбінацією байт ключа. Ці зміщення були доведені до 2007 року, коли Пол, Рафі і Майтре довели кореляцію ключа і ключового потоку. Також Пол і Мейтре довели кореляцію перестановки і ключа. Остання робота також використовує кореляцію ключа і перестановки для того, щоб створити перший алгоритм повного відновлення ключа з останньої перестановки після KSA, не роблячи припущень про ключі і векторі ініціалізації (англ. IV or Initial Vector). Цей алгоритм має постійну ймовірність успіху в залежності від часу, яка відповідає квадратному кореню з складності повного перебору. Пізніше було зроблено багато робіт про відновлення ключа з внутрішнього стану RC4 (Roos, 1995).

У 2001 році, Флурер, Мантінєє і Шамір опублікували роботу про уразливість ключового розкладу RC4. Вони показали, що серед усіх можливих ключів, перші кілька байт ключового потоку є зовсім невипадковими. З цих байт можна з високою ймовірністю отримати інформацію про використаний шифром ключ. І якщо довготривалий ключ і оказія (англ. Nonce) просто конкатенуються для створення ключа шифру RC4, то цей довготривалий ключ може бути отриманий за допомогою обробки достатньої кількості повідомлень, зашифрованих з використанням даного ключа. Ця вразливість і деякі пов'язані з нею ефекти були використані при зламі шифрування WEP в бездротових мережах стандарту IEEE 802.11. Це показало необхідність якнайшвидшої заміни WEP, що спричинило за собою розробку нового стандарту безпеки бездротових мереж WPA. Криптосистему можна зробити несприятливою до цієї атаки, якщо відкидати початок ключового потоку. Таким чином, модифікований алгоритм називається "RC4-drop [n]", де n – кількість байт з початку ключового потоку, які слід відкинути (Schneier, 2017).

Хоча про слабкості алгоритму давно і широко відомо, але реальні доступні способи реалізації атаки були опубліковані лише недавно. В ході криптографічної конференції Fast Software Encryption, що пройшла у Сінгапурі у березні 2013 року, професор Ден Бернштейн представив метод обходу захисту протоколів захисту TLS та SSL у випадку якщо вони використовували алгоритм шифрування RC4. Для розшифровки даних потрібно перехопити велику кількість зашифрованих даних, на розшифровку яких буде витрачено близько 30 годин. Проте можна захистити себе частою зміною ключа (приблизно раз у годину) та генерацію нового ключа за допомогою алгоритмів генерації ключів (наприклад PBKDF2).

У прикладі на (рис. 4) зловмисник інтегрує код JavaScript на незахищений веб сайт. Цей код спонукає передавати зашифровані запити, що містять веб файли cookie жертви. Відстежуючи дані зашифрованих запитів, можна відновити можливі значення файлів cookie. Всі файли в цьому списку перевіряються, поки не буде знайдено потрібний.

Для успішного дешифрування файлу cookie з 16 символів із ймовірністю успіху 94% потрібно відібрати приблизно $9 \cdot 2^{27}$ шифрувань файлів cookie. Оскільки ми можемо змусити браузер передавати 4450 запитів за секунду, ця сума може бути зібрана всього за 75 годин. Якщо зловмиснику трохи пощастить, потрібно захопити менше шифрувань. Подібна атака не обмежується дешифруванням файлів cookie. Будь-які дані чи інформація, які неодноразово шифруються, можуть бути відновлені. Приклад

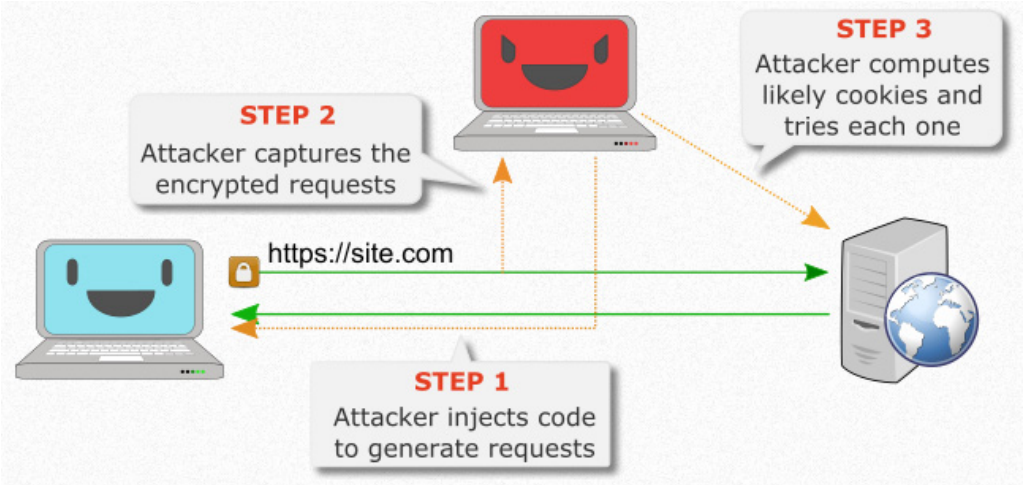

Рис. 4. Приклад злому шифру у три етапи

наданий на веб-файлах cookie в HTTPS, через те, що це ілюструє слабкі сторони RC4 та потужність можливої атаки.

Результати та обговорення

RC4 – це потоковий шифр та алгоритм ключа змінної довжини. Цей алгоритм шифрує один байт (або більше) за один раз. Основою роботи алгоритму є генератор псевдовипадкових бітів,

Таблиця 1

Таблиця елементів S

Номер елементу	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Значення елементу	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Таблиця 2

Таблиця елементів K

Номер елементу	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Значення елементу	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4

Таблиця 3

Таблиця елементів M

Номер елементу	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Значення елементу	10	13	14	12	2	15	6	4	5	3	1	9	8	7	0	11

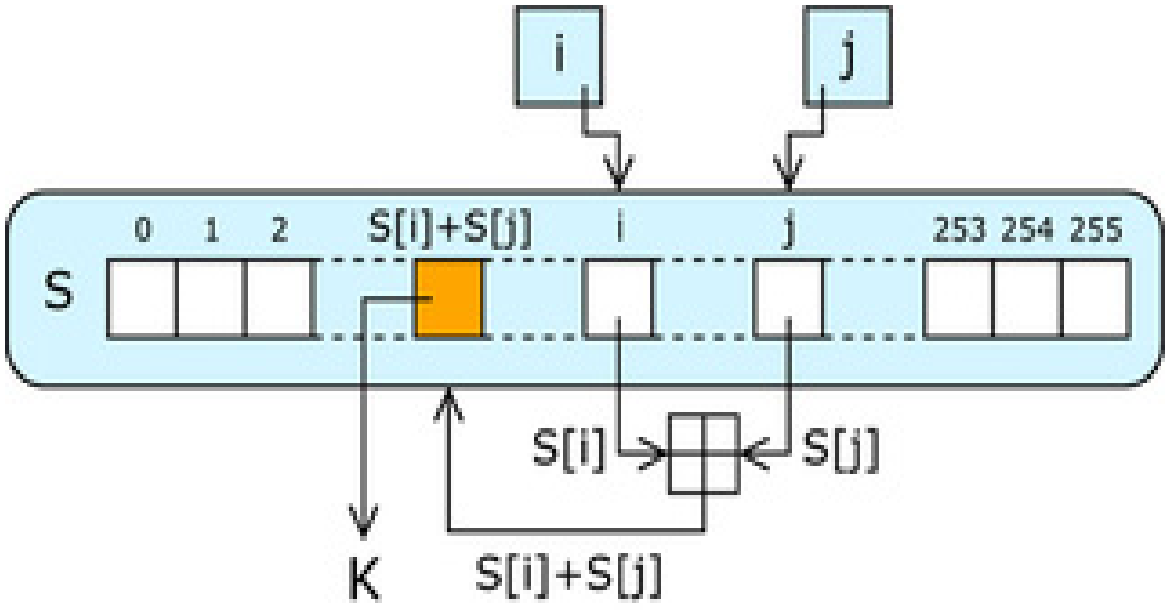

Рис. 5. Приклад генератору ключового потоку RC4

який створює 8-бітове число потоку, яке непередбачуване без знання вхідного ключа. Вихід генератора псевдовипадкових бітів називається ключовим потоком, який поєднується по одному байту з шифровим потоком відкритого тексту за допомогою XOR операції.

Алгоритм RC4 складається з двох етапів. На першому, підготовчому етапі проводиться ініціалізація таблиці замін S. На другому, основному етапі обчислюються вихідні числа.

Приклад заповнення таблиці змін S виконаний на псевдокоді:

```
j = 0;
i =0;
j = (j + Si + Ki) mod 16;
поміняти містами S[i] и S[j];
i = i +1;
якщо i <16, то перейти на п.2
```

В результаті виконання цього алгоритму проводиться заповнення таблиці замін S, причому це початкове перемішування значень проводиться в залежності від секретного ключа. Приклад генерації ключового потоку наведено на (рис. 5.).

Пояснення роботи алгоритму:

У приклад візьмемо ключ який складається з шести значень: “1 2 3 4 5 6”.

Заповнюємо таблицю S (табл. 1) послідовними числами від 0 до 15.

Заповнюємо таблицю K (табл. 2) записавши в неї ключ необхідну кількість разів.

Після заповнення таблиць, перемішуємо значення таблиці S(табл. 1) використовуючи Алгоритм 1:

1. j = 0; i =0
2. j = j + S[i] + K[i] = 0 + 0 + 1 = 1
3. Міняємо місцями S[i] та S[j]
4. i = i +1
5. i < 16, переходимо на пункт 2

Після виконання алгоритму, отримаємо таблицю 3 і зашифроване повідомлення.

Основний код програми складається з короткого алгоритму який може бути інтегрований та реалізований в багатьох програмах виконаних на C++ та інших мовах програмування.

Висновки

Функціонування майже всіх цифрових сервісів в наш час неможливе без шифрування даних. Ще декілька десятиліть назад проблема шифрування даних була не настільки гострою, оскільки передача даних проходила через відносно захищені канали даних, наприклад, підземний телефонний кабель (навіть якщо по ньому передавали незахищені дані, складність полягала в складності фізичного доступу до кабелю). Зараз же коли багато операцій проходять через бездротові мережі (Wi-Fi, мобільний інтернет, супутниковий зв’язок і т. д.) то захист каналів це першочергове питання. Вихід із цієї ситуації є, використання сучасних алгоритмів шифрування.

Список використаних джерел / References

1. Alagic G., Alperin-Sheri J.M., Apon D.C. et al. (2019) Status report on the first round of the nistpost-quantum cryptography standardization process, NIST Interagency/Internal Report (NISTIR) 8240
2. Aljawarneh, S., Yassein, M.B. & Talafha, W.A. (2017) A resource-efficient encryption algorithm for multimedia big data. Multimed Tools Appl 76, 22703–22724. https://doi.org/10.1007/s11042-016-4333-y
3. Bitansky N., Vaikuntanathan V. (2018) Indistinguishability Obfuscation from Functional Encryption Journal of the ACM November Article No.: 39 https://doi.org/10.1145/3234511
4. Chen L., Jordan S.P., Liu Y.K. et al. (2016) Report on post-quantum cryptography, NIST Interagency/ Internal Report (NISTIR) 8105.
5. Halunen K., Latvala O.-M., (2021) Review of the use of human senses and capabilities in cryptography, *Computer Science Review* 39 100340 https://doi.org/10.1016/j.cosrev.2020.100340
6. Kai Li, Pei-Gen Yan, Qing-Yu Cai, (2021) Quantum computing and the security of public key cryptography, Fundamental Research 1 85–87 https://doi.org/10.1016/j.fmre.2020.12.001
7. Kölbl S. (2017) Design and analysis of cryptographic algorithms: Ph.D Thesis. Lyngby
8. Özkaynak, F. (2018) Brief review on application of nonlinear dynamics in image encryption. Nonlinear Dyn 92, 305–313. https://doi.org/10.1007/s11071-018-4056-x
9. Roos A. (1995) A Class Of Weak Keys In The RC4 Stream Cipher 22 Sep 1995. [Електронний ресурс] – Режим доступу: URL: https://netfuture.ch/1995/
10. Ronald L. (2005) The RC5 encryption algorithm Fast Software Encryption pp. 86-96 https://doi.org/10.1007/3-540-60590-8_7
11. Schneier B. (2017) Applied Cryptography: Protocols, Algorithms and Source Code in C
12. Searle J. R. (1969) *Speech Act*. Cambridge University Press.
13. Thabit F., Alhomdy Sh., Abdulrazzaq H.A. Al-Ahdal, Jagtap S., (2021) A New Lightweight Cryptographic Algorithm for Enhancing Data Security In Cloud Computing, *Global Transitions Proceedings.* https://doi.org/10.1016/j.gltp.2021.01.013
14. Xiuli C., Yiran C., Lucie B. (2017) A novel chaos-based image encryption algorithm using DNA sequence operations, Optics and Lasers in Engineering, Vol. 88, Pages 197-213, ISSN 0143-8166 https://doi.org/10.1016/j.optlaseng.2016.08.009
15. White S. R. (1990) Covert Distributed Processing with Computer Viruses, Advances in Cryptology CRYPTO ‘89 Proceedings, *Springer-Verlag*, 616-619.
16. Whitman M. & Mattord H. (2005). Principles of information security.
17. Wiener M. J. (1993) *Efficient DES Key Search*. presented at the rump session of CRYPTO ‘93, Aug 1993
18. Wiener M.J. (1994) *Efficient DES Key Search*, TR-244, School of Computer Science, Car leton University, May 1994.
19. Yongfei W., Liming Z.,Tao Q. et al. (2021) Content-adaptive image encryption with partial unwinding decomposition Signal Processing Vol. 181, April 2021, 107911 https://doi.org/10.1016/j.sigpro.2020.107911