УДК 681.3

Добровольський А. О., магістрант, Скрипник І. А., доц., к.ф.-м.н.

ДОСЛІДЖЕННЯ ТА МОДИФІКАЦІЯ АЛГОРИТМІВ БЛОКОВОГО ШИФРУВАННЯ ДАНИХ

Шифрування – оборотне перетворення інформації з метою приховування від неавторизованих осіб, з наданням в цей же час авторизованим користувачам доступу до неї. Головним чином, шифрування служить завданням дотримання конфіденційності інформації, що передається. Важливою особливістю будь-якого алгоритму шифрування є використання ключа, який стверджує вибір конкретного перетворення з сукупності можливих для даного алгоритму.

В сучасних криптосистемах (з відкритим ключем) для шифрування, розшифрування даних можуть використовуватися різні ключі. Однак, з розвитком криптоаналізу, з'явилися методики, що дозволяють дешифрувати закритий текст без ключа [1-3].

Метою роботи є дослідження алгоритмів блокового шифрування, знаходження сильних та слабких сторін серед існуючих алгоритмів, оцінка швидкості шифрування та розшифрування тексту, визначення криптостійкості щодо алгоритмів шифрування.

Алгоритми модернізовані за допомогою додавання випадкових послідовностей у шифр та зашифрування ключа у повідомленні, щоб не використовувати спеціальний захищений канал передачі даних для передачі ключа.

Симетричні системи мають перевагу над асиметричними в швидкості шифрування, що дозволяє їм залишатися актуальними, незважаючи на більш слабкий механізм передачі ключа – одержувач повинен знати секретний ключ, який необхідно передати по вже налагодженому зашифрованому каналу.

Гнучкість блокових шифрів дозволяє використовувати їх для побудови інших криптографічних примітивів: генератора псевдовипадкової послідовності, поточного шифру, імітовставки і криптографічних хешів.

Початковий текст – це набір вхідних даних для шифрування. Він може подаватись у вигляді символів, введених у текстове поле інтерфейсу або файлу, який буде завантажено у додаток.

Алгоритм шифрування:

1. Генерується 56-ти бітовий ключ на основі випадкових даних за допомогою таблиці ASCII та вбудованого класу Random.
2. Повідомлення розбивається на стандартні блоки по 64 біта і шифрується цим ключем (якщо останній блок менше 64 біт, то додаються данні у виді нулів таким чином, щоб довжина стала 64 біти).
3. Генерується 2 числа на основі поточної дати: n1 – це номер дня місяця, n2 – це номер місяця.
4. Ключ у незашифрованому виді додається у зашифроване повідомлення після n1-го біта.
5. Додавання випадкових послідовностей. У отримане повідомлення додаються випадкові дані з таблиці ASCII довжиною n1*n2. Дані додаються у зашифроване повідомлення з ключем після n2-го біта.

Далі користувач отримує зашифроване повідомлення, готове до відправки.

Алгоритм розшифрування:

1. На основі поточної дати генеруються 2 числа.
2. Спочатку видаляються випадкові послідовності починаючи з n2-го біта довжиною n1*n2.
3. Видаляється ключ довжиною 56 біт.
4. За допомогою ключа та алгоритму DES розшифровується повідомлення.

Генерації чисел n1 та n2 проводиться на основі системного часу. Тут використовується мітка часу (Timestamp) і захищена мітка часу. Остання створюється сервером аутентифікації (timestampingAuthority, TSA) і використовується для підтвердження існування певних даних до певного моменту часу без можливості дописування заднім числом.

Техніка створення TSA заснована на цифрових підписах і хеш-функціях. Спочатку хеш обчислюється з даних. Якщо оригінальні дані були змінені, то вийде вже повністю інший хеш. Цей хеш надсилається TSA, TSA генерує timestamp для хешу і обчислює хеш їхньої конкатенації. Новий хеш, наприклад, може бути підписаний в цифровій формі з приватним ключем TSA. Цей підписаний хеш і timestamp повертаються на підписавшу сторону і зберігаються разом з оригінальними даними.

Згодом оригінальні дані не можуть бути обчислені з хешу, TSA ніколи не бачить оригінальні дані, які допускається використовувати в цьому методі для конфіденційних даних.

Для шифрування мітки часу ми будемо використовувати хешування, а саме схеми безпечного хешування.

Результати теоретичних та практичних досліджень.

Після розробки модернізованого алгоритму DES , який полягає у зашифруванні даних та додавання випадкових послідовностей, можна побачити що алгоритм потребує більшого часу на шифрування і має більшу крипто стійкість.

Порівняння представлено у таблиці, де DESM – модернізований алгоритм DES.

	DES	DESM 1 раз
Швидкість зашифрування	65.87 mb/s	38.22 mb/s
Швидкість розшифрування	71.41 mb/s	40.04 mb/s
Кількість операцій для злому	2^{55}	-

Оскільки ми додаємо випадкові послідовності даних та дописуємо ключ безпосередньо у наше повідомлення, то наш модернізований алгоритм й програє у швидкодії, але якщо взяти до уваги те, що розшифрувати його можливо, лише знаючи, які біти потрібно видаляти та звідки брати ключ, це робить його абсолютно криптостійким.

Після дослідження предметної області та детального аналізу поставленої проблеми було встановлено актуальність модернізації алгоритму шифрування DES. Шляхом аналізу його швидкодії й криптостійкості та проведення теоретичних і практичних досліджень було сформовано набір методів для дослідження, проведено їх порівняння та аналіз, а також було визначено набір технологій та інструментів розробки, визначено їх відносну продуктивність, ступінь інтеграції та зручність використання.

Основним інструментом для розробки стало інтегроване середовище розробки Visual Studio 2017 та .NET Framework. Алгоритм шифрування розробляється на мові C#.

За допомогою досліджень швидкодії та криптостійкості алгоритмів було визначено, що шифрування тексту модернізованим алгоритмом проходить повільніше, ніж стандартним DES, що не досить істотно впливає на результат, але криптостійкість суттєво збільшується, тому DESM може конкурувати з сучасними алгоритмами, як симетричними так і асиметричними.

Література

1. Бабенко Л.К., Ищукова Е.А. Современные алгоритмы блочного шифрования и методы их анализа. – М.: Гелиос АРВ, 2006.-376 с., ил.

2. Нильс Фергюсон, Брюс Шнайер, Практическая криптография. Вильямс, 2005 г., 424 с.

3. Шнайер Б. Прикладная криптография. – М. : Триумф, 2002. – 816 с.